9-2021

Classroom and Computational Investigations of Camel Up

Thomas J. Clark

Classroom and Computational Investigations of Camel Up

Abstract

Camel Up is a popular board game in which players score points by betting on camels which move randomly via a dice mechanic. The game is available both as a board game [1], as well as an IOS App [2]. Because of the random nature of the camels it is generally difficult to play optimally, but one can nevertheless develop various strategies. Probabilistic knowledge proves helpful in assigning relative value to potential game choices. We discuss how this game can be used to motivate and provide context for learning about the concepts of conditional probability and expected value. Also we present R code which can provide exact probabilistic information which is very valuable to players who typically can only get a rough sense of the likelihood of outcomes by observing the board state.

Keywords

Camel Up, strategies, probability, value, learning, outcomes

Disciplines

Mathematics

Comments

This is a pre-print of the article. The published version has a different title. For access to the published version see the following link:

- https://doi.org/10.1080/07468342.2021.1941538

Conditional Probability Done Computationally with Camels

Thomas J. Clark

Thomas J. Clark (tom.clark@dordt.edu) received his PhD in mathematics from the University of Nebraska - Lincoln. He is an assistant professor of mathematics at Dordt University. He is active in scholarship and service in the IBL and Math Circles communities. A former high school teacher, he is committed to improving math education by working to improve the preparation of teachers. He is also an avid board game player, especially enjoying those with interesting mathematical connections.

Summary. *Camel Up* is a popular board game in which players score points by betting on camels which move randomly via a dice mechanic. The game is available both as a board game [**1**], as well as an IOS App [**2**]. Because of the random nature of the camels it is generally difficult to play optimally, but one can nevertheless develop various strategies. Probabilistic knowledge proves helpful in assigning relative value to potential game choices. We discuss how this game can be used to motivate and provide context for learning about the concepts of conditional probability and expected value. Also we present R code which can provide exact probabilistic information which is very valuable to players who typically can only get a rough sense of the likelihood of outcomes by observing the board state.

Introduction

Board games have become increasingly popular and are generating interest among mathematicians. For example many have analyzed the deep geometric beauty in the games SET and Spot It!, see e.g., [**5**] and [**4**]. Recently the game Carcasonne was used as a way to teach probabilistic concepts to undergraduates in [**3**]. When I began my career as a high school mathematics teacher I struggled to teach my AP Statistics students the concepts of conditional probability in a clear and compelling way. Now I believe that by placing concepts into a meaningful context the students would be more motivated and able to develop deep understanding. The recent rise in popularity of board games makes this a novel and fertile area to farm for good problem contexts.

The board game *Camel Up* provides a rich soil to cultivate such problems because it delivers an environment which combines randomness and player choice together in a clever way. While five camels race around the track, often riding on top of each other subject to the whims of dice rolls, the players make bets on which camel will finish first. Earlier bets are more valuable than later bets, so greater risk can lead to greater rewards. The novel aspect of the game is that camels can stack up on each other and ride along if one of the camels below it advances. Because of this, it is possible for camels in the back of the pack to get advanced to the front if the right sequence of rolls occurs - leading to chaotic outcomes. This makes the game interesting but also hard to predict. Players often have to rely on gut instinct about the potential value of available bets. The players also have some limited ways in which they can influence the movement of the camels by playing desert and oasis tokens on the track in lieu of betting on the race. The complete game rules[1] are available online as are brief videos explaining the rules.[2] In brief, most player turns come down to making one of four choices:

1. Roll a die to advance one of the five camels, and score a single point.
2. Place a bet on which camel will be in first place at the end of the current round of camel moves (each camel moves once per round of bets). Possible values are 5, 3, and 2 points for the first, second, and third player respectively to bet on any particular camel. If that camel finishes the round in first place, those points are earned, if that camel finishes in second place, one point is earned, and if that camel finishes in third, fourth, or fifth place, one point is lost.
3. Place a bet on which camel will cross the finish line first (or last). Possible values are 8, 5, 3, and 2 points for the first, second, third, and fourth player to make such a bet; incorrect bets cause the player to lose 1 point at the end of the game.
4. Place a oasis or mirage tile on the track. These tokens score 1 point each time a camel lands on that space during the round. They also are strategically useful to affect the movements of camels to improve the chances of scoring on a previous bet; a camel landing on an oasis tile advances one space and moves back one space on a mirage.

The reason this game lends itself so well to probabilistic question is that camels move via the roll of dice with both the order in which camels move, and the length of movement are random. In addition, as camels can stack up and move together, it is possible for camels far behind to catch up through fortuitous dice rolls; knowing the likelihood of such events is crucial to playing well. Knowing the relative probabilities of various outcomes can be valuable when making decisions about which camels to place a bet on.

To give a sense of how the game plays, consider the following scenario shown in Figure 1. The orange camel is both in the front of the race and also on top of the white camel, so it is the camel most likely to win this round of the race. It also appears the players noticed that as well as both the 5 and 3 bets have been taken on the orange camel. However, it is possible that the orange camel could move first, and the white camel advance past it. It's also possible that the yellow camel could move on top of the orange. But with so many possible outcomes, it is hard to choose. In this situation, the AI player would most likely choose to roll a die to secure a guaranteed point; perhaps that is the best choice for the player as well because how could one possibly know any better? The variety of possibilities is beyond what one can compute in space of a game turn. But the fact of the matter is that the orange camel has a 57% chance of first, and 31% chance of second, while the white camel has a 29% chance of first and a 56% chance of second. Thus the expected value of betting 5 on white 1.86 points, whereas placing a bet of 2 on the orange camel, the more likely outcome, only has an expected value of 1.33 points. Both of these options are more valuable on average than rolling the dice for one point. But how could one compute those probabilities? Enter conditional probability.

[1] See `https://cdn3.trictrac.net/documents/originals/5a/83/87ca6fba0066ee9abf12857873afc0c664a5.pdf` for the *Camel Up* rulebook.

[2] A short 10-minute video introducing the rules of the game is available here: `https://www.youtube.com/watch?v=x3NT3y3Bw8o`.

Figure 1. What is the most valuable move in this situation?

Conditional Probability

Many probability/statistics textbooks provide a similar treatment of conditional probability. First the standard formula is given about the topic, namely:

Theorem 1 (Conditional Probability). *When* $P(A) > 0$, *the* ***conditional probability*** *of* B *given* A *is* $P(B|A) = \dfrac{P(A \text{ and } B)}{P(A)}$.

Some explanation is generally provided and an example is given. For example from [**6**], given a table of data about video game sales including various proportions of various types of games, "what is the conditional probability that a computer game is a strategy game, given that it is not a family or children's game?" Students could probably get this problem correct by noting that there are three pieces of information, find two, and guess them into the right positions without having a deep understanding of conditional probability. Students may also not find the problem set in a meaningful context that helps elucidate the concept.

What problems in the context of the game *Camel Up* do is build intuition about the probability because the condition is temporal in nature. Thus the order of the actions is the condition of the actions. Consider the scenario in Figure 2. The only camels left to move are orange and yellow; the blue camel is in the lead. Since the orange camel is on the bottom and can only move up to three spaces, it cannot win, so either the blue camel will win or the yellow camel. For yellow to win, it must first hitch a ride on the orange camel and then advance a total of at least four spaces. The question "what is the probability that yellow wins" is one of conditional probability. It is a concrete question that requires thinking, and the thinking is actually valuable in making sense of the answer. The question is equivalent to $P(A \text{ and } B)$ where A is orange goes first, and B is the sum of the two dice rolls is four or more. There is a $1/2$ chance that orange will move first. Assuming that, there are nine possible dice combinations, $(1,1)$, $(1,2)$, $(1,3)$, $(2,1)$, $(2,2)$, $(2,3)$, $(3,1)$, $(3,2)$, and $(3,3)$; six of the nine sum to four or more, and thus there is a $\frac{1}{2} \cdot \frac{6}{9} = \frac{1}{3}$ probability that yellow wins. The probability that blue wins is thus $\frac{2}{3}$. Surprisingly in this scenario, betting 5 on yellow is more valuable than betting 2 on blue on average. I wonder if Las Vegas will consider setting up *Camel Up* tables in their casinos.

Figure 2. Only two camels left to move, which bet is best?

In Game Examples

In the previous example, the probability could be computed by hand by thinking about the problem. What happens when this is impossible? Well, impossible for a human but not for a computer. Consider the *Camel Up* board state early in the game as shown in Figure 3. Often early in a round, the computer AI elects to roll the die, earning a guaranteed one point. However in this scenario, had the AI chosen to take the 5 point bet on the orange camel, the expected value of the move would have been $1.65 = 5 \cdot 0.361 + 1 \cdot 0.244 - 1 \cdot 0.395$ points. But who could possibly do such a computation on the fly during the game?

Likely no human could do the computation in a short time, but a computer certainly could with the right program. I wrote an *R* code that does just that. It computes the probability of orange placing first as 0.361, second as 0.244, and third, fourth, or fifth as 0.395, from which the expected value calculation can be done. One might argue that it is "better" strategy to take guaranteed points over risky ones, but that strategy proves to be incorrect in the long run on average. Using the *R* code to simulate the possible game outcomes, I played several games against the most difficult AI player. In each game, by consistently making the move with the highest expected value I could on average score more points in each leg of the race, and over time build up a significant lead by the end of the game. In this case, by rolling the dice, the AI gave me first pick of the bets with only four dice remaining. Because by chance the orange camel rolled before yellow and green, it did not get to be advanced as far, and now the green camel is most likely to win, I could place a bet on green and expect

$1.758 = 5 \cdot 0.387 + 1 \cdot 0.217 - 1 \cdot 0.396$ points. After this choice, no other bets, are more valuable than rolling the dice which is what the AI player will likely do. This will give me the first choice at bets with only three dice remaining. The fewer the dice that remain, the less risk there is in making bets as the possible outcomes are narrowed. Nevertheless, exact probabilistic information can direct the player to make marginally better choices over the whole game, leading to a win in the end.

Figure 3. An early game state against one AI player.

As the game progresses, we find another interesting board state in Figure 4. It may appear as if the green camel is certain to win this leg of the race, but since the blue camel has yet to go, there is a chance the white camel riding on it may be the winner. On the other hand, the yellow camel could potentially get onto the white camel, and then be carried into the lead. So what is the best move here? In this case, a combination of conditional probability and expected values will help sort out the situation because there are only two camels left to go, the math is straight-forward.

- Green wins with probability $\frac{1}{3}$, if blue rolls a 1. Expected value of a green bet is $0 = 2 \cdot 0.333 + 1 \cdot 0 - 1 \cdot 0.667$.
- White wins if blue rolls a two or three, and does not have the yellow camel on its back. If yellow goes first and rolls a two (using the oasis) or a three, it will land on the back of the white camel. Thus the probability of the white camel winning is the sum of two probabilities, either the blue camel goes first and rolls a two or three, or the yellow does, but rolls a 1, and the blue rolls two or three. Specifically, $\frac{4}{9} = \frac{1}{2} \cdot \frac{2}{3} + \frac{1}{2} \cdot \frac{1}{3} \cdot \frac{2}{3}$.
- The yellow camel finishes in first place if it rolls a two or three, before blue, and has the blue camel roll a two or three. Namely, $\frac{2}{9} = \frac{1}{2} \cdot \frac{2}{3} \cdot \frac{2}{3}$.

Computing these probabilities proves to be an interesting question, that is motivated by the game, and requires logical thinking. Moments like these arise throughout the game providing excellent opportunities for students to compute probabilities in a meaningful setting. Turning probabilities into expected values adds another layer of valuable mathematics. It is an exercise for the reader to determine the probabilities for each camel to end in second place. With this information in hand we can compute the value of betting 5 on the white camel to be $2.55 \approx \frac{23}{9} = 5 \cdot \frac{4}{9} + 1 \cdot \frac{4}{9} - 1 \cdot \frac{1}{9}$, and the value of betting 5 on the yellow camel to be $0.55 \approx \frac{5}{9} = 5 \cdot \frac{2}{9} + 1 \cdot \frac{1}{9} - 1 \cdot \frac{2}{3}$. In this case, betting, 5, 3, or even 2 on white all on average deliver a better return(2.55, 1.66, 1.22 respectively) than rolling the dice in this case. No other bets do.

Figure 4. An interesting mid game state against one AI player.

The end of the game proves to be the most interesting. One complication is that one must weigh the value of betting on a camel to win the leg with the value of betting on a camel to finish the race. Often times there is a chance a camel may finish the race in the current leg, but it is not guaranteed. When a leg finishes and all camels can move again, the probabilities tend to shift significantly. It would be incredibly valuable to have a late game function that can track the likelihood of a particular camel finishing the race not only on the current leg, but the following leg as well. However, a full leg involves $5! \cdot 3^5 = 29,160$ computations; to complete a new 29,160 computations for each possible outcome of the current leg would take too long to be practicable.

In view of this, it is difficult to have a complete end game strategy based on maximizing of the expected value of each action, though a flexible strategy that incorporates the available probabilistic information is nevertheless valuable. Consider the situation shown in Figure 5. In this case one can compute some probabilities by hand. One could propose the question to students what is the best choice here, to make an end of game bet on the blue camel, a lap bet on the yellow camel, an end of game bet on the yellow camel, or just roll the dice for one point? How does it depend on what bets are still available?

Figure 5. An interesting late game state against one AI player.

R code

In order to compute probabilities as specialized R code is needed to loop through the possible outcomes. This code is available from the author. The code is relatively easy to use. After loading the functions into memory, the board state is entered and the master function is called. Each camel's position and "height" is entered in as an ordered pair. The location of an oasis or mirage is included as well when applicable, and the camels yet to move are indicated. A screenshot is included in Figure 6 with the output shown in Figure 7. Note that the code produces both probabilities and expected value information as well as the likelihood of which spots may be landed on (valuable for placing and oasis or mirage effectively).

To test whether the information about expected values is actually valuable during the game, I used the *R* code to play "optimally" against a single AI player (on hardest difficulty) using the iOS version of the game [**2**]. In all I played 12 games, where I fastidiously tracked every action taken by the AI player and me and the expected value of that action at the time it was taken. I also tracked the end of leg scores for each of the 57 legs contained in those 12 games. The average expected score per leg for me was 9.52 points per leg; the average actual score per leg was 9.47. Which makes sense, since the dice rolls are random the points scored are close on average to the expected points on average. The corresponding values for the AI player were 6.91 points expected and 6.98 points scored per leg on average over 57 legs. This is very encouraging because it shows that playing with more probabilistic information allows the player to make better decisions, even better than the AI, which leads to nearly 2.5 more points per leg. In fact I beat the AI player in every game, and the average expected win difference was 12.3 points and the average actual win difference was 11.75 points. Looking at matched pairs data for the difference in score for each round yields and expected average difference of 2.61 points with a sample standard deviation of 2.10 points and actual average difference per leg of 2.49 with a sample standard deviation of 2.97 points.

One caveat that must be added to the previous paragraph is that at the end of the game, one can also bet on which camel will win not the leg, but the whole race. A shortcoming of the code is that it focuses on a single lap, it doesn't compute which camel is likely to finish the race because those probabilities require simulating future legs as well. While it may be interesting and valuable to project ahead beyond the current lap, the time needed to do such a computation is practically prohibitive. It also does not add anything new in terms of pedagogical significance, that is, the computations are typically far too difficult to be done by hand and so they don't provide new education opportunities. Though this certainly could be a direction for future work on this game. That said, any time either player made an "end of game bet" I set the "value" of such a bet to be 0. On average both players make a similar number of end of game bets, and typically only in the last leg of each game, so it has minimal outcome on the expected value calculations done in the previous paragraph. The true end of game scores were on average 14.25 points higher for me over the AI. So, apparently my end of game strategy was slightly better than the AIs improving my average win differential by 2.5 points.

```
####### Camel Up Run Files
C1B = c(4,1)
C2G = c(8,1)
C3O = c(8,3)
C4W = c(8,2)
C5Y = c(6,1)
OSMR = matrix(0,1,19)  # location of oasis ( 0 = none, 1 = oasis, 2 = mirage)
########## OSMR[N] = 1 or 2  if oasis or mirage in postion N
#OSMR[9] = 1  # would set an oasis at 9
#OSMR[11] = 2 # would set a mirage at 11
###### 1B 2G 3O 4W 5Y   ###########################  1B 2G 3O 4W 5Y
ToGo = c(1,3,4,5) # Omit which camels have gone already from c(1, 2, 3, 4, 5)
CamelMaster(C1B, C2G, C3O, C4W, C5Y, OSMR, ToGo)
```

Figure 6. R code used to compute probabilities.

```
> CamelMaster(C1B, C2G, C3O, C4W, C5Y, OSMR, ToGo)
# First -   Blue     Green   Orange    White     Yellow
           [,1] [,2]      [,3]      [,4]      [,5]
[1,] 0.01697531    0 0.5740741 0.2901235 0.1188272
# Second -  Blue     Green  Orange    White     Yellow
           [,1] [,2]      [,3]      [,4]      [,5]
[1,] 0.02006173    0 0.3055556 0.5601852 0.1141975
#3rd-5th - Blue     Green   Orange    White     Yellow
         [,1] [,2]      [,3]      [,4]      [,5]
[1,] 0.962963    1 0.1203704 0.1496914 0.7669753
Which Spots Landed On
     [,1] [,2] [,3] [,4]      [,5]      [,6]      [,7]      [,8]      [,9]     [,10]     [,11]     [,12]
[1,]    0    0    0    0 0.3333333 0.3333333 0.6666667 0.3333333 0.8333333 0.5555556 0.6111111 0.1666667
         [,13]      [,14] [,15] [,16] [,17] [,18] [,19]
[1,] 0.1111111 0.05555556     0     0     0     0     0
  Bets      Blue1 Green2  Orange3    White4      Yellow5
1    5 -0.8580247     -1 3.055556 1.8611111 -0.05864198
2    3 -0.8919753     -1 1.907407 1.2808642 -0.29629630
3    2 -0.9089506     -1 1.333333 0.9907407 -0.41512346
> 
```

Figure 7. R code output with probabilities and expected values.

Conclusion

Playing board games is becoming increasingly popular and many games have led to deeply interesting mathematics. In this case, the mathematics involved lives at the intersection of conditional probability, expected value, and numerical computations. *Camel Up* delivers a wonderful context for all these problems to be explored deeply from simple to more complex scenarios. This makes the game well-suited for use in the classroom. If instead you just want to feel superior to the AI or your friends in a fun board game, the *R* code will deliver spectacular results.

References

1. Steffen Bogen. *Camel Up.* Z-Man Games, 2014. Board Game.
2. Camel Up. (2015). Outline Development (Version 1.1.0) [Mobile application software]. Retrieved from `http://itunes.apple.com`
3. M. Capaldi and T. Kolba, Carcassonne in the Classroom, *College Math. J.* **48:4** (2017) 265-273.
4. T. Clark, Spot It! Mathematical Structure in a Children's Game." MTCircular Winter/Spring 2017, pp. 10-12.
5. T. Goldberg. Algebra From Geometry in the Card Game SET. *College Math. J.* **47:4**, (2016) 265-273.
6. D. Moore, *The Basic Practice of Statistics*. Fourth Edition. W.H. Freeman, New York, 2007.