Artificial Intelligence: Term Paper

Implementation of Cheat

b99901124 葉季昇
b99901132 余朗祺
b99901164 趙冠琳

摘要

我們用撲克牌遊戲“吹牛”來研究人類的智慧行為之一：說謊與提防說謊的模式。為簡化問題與分析，我們使用具特定玩牌模式的 AI 玩家來測試我們的成果，而非直接由人類玩家來測試。在第一階段中，遊戲簡化成只有兩種點數(黑白)，AI 觀察大量數據後逐步調整“抓”的策略，最後可以學習出對手說謊的傾向。第二階段中，我們嘗試更接近真實遊戲的場景：學習 AI 採用強化學習(Reinforcement Learning)，從遊戲狀態中抽取出若干特色函數用來學習 Q 值，結果發現：對於不同種類的個性 AI 我們的學習 AI 也會表現不同，某些個性 AI 的個性較單純，學習 AI 會有較高勝率。此外，使用不同的特色函數組合也會導致學習 AI 表現得不一樣：記憶越多“說謊歷史”，對上某些個性就會有較高勝率。

一、簡介

(一) 傳統吹牛的規則

把 52 張撲克牌全發給所有玩家。決定出牌順序後，第一位玩家便可開始出牌，一般情況由發牌者首先出牌。玩家依照 A、2、3、……、J、Q、K、A、……以此類推的出牌點數順序出牌，玩家必須出一至 MAX_PLAY 張牌（MAX_PLAY：玩家可以出的最大牌數），背面向上，覆蓋在桌上。玩家可以說謊，即出的牌點數不一定如同應該出的。其他玩家可以相信，又或指證剛出牌的人說謊（只能指証剛剛出牌者），並把牌翻過來看。如果出牌人真的說謊，整疊牌就要還給出牌人，並由揭謊者開始出牌。如果出牌人沒有說謊，整疊牌就要給予揭謊者，並由出牌人開始出牌。最後，只要出完牌又躲過最後一次指証者，即為勝者。

(二) 選擇「吹牛」的目的

若 AI 能夠下棋或玩其他具有規則的遊戲，代表該 AI 具備在複雜的多變量狀況邏輯、預測、決策的能力。我們希望寫出的 AI 不只是在各種牌面狀況下進行判斷，而需要針對更複雜的遊戲局勢，推測對手的下手動作，採取較多元的動作，更接近人類智慧的本質。

因為吹牛遊戲最大的特色就是玩家可以「說謊」以及「推測」其他玩家說謊的模式。而說謊行為在一般認知中，被視為是一種智慧的表現，我們認為電腦在吹牛遊戲中的說謊行為，比起普通的下棋程式，更接近人性及智慧的表現。

(三) 具個性的 AI 玩家

具有部分隨機特性的個性 AI 玩家可以產生大量的測試資料，而且個性鮮明的個性 AI 產生的測試資料較容易驗證學習 AI 的學習效果（可以從特色函數的權重看出來，各種個性 AI 在各種特色函數的權重有高度差異）。

二、第一階段

(一) 目的

由於「吹牛」遊戲與人類決策十分複雜，在第一階段中，我們將遊戲進行簡化，並嘗試利用「具個性 AI 玩家」來模擬人類決策。

(二) 第一階段遊戲規則

簡化過後的遊戲規則如下：(1)兩方各拿十張牌，其中黑牌與白牌的數量隨機分配。 (2)雙方分別將黑牌與白牌放在編號 1~10 的 10 個位置上，10 個位置的指定牌別依序是「黑白黑白黑白黑白黑白」，若與指定牌別不同，就是在該位置上說謊。 (3)同時，雙方分別猜測對手在 10 個位置上是否說謊。「無」表示不去抓說謊，「抓」表示去抓說謊，因此會送出類似下面的序列：「抓無抓抓無無無抓無無」。 (4)最後做分數結算，同一位置上，說謊被抓到，說謊者扣 1 分；說謊沒被抓到，說謊者加 2 分；沒說謊誤抓，誤抓者扣 1 分。將 10 個位置的分數累加，分數高者勝，一樣則平手。

(三) 具個性 AI 玩家

我們針對第一階段的遊戲規則，設計了 5 個「有個性的 AI 玩家」如下：

玩家 1 說謊與誠實機率相同，各個位置說謊的機率也相同；玩家 2 說謊與誠實機率相同，前面位置的說謊機率較高；玩家 3 說謊與誠實機率相同，後面位置的說謊機率較高；玩家 4 說謊與誠實機率相同，中間位置的說謊機率較高；玩家 5 誠實的機率較高，說謊的機率與位置無關。

(四) 學習機制

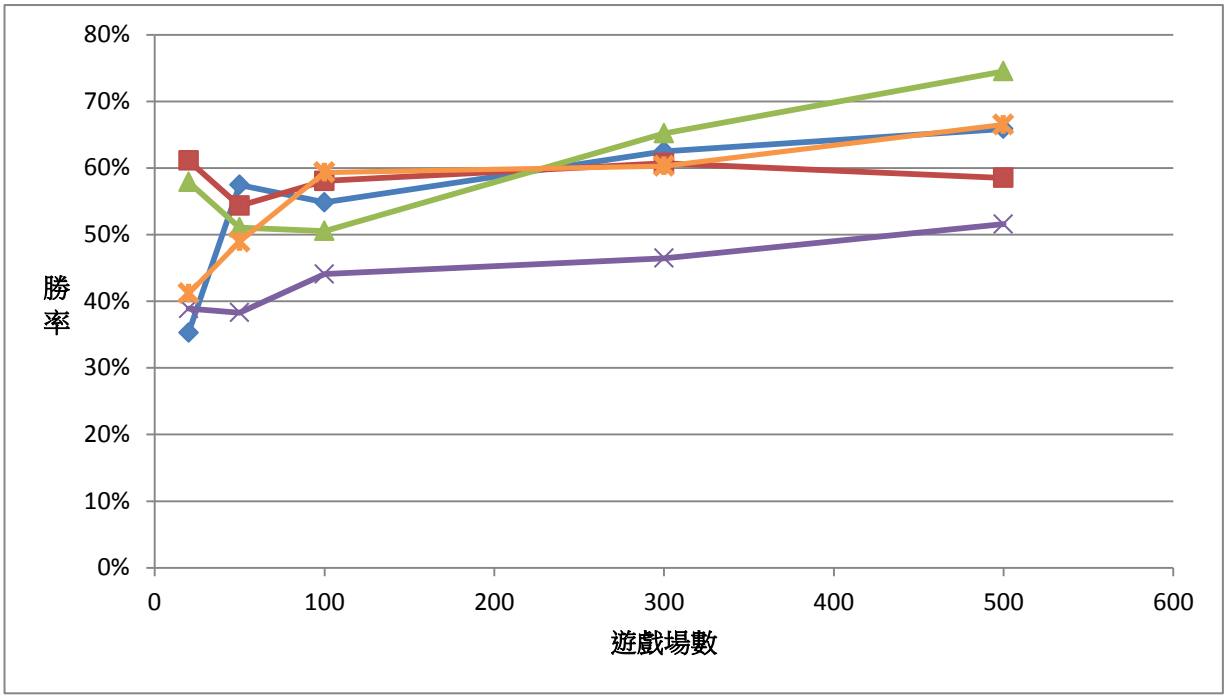
為簡化學習目標，學習 AI 只對「抓」的決策做出改進，自己出牌部分則始終維持說謊位置隨機。學習 AI 會記憶特定對手在 10 個位置個別的說謊紀錄，譬如經過 500 場遊戲，對手在 6 號位置上說謊 343 次。如此一來，學習 AI 可藉由紀錄遊戲結果改變「抓」的決策。

學習 AI 在每個位置上，「抓」的機率為 p+[(該位置說謊次數)/(所有位置說謊次數總和)-0.1]*k，其中 p 和 k 稱作 basic probability、influence factor，會隨著遊戲一次次的進行跟著改變。採用上列公式，是希望各個位置上，過去說謊機率高就增加「抓」的機率，反之則減少。此外，p 代表學習 AI 對於「整體來說，『抓』或『不抓』哪個好」的觀察，k 代表過去說謊機率對「抓」的機率之影響程度。

採用特色 AI 讓我們能在短時間內重複許多場遊戲，我們的「高成本」學習，正是建立在擁有大量數據的基礎上。每場遊戲結束後，learning agent 都會由(p,k)這組參數產生(p-Δp,k-Δk), (p-Δp,k), (p-Δp,k+Δk), (p,k-Δk), (p,k), (p,k+Δk), (p+Δp,k-Δk), (p+Δp,k), (p+Δp,k+Δk)這九組參數，每組參數分別進行十場模擬遊戲，積分(勝得 2 分、平手得 1 分)最高的參數們，隨機選一組成為新的參數。之所以使用「勝得 2 分、平手得 1 分」的積分而不是規則內的分數結算法，是因為希望的學習方向為贏越多場次越好，而非總勝分差越大越好。

(五) 成果與結論

我們並未特別把訓練階段獨立出來，而是直接進行遊戲，藉著勝率的變化觀察學習情形。實際的遊戲結果如下表：(藍：個性 1；紅：個性 2；綠：個性 3；紫：個性 4；橘：個性 5)



註：五個資料點對應的遊戲場數分別是 20、50、100、300、500

特色 AI 玩家 1 和 5 所有位置說謊的機率皆相同，但他們的勝率並未較其他玩家低，這表示每個位置說謊次數的統計，對勝率影響還沒到很顯著的地步。儘管如此，觀察與每個玩家對戰的成績，勝率幾乎都超過五成且往上提升，尤其是 100 場到 300 場、500 場時。這表示「抓」的機率公式設定合理(勝率超過五成)，且參數的學習機制是對的(勝率提升)。由此可知，一個人工智慧程式要學習「遊戲」，找出與遊戲規則對應的決策法，會比學習「人的特性」，針對人不經意的習慣加以利用來得容易。藉由統計，我們的確能知道不同對手的差異，譬如到五百場時，不同 agent 累積的說謊次數分布會有著明顯差異，但如何有效利用這差異又是另一個難題，至少第一階段採用的類似基因演算法之作法，尚未有效達成此點。

特色 AI 玩家是否能模擬人類行為，仍是懸而未決的問題，畢竟一個指令，特色 AI 就能產生大量數據，而人類無法。但第一階段之特色 AI 涵蓋了對特定位置有不同反應與沒有不同反應兩種情形，且它們的表現有一定之相似度，這表示，無論人類在這遊戲當中有說謊之偏好位置與否，應該都會產生類似特色 AI 的對戰數據。雖然沒有實際數據佐證，但學習 AI 玩家遇上人類對手，勝率沒意外還是會越玩越高。

三、第二階段

(一) 概述

在第二階段中，我們寫出了完整的吹牛遊戲程式，觀察我們的策略在真實遊戲中是否也能奏效。這個遊戲可以多人對戰，也能加入如第一階段那樣各種具有個性的 AI 玩家。然而，傳統的吹牛遊戲很容易陷入無限循環，這是因為玩家勝出的條件太嚴苛：在最後出牌時，牌的種類沒什麼選擇但還是必須照實出牌。為了解決這問題和簡化分析，我們自訂與修改了一些規則：(1)只要手牌數歸零就勝利，其他玩家不能抓最後一次的出牌；(2)只能抓上家，也就是剛出牌的人；(3)每人手牌數設為 5，整副牌有 5 種數字。為了降低記牌的意義，每種數字的牌的總數並非 4 張，而是隨機；(4)遊戲超過一千回合(出一千次牌)即中止；以及(5)應該出的數字是照順序循環的，例如第一個人應該出“A”，第二個

人出“2”…依此類推，而非像傳統那樣由抓人成功者/被抓失敗者決定。我們的研究著重在兩人對戰，故這些規則其實對遊戲進行影響不大。

(二) 學習機制：

我們的學習 AI 玩家使用增強學習(Reinforcement Learning)機制。遊戲狀態以下列七點特色來表現：(1) stack 中的總牌數；(2)剛剛出牌者(也就是對手)的手牌數；(3)剛剛出牌者的出牌數；(4)目前盤面上持牌最少玩家的手牌數；(5)目前玩家總人數(扣除已獲勝的玩家)；(6)自己最近三次說謊是否成功；以及(7)剛剛出牌者最近三次說謊是否成功。以上並非使用真正的數值，而是用級分制區分：像是牌數以[起始手牌數]/3 來區分為多、中、少三個等級。利用這些特色的線性疊加，我們就能得到遊戲狀態的 Q 值，進而求出在該狀態下最理想的行動。行動粗略分為抓人、說謊與說實話。獎勵(reward)暫且定義為：若說謊失敗，2x；抓人抓錯，x；說謊而沒被抓，2x；抓人抓對，x；勝利，100；失敗，-100。其中 x 代表該回合結束後手牌的**減少量**(負數代表增加)或對手手牌的**增加量**(負數代表減少)。

至於學習，因為行動的獎勵在對手回應前是無法得知的(未知會不會被對手抓)，AI 玩家並非在每次行動後就更新特色的權重，而是在清空 stack 之際再根據歷史紀錄學習。更新特色權重公式如下：

$$\theta_i \leftarrow \theta_i + \alpha\left(R(s) + \gamma \max_{a' \in A(s')} \hat{Q}_\theta(s', a') - \hat{Q}_\theta(s, a)\right) * f_i(s, a)$$

這是定義$\hat{Q}_\theta(s,a) = \sum_i \theta_i f_i(s,a)$的情況下，其中$f_i(s,a)$是我們的特色函數，$\theta_i$是對應特色函數 i 的權重。

(三) 學習場景

我們的學習場景有三種變因：(1)與不同個性的 AI 玩家對戰；(2)改變訓練次數；以及(3)改變特色。這些改變的定義如下：

(1) 各種特性 AI

Agent Type	階段性策略	作法	
a	• 積極出完所有的牌 • 不抓人	有牌	誠實並出最多張
		沒牌	說謊並出最不適合的rank最多張
b	• 分遊戲進行的前中後期 • 前期積極，中期以後保守	前期	有牌 誠實並出最多張
			沒牌 說謊並出最不適合的rank最多張(一次全出)
		中期	有牌 誠實並出最多張
			沒牌 盡量誠實，如要說謊就出隨機張數(一次全出)
		後期	有牌 誠實並出最多張
			沒牌 就抓，如果場上沒牌，就出一張
c	• 分遊戲進行的前中後期 • 前期積極，中期以後保守	前期	有牌 誠實並出最多張
			沒牌 說謊並出最不適合的rank最多張(各種點僅一張)
		中期	有牌 誠實並出最多張
			沒牌 盡量誠實，如要說謊就出隨機張數(各種點僅一張)
		後期	有牌 誠實並出最多張
			沒牌 就抓，如果場上沒牌，就出一張
d	• 永遠誠實	有牌	誠實並出最多張
		沒牌	就抓，如果場上沒牌就只出一張
e	• 考慮前幾次的說謊被抓情形 • 被抓的少積極，被抓的多保守	有牌	誠實並出最多張
		沒牌	前三次說謊被抓0次，出三張、1次，出兩張、2次，出一張、3次，抓、如果場上沒牌，就出一張；隨機出牌
f	• 考慮前幾次的說謊被抓情形 • 被抓的少積極，被抓的多保守 • 不抓人	前三次說謊被抓0,1次，出三張、2次，出兩張、3次、出一張 不論有無牌:都出最不適合的牌	

(2) 訓練 50、100、300、500、1000 場。

(3) 參照(二)學習機制：1. 僅考慮(1)~(5)；2. 考慮(1)~(7)；以及 3. 考慮(1)~(7)，但(6)(7)的定義更

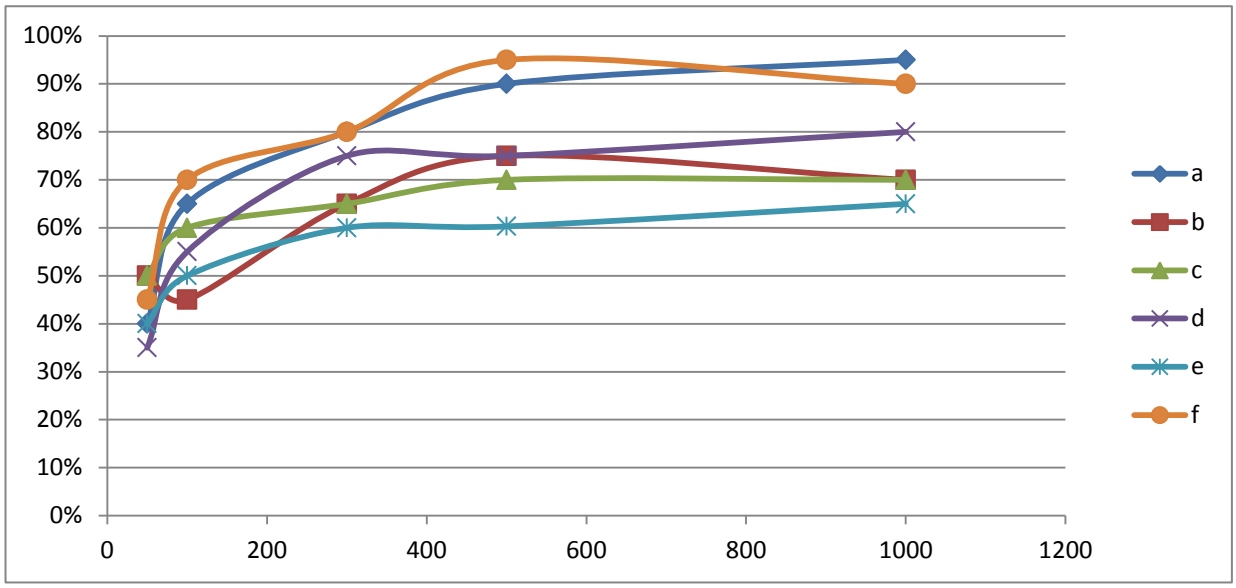
改為“最近五場”。

分析過程一次只改變(2), (3)其中一個變因，並針對不同個性 AI 比較學習的效果。若只改變(2)，則(3)採取第 2 方案；若只改變(3)，則訓練 300 場。預計成果是：訓練次數越多、考慮越多特色、記得越多說謊成敗歷史勝率就會越高。

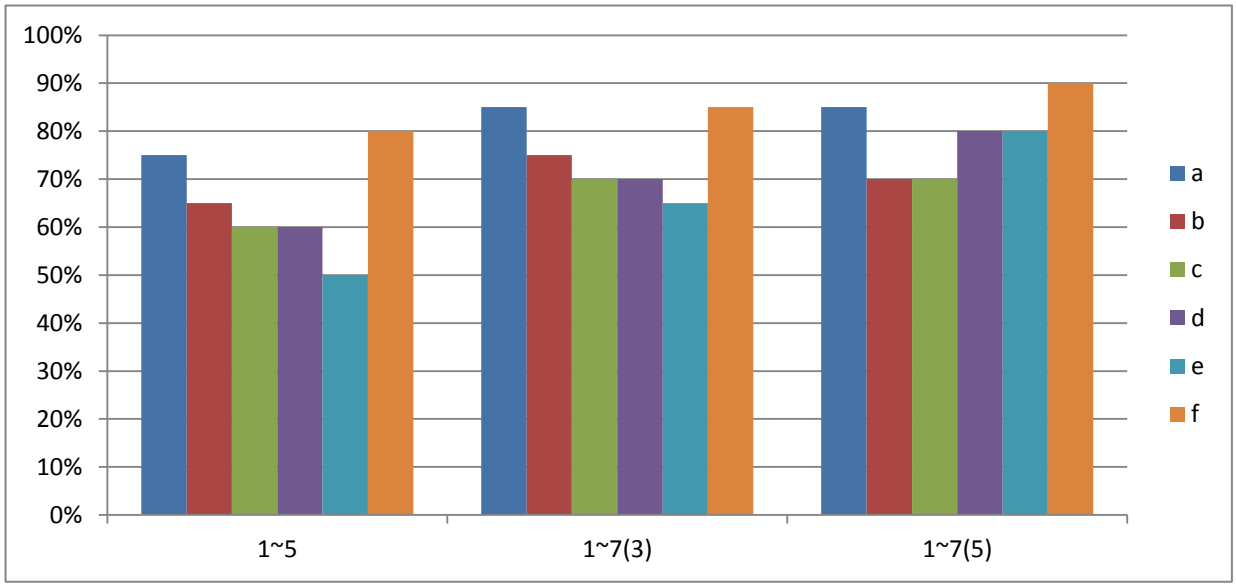
學習時，α 定為 0.7，γ 定為 0.9，雜音(隨機選擇行動的機率)為 0.5。測試時，直接把 α 和雜音重設為 0。

(四) 二人對戰成果

1. 訓練場數與勝率(測 20 場)



2. 選用特色函數與勝率(測 20 場)



很明顯的，隨訓練場數增加勝率也隨之提高，但超過 100 場後上升趨緩，這點和第一階段結果類似。值得注意的是，學習 AI 對抗特性 a, f 的勝率最好，對其他特性就平平。觀察這些特性，a 和 f 特性都不抓人，這應該會誘使學習 AI 說謊，因此更容易獲勝。其他四者，d 的勝率似乎略高於 b, c, 和

e，這應該是因為 d 的個性比其他三者明確：d 幾乎都誠實，但 b, c, 和 e 的個性比較抽象，可能較不好用我們選定的特性函數表現。雖然 d 的個性明確，但對抗 d 卻無法有和對抗 a, f 一樣高的勝率，這應該是因為對抗 d 的最佳策略是不說謊、不抓，盡量說實話，但“說實話”並不是隨時都能進行的行動。相對的，對抗 a, f 最好就是沒牌都說謊話，這點是必然可達成的。

特色方面，不考慮說謊成敗歷史確實讓勝率有顯著下降(約 10%左右)，但多考慮兩場，勝率反而沒有很大的改變。有趣的是，考慮越多歷史，對抗 d 與 e 的勝率會較對其他個性的勝率提升較多。e 的情況很明顯，因為它的出牌模式與歷史有關；d 方面較難解釋，可能是因為它說謊時確定只出一張牌，這簡化了預測 d 說謊模式的工作。

四、結論

增強學習對於預測說謊模式相當有效，即使我們只是隨意定義了一些特性函數，在大量資料的訓練下還是可以看到勝率的提升。如果想要更有效預測說謊模式，最經濟的方法是加入特性“說謊歷史”，只要記住近三次自己/對手說謊的成敗情形，就能提升勝率約 10%。記住更多歷史對一些個性可能有用，但整體改變不大。

雖然我們兩階段實驗都是雙人遊戲且對象並非人類，但相同策略可以推廣到多人遊戲：多人遊戲可近似成同時與許多人玩雙人遊戲，這樣就可以採用與上面提到相同的方式設計 AI；又或者可以嘗試新的、能代表多人遊戲性質的特色函數。這個策略要用在人類玩家上，最大問題就是訓練次數的不足，或是在達到訓練效果前人類玩家又改變了策略等等。在這種情況下，我們就要能找到更好的一組特色函數來進行學習。

參考書目

Richard S. Sutton and Andrew G. Barto, *Reinforcement Learning: An Introduction*. The MIT Press