Results

Raphaël Morsomme

2019-01-04

Contents

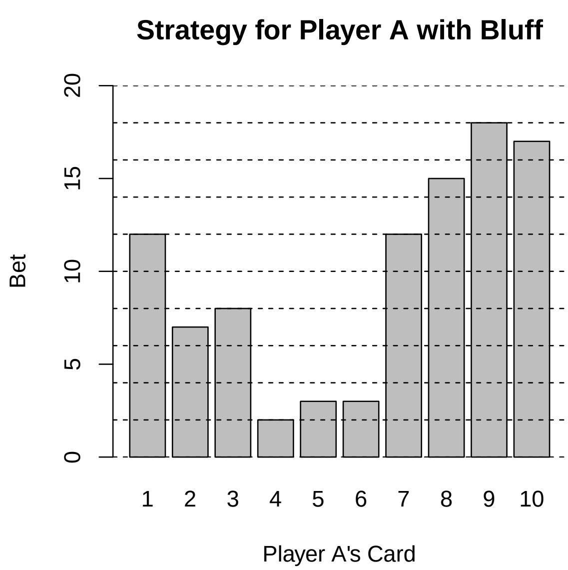
```r
library(tidyverse)
library(gdata)
```

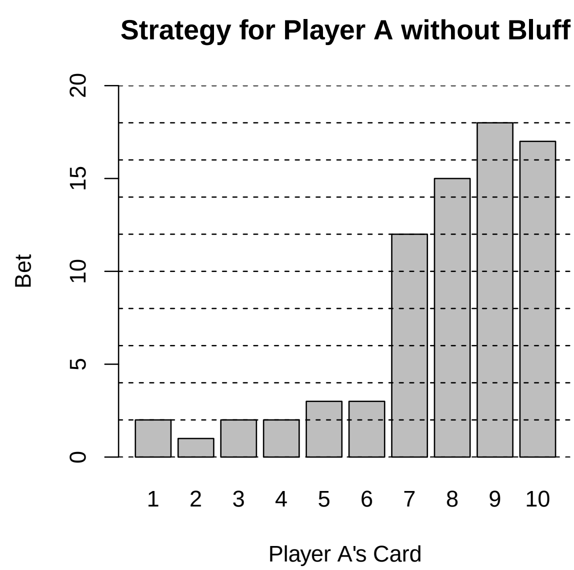
1 Introduction

Now that we have the function `my_GA` to learn strategies for poker with a GA, let us experiment with the parameters of the algorithm to anwser two questions: (i) what conditions encourage player A to bluff and (ii) is there any trick that player B can adopt to stop player A's bluff. It appears[1] that the size of the ante and the mutation rate have an impact on player A's level of bluff. The former determines the profitability of the bluff with large antes encouraging player A to bluff more and the latter determines the stability of the environment with low rates making the environment more stable, hence more favorable to bluff. After discussing these results, I mention elements to incorprate in the analysis in the future.

1.1 Defining Bluff

Since our version of poker only allows player A to bluff (player B can only call or fold), we exclusively focus on player A's strategy and how often (s)he bluffs. We use a simple definition of bluff for player A: player A bluffs if (s)he bets large amounts for small cards. In this script, we assess whether player A bluffs *visually*[2]. The following two plots displays a strategy with bluff and the same strategy without bluff. They should guide us in our interpretation of the results.





2 Results

Two factors influence player A's level of bluf: the size of the ante and the mutation rate.

[1] Since the number of plots is very large, I discuss the results first and show the graphs at the end.
[2] I discuss alternative methods in the last section.

2.1 Ante

The size of the ante determines how profitable it is for player A to bluff. Indeed, player A wins the ante if player B does not call the bet. A large ante encourages player A to bluff more often and a small ante has the opposite effect. In the extreme case, when `ante = 0`, player A should never bluff as (s)he can only lose by doing so: even if player B does not call the bet, (s)he wins nothing (ante is `0`).

This is exactly what we find in the simulations. We can see that when we set the ante to `0` (first series of plots), player A adopts an extremely conservative strategy: only bets when (s)het receives a `10` (see generation `40` and after)[3]. On the contrary, when we increase the ante to `10` (second series of plots), player A bluffs much more often (e.g. generation `120`). The latter situation is more favorable to player A who wins on average `3` units per hand played (versus `0` when the ante is `0`).

This shows that, unsurprisingly, the more profitable the bluff, the more player A bluffs. This has the following consequence for situations involving negotiation: bluf (and expect people to bluff) if, and only if, it is worth it. This means that a way to prevent someone else form bluffing is to convince the person that bluffing is not worth it, that the gain of a bluff are too small for the bluff to be profitable. As for people who love watching poker, this result means that the level of bluffs should increase as the game progresses and the size of the ante increases.

2.2 Mutation Rate

By determining how different the child strategies are from their parent strategies, the mutations rate determines how stable the environment is. A small mutation rate results in child strategies that are very similar to their parents, hence a stable environment where children share the features of their parents. A large mutation rate makes the environment more unstable since the generated child strategies have a large number of *random* mutations.

The link between stability and bluff is not obvious at first glance (at least it was not to me). Yet, we observe that player A's level of bluff increases when the mutation rate is low (e.g. generation `60` for `mutation_rate = 0.01`) and decreases when the rate is high (e.g. all generations for `mutation_rate = 0.1`)[4]. This indicates that a stable environment encourages bluff while a unstable one discourages it. Indeed, bluff has no chance to succeed in an environment where the opponent plays randomly (due to random mutations) and thereby calls many bluffs. Bluff can only succeed when the opponent plays in a relatively predictive way (in our case, by closely following the strategies of their parents). When the mutation rate is low (`0.01`), we observe that player B adopt a cautious approach, folding most of the time – especially when player A bets a large amount – meaning that player A's bluffs are rarely called and making bluffs attractive. On the contrary, when the mutation rate is high (`0.1`), player B plays more erratically due to the many random mutations occuring in her/his strategies, calling, as a result, player A's bluffs more often. In such situation, bluffing is less profitable to player A. yet, one should note that by playing more erratically, player B also decreases its average gain per hand.

This implies, that in very stable situations where people behave in a predictable way, bluffing is most profitable. Interestingly, this also means that player B can protect her/himself from player A's bluffs by playing more randomly in order to call player A's bluffs more often, making her/him more reluctant to bluff in the future. If we translate this to real life, if we want to stop someone else's bluff, we should behave in a less predictable way and call the other person's bluff more often. Yet, such behavior may result in smaller gains in the long terms.

[3] The small bets for cards others than `10` are due to the random mutations. I set the mutation rate to a very small value (`0.01`) to minimize these discrepancies.

[4] Note that if the mutation rate is too small (e.g. `0.001`, last series of plots), then the strategies merely respond to one another in a cyclical way. See the lack of continuity across generations between generations `405` and `425` and how homogenous player B's strategies are (indicated by entirely white or black squares on the upper right graphs)

2.3 Other Parameters

I also explored the effect of the other parameters of the GA on player A's level of bluff and found that they do not influence it. Increasing the number of cards or bets only slows down the algorith and makes the patterns need a larger number of generations to appear. The number of strategies in the population also has no impact on player A's level of bluff. Yet, one needs a sufficient amount of strategies in the populations for the algorithm to work. If there are not enough strategies in the population (`20` for instance), then the strategies struggle to evolve after a few generations. Finally, the proportion of strategies that we select to form the set of parent strategies has to be, to my surprise, relatively large. A small value also results in population that struggle to evolve.

3 Looking Forward

There are several elements which we could include in the analysis to improve it.

3.1 Efficiency

We should evaluate which parameters of the GA have the most influence on it speed. This is relatively straightforward to do and I will look into it after the Christmas break.

3.2 Measuring Player A's level of bluff

So far, we have conducted a visual assessment of player A's level of bluff. Yet, it would be beneficial to have a numerical measure for it as it would allow us to conduct statistical test on the results. We could for instance measure the distance between player A's strategies and a strategies where there is no bluff.

3.3 Measuring the Environment's Stability

Similarly, we could also come up with a measure of how stable the environment is from player A's perspective. We could for instance quantify the homogeneity of player B's strategies. Population of similar strategies (resulting in black/white square on the plots instead of grey squares) would be associated with a more stable environment (see last series of plots where `mutation_rate = 0.001` for extremely stable environment).

3.4 Elitism

Note that our version of the GA does not include *elitism*. Elitism refers to the practice of keeping the best parent strategies (the elites) for the following generation. If we set the proportion of elites to, say, `10%`, then the `10%` fittest strategies are kept and the children strategies replace the remaining strategies. Elitism ensures that, regardless of the mutations occuring in the children strategies, we still have good strategies (the elites) in the new population. Elitism makes the GA more efficient.

3.5 Stochastic Strategies

Finally, we could introduce a *stochastic* element in the strategies of the players. For player A, this could take the form of a strategy that gives a density distribution for each card (s)he could receive. Player A's bet then corresponds to a value randomly chosen according to the density distribution. Such strategy could for instance consists of matrix with two columns which provide a mean and a standard deviation for a normal distribution for each possible card. The current version of player A's strategy is a special case of such stochastic strategy where the standard deviations are `0`. As for player B, the strategy could indicate the probability of calling player A's bet given her/his card and player A's bet. The current version of player B's strategy is a special case of such stochastic strategy where the two allowed probabilities are `0` (fold) and `1` (call). Designing strategies containing a stochastic element will be the topic of my next script.