

Math 23A R term project

Maggie Ji and Emmet Murphy

12/7/2017

War Simulations: From Card Game to Korea

Overview

The first part of this paper generates a Markov process to predict the probability of a hand winning in the War card game.

The second part turns toward a more serious application: Using Bayesian updating to model the outcomes of a real war, such as with North Korea.

Generating a Markov Process

Our goal is to predict the probability of a hand winning in n number of plays, given x number of cards in the hand. We accomplish it in three steps:

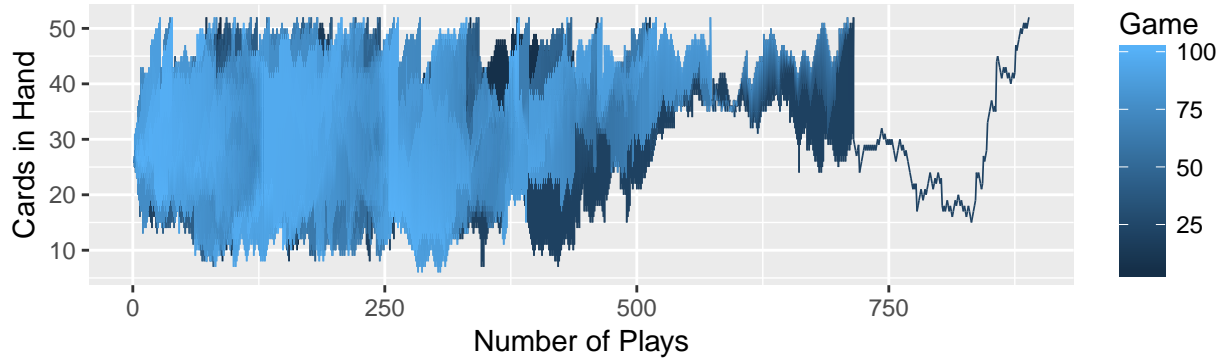
1. Simulate games, collect and explore the data
2. Model the probabilities as a Markov Chain
3. Predict a hand's probability of winning

Part 1: Simulate and Explore the Data

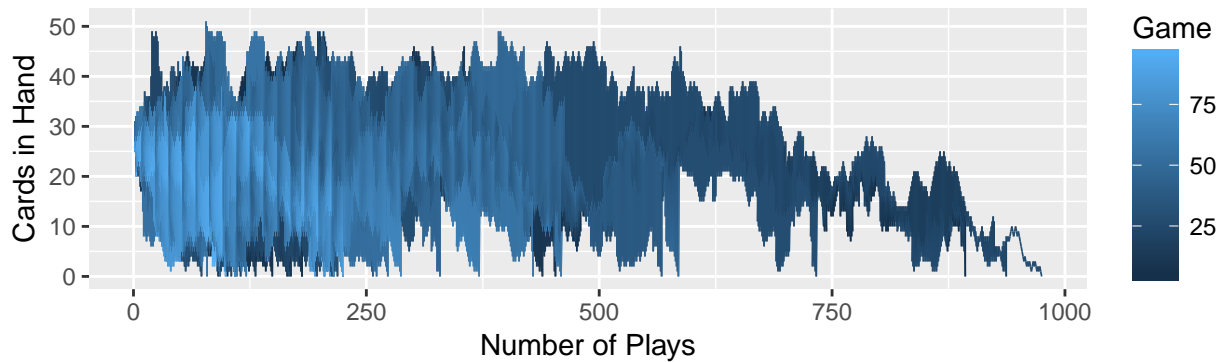
In order to train our Markov process, we first simulate 10,000 games of War. Details on that are in the Appendix. The simulations generate and store data on starting hand strength, the number of cards in a player's hand after each play, etc. The following script reads in the data and plots the number of cards by the number of plays. The plots give a sense of the dispersion of outcomes.

```
# plot number of cards in hand over the course of games
deck_sizes <- read_csv("no-ties/deckSizes100random.csv")
games <- read_csv("no-ties/games100random.csv")
gamePlays = merge(deck_sizes, games, by = 'id')
gamePlays$cards = gamePlays$p1Cards
gamePlays$Game = gamePlays$id
wins <- gamePlays[gamePlays$result=='W',]
losses <- gamePlays[gamePlays$result=='L',]
ties <- gamePlays[gamePlays$result=='T',]
p1 <- ggplot(data=wins, aes(x=play, y=cards)) + geom_line(aes(color=Game), size=0.3) +
  ggtitle(paste("Plays per Game in", length(unique(wins[, 'id'])), "Wins")) +
  xlab('Number of Plays') + ylab("Cards in Hand")
p2 <- ggplot(data=losses, aes(x=play, y=cards)) + geom_line(aes(color=Game), size=0.3) +
  ggtitle(paste("Plays per Game in", length(unique(losses[, 'id'])), "Losses")) +
  xlab('Number of Plays') + ylab("Cards in Hand")
grid.arrange(p1, p2, ncol=1)
```

Plays per Game in 53 Wins



Plays per Game in 47 Losses



To a player it can seem like a game of War will go on forever (as discussed in more detail in the Appendix). The following summarizes how many plays our simulated games require in order to converge on a winner:

```
summary(games$plays)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      26.0  121.5   211.0   275.7  372.5   974.0
```

Part 2: Generate a Markov Process

The Markov chain calls for a square matrix M in $\mathbb{R}^{\mathbb{N} \times \mathbb{N}}$ (there are 52 cards and a +1 for the possibility of a hand ending with 0 cards). Each cell in row i , column j has a probability of a hand going from $i - 1$ number of cards to $j - 1$ number of cards.

The probability of a hand with x cards winning in n number of plays is then:

$$M^n e_x$$

Following is the script to generate the Markov matrix M :

```
# generate Markov matrix
# for probabilities, this is the denominator -- frequency of each hand size
handCountFreq <- plyr::count(gamePlays, vars='cards')

# the numerator is trickier -- need to count how many times a hand size follows any given hand size
gamePlays$nextCards <- lead(gamePlays$cards, 1)
counts <- dplyr::ddply(gamePlays, .(gamePlays$cards, gamePlays$nextCards))
names(counts) <- c("cards", "nextCards", "Freq")
```

```

# The probability of transitioning from column p to column q
probTransit <- function(p, q) {
  # Columns p, q correspond to card counts p-1, q-1
  x <- p - 1; y <- q - 1
  numerator <- counts[counts$cards == x & counts$nextCards == y,]
  denominator <- handCountFreq[handCountFreq$cards == x,]$freq
  prob <- (nrow(numerator) / denominator)
  if (is.na(prob) || is.nan(prob)) {
    prob <- 0
  }
  return (prob)
}

M <- matrix(0L, nrow=53, ncol=53)
# Fill the transitions to 0 and 52 cards outside the main loop, since the
# transitions are only one-way
for (j in 2:52) {
  M[1,j] <- probTransit(j, 1)
  M[53,j] <- probTransit(j, 53)
}
# When have 0 or 52 cards, probability of keeping that many is 1
M[1,1] = 1.0
M[53,53] = 1.0

# populate the matrix
# value in cell i,j is the likelihood of moving from i to j
for (i in 2:52) {
  for (j in 2:52) {
    if (i == j) {
      # we resolve ties within a play, so the number of cards always changes
      # between plays (i.e. probability of staying on a card count is 0)
      M[i,j] <- 0.0
    } else {
      M[i,j] <- probTransit(i, j)
    }
  }
}

```

Part 3: Prediction

Generating the Markov matrix is the difficult part. Predicting a hand's probability of winning can then be coded as a function that implements the matrix multiplication $M^n \vec{e}_x$.

```

# predict the probability of a hand with x number of cards winning after n plays
winProbability <- function(M, numCards, numPlays) {
  # create a vector for the starting position
  v <- numeric(53)
  v[numCards + 1] = 1

  # multiply by the Markov Matrix to get the probability of winning
  for (i in 1:numPlays) {
    M <- M %*% M
  }
}

```

```

Mprob <- M %*% v
return (Mprob[53])
}
data.frame(OneIn10 = winProbability(M, 1,10), FiftyOneInNext=winProbability(M,51,1),
           TwentySixIn20=winProbability(M,26,20))

```

```

##      OneIn10 FiftyOneInNext TwentySixIn20
## 1 0.02061325      0.4886364      0.2449805

```

The sample values show the probability of starting with:

- 1 card and winning in 10 plays
- 51 cards and winning on the very next play
- 26 cards and winning in the next 20 plays

Further research would be to incorporate hand strength when calculating the probability (a non-trivial improvement, as it calls for extending the model to include conditional probabilities).

Real World Application: Bayesian Updating in a State of War

So far, our simulations of the card game have assumed that the deck held by each player is randomly drawn, so that neither player knows (1) whether the decks are unequal; and (2) who has the better deck. This begs the question: is it possible to use the information provided by the already revealed cards to make inferences about the strength of a player's deck? *POINT: Includes two related but distinct topics.*

Indeed, in real-world situations that are analogous to this simple card game, a key first step is to assess the position of your opponent, relative to your own position. *POINT: Project is related to a topic that you have studied in another course this term: API-302 Analytic Frameworks for Policy.* Take the case of the U.S. and North Korea as an example. In the face of escalating tensions, each side is desperately trying to figure out what “deck of cards” the other holds, in terms of nuclear capabilities. Although there is not officially a state of war, there are sufficiently many small-scale movements by each side, through which information can be collected. Repeated interactions between the U.S. and North Korea, in the form of diplomatic relations, missile testing, and military demonstrations, could reveal much about each country's true capabilities.

For ease of demonstration, we will make the simplifying assumption that there is no bluffing or strategic omission of information involved on either side (which is admittedly unrealistic). Therefore, the outcome of each interaction is purely based on a random draw from each country's true distribution of military strength (i.e. their “deck of cards”).

How Bayesian Updating Works

To start, let's assume that the U.S. holds an infinite arsenal of weapons ranging in power from 1 to 10. North Korea could either have a stronger arsenal, of weapons ranging between 6 and 10 in power, or a weaker arsenal, of weapons ranging between 1 and 5. Through repeated interactions in the form of small interactions, the U.S. can use Bayesian updating to inform its perception of the true range of North Korea's nuclear arsenal.

R CODE: CONDITIONAL PROBABILITIES

As shown by these permutation tests:

- If North Korea has a relatively weaker arsenal, then the U.S. would win 70 percent of the small-scale conflicts, lose 20 percent of the time, and tie with North Korea during the remaining 10 percent of conflicts: $p(\text{win}|\text{weaker}) = 0.7, p(\text{lose}|\text{weaker}) = 0.2, p(\text{tie}|\text{weaker}) = 0.1$
- However, if North Korea has a relatively stronger arsenal, then the probabilities are reversed: $p(\text{win}|\text{stronger}) = 0.2, p(\text{lose}|\text{stronger}) = 0.7, p(\text{tie}|\text{stronger}) = 0.1$

Say that the U.S. starts with no knowledge of North Korea's true strength, such that it is equally possible that it has a stronger arsenal or a weaker arsenal. In other words, the U.S. holds the prior that $p(\text{weaker}) = 0.5$ and $p(\text{stronger}) = 1 - p(\text{weaker}) = 0.5$. (Note here that we can focus on only $p(\text{weaker})$ without loss of information, since $p(\text{stronger}) = 1 - p(\text{weaker})$.)

If a small military conflict erupts in the DMZ between U.S. troops and North Korean forces, then the U.S. can update its perception of North Korea's likely strength, based on the outcome of that conflict: (1) If the U.S. wins, then $p(\text{weaker}|\text{win})$ becomes the new prior, rather than 0.5: $p(\text{weaker}|\text{win}) = \frac{p(\text{win}|\text{weaker})p(\text{weaker})}{p(\text{win})}$

$$= \frac{p(\text{win}|\text{weaker})p(\text{weaker})}{p(\text{win}|\text{weaker})p(\text{weaker}) + p(\text{win}|\text{stronger})p(\text{stronger})} = \frac{0.7*0.5}{0.7*0.5 + 0.2*0.5}$$

$$(2) \text{ If the U.S. loses, then } p(\text{weaker}|\text{lose}) \text{ becomes the new prior, rather than 0.5: } p(\text{weaker}|\text{lose}) = \frac{p(\text{lose}|\text{weaker})p(\text{weaker})}{p(\text{lose})} = \frac{p(\text{lose}|\text{weaker})p(\text{weaker})}{p(\text{lose}|\text{weaker})p(\text{weaker}) + p(\text{lose}|\text{stronger})p(\text{stronger})} = \frac{0.2*0.5}{0.2*0.5 + 0.7*0.5}$$

$$(3) \text{ If there is a tie, then } p(\text{weaker}|\text{tie}) \text{ becomes the new prior (in this particular case, it is still 0.5): } p(\text{weaker}|\text{tie}) = \frac{p(\text{tie}|\text{weaker})p(\text{weaker})}{p(\text{tie})} = \frac{p(\text{tie}|\text{weaker})p(\text{weaker})}{p(\text{tie}|\text{weaker})p(\text{weaker}) + p(\text{tie}|\text{stronger})p(\text{stronger})} = \frac{0.1*0.5}{0.1*0.5 + 0.1*0.5}$$

To generalize, we can define a sequence of updated priors as: $P = \{p_1, p_2, p_3, \dots\} = \{p_n | n \in \mathbb{R}\}$ where n denotes the n th small-scale conflict. This is a somewhat complicated sequence, where $p_n = f(p_{n-1})$ is a different function for each possible outcome: $p_n = \frac{0.7p_{n-1}}{0.7p_{n-1} + 0.2(1-p_{n-1})}$ if the U.S. wins in the n th conflict; $p_n = \frac{0.2p_{n-1}}{0.2p_{n-1} + 0.7(1-p_{n-1})}$ if the U.S. loses in the n th conflict; and $p_n = \frac{0.1p_{n-1}}{0.1p_{n-1} + 0.1(1-p_{n-1})}$ if there is a tie in the n th conflict.

R CODE: SINGLE SIMULATION OF BAYESIAN UPDATING

In this example, a sequence of randomly drawn outcomes is simulated based on underlying "true" distributions where the U.S. has a stronger arsenal than North Korea. Here, the sequence of updated Bayesian probabilities are as follows:

$$p_0 = 0.0000000$$

$$p_1 = 0.7777778$$

$$p_2 = 0.9245283$$

$$p_3 = 0.9772080$$

$$p_4 = 0.9933802$$

$$p_5 = 0.9980996$$

...

$$p_{14} = 1.0000000$$

...

Convergence to Underlying Distribution

In the previous simulation, the sequence P of Bayesian updated probabilities converges to 1 around the 14th trial, reflecting the true underlying distribution (i.e. that the U.S. has a stronger arsenal). Is this always the case?

Let us represent the process of Bayesian updating using terminology from Math 23A. *POINT: Incorporates ideas both from linear algebra and from real analysis.* The updating function depends on the outcome of each interaction, such that it can be represented as:

$$p_n = [\vec{\mathbf{B}}] \vec{v}_n(p_{n-1})$$

where $\vec{\mathbf{B}} = [f_{\text{win}} \quad f_{\text{lose}} \quad f_{\text{tie}}] = \left[\frac{0.7*p}{0.7*p + 0.2*(1-p)} \quad \frac{0.2*p}{0.2*p + 0.7*(1-p)} \quad \frac{0.1*p}{0.1*p + 0.1*(1-p)} \right]$ and $\vec{v}_n = \begin{bmatrix} \text{win} \\ \text{lose} \\ \text{tie} \end{bmatrix}$ represents the outcome of each interaction.

For example, the outcome where the U.S. wins would be represented as $\vec{v}_n = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$, with the updating function

$$[\vec{\mathbf{B}}]\vec{v}_n(p_{n-1}) = \begin{bmatrix} f_{win} & f_{lose} & f_{tie} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = f_{win} = \frac{0.7 * p_{n-1}}{0.7 * p_{n-1} + 0.2 * (1 - p_{n-1})}.$$

In general, after k number of interactions, the updated prior would be some function of the initial starting prior p_0 , depending on how the sequence of outcomes turns out from the interactions:

$$p_k = [\vec{\mathbf{B}}]\vec{v}_k[\vec{\mathbf{B}}]\vec{v}_{k-1} \dots [\vec{\mathbf{B}}]\vec{v}_1(p_0)$$

There are two useful properties of this sequence, that make Bayesian updating useful:

- (A) The marginal value of information diminishes with each interaction, such that there is an optimal stopping point after a finite number of interactions:

$$\lim_{n \rightarrow \infty} p_n - p_{n-1} = \lim_{n \rightarrow \infty} ([\vec{\mathbf{B}}]\vec{v}_n \dots [\vec{\mathbf{B}}]\vec{v}_1(p_0)) - ([\vec{\mathbf{B}}]\vec{v}_{n-1} \dots [\vec{\mathbf{B}}]\vec{v}_1(p_0)) = 0$$

$$\Rightarrow \exists k < \infty \text{ s.t. } [\vec{\mathbf{B}}]\vec{v}_n \dots [\vec{\mathbf{B}}]\vec{v}_1(p_0) \approx [\vec{\mathbf{B}}]\vec{v}_{n-1} \dots [\vec{\mathbf{B}}]\vec{v}_1 \Rightarrow [\vec{\mathbf{B}}]\vec{v}_n \approx \mathbf{I}.$$

- (B) The sequence converges to a certainty (i.e. either $p = 0$ or $p = 1$) that reflects the true underlying difference in the distribution of military weapons between the two countries.

$$\lim_{n \rightarrow \infty} p_n = s, s \in 0, 1$$

These two properties can be observed using the following R code, where we simulate different p_n sequences that draw from a range of underlying differences between the distribution of weapons possessed by each side.

R CODE: MULTIPLE SIMULATIONS OF BAYESIAN UPDATING

These plots illustrate a key takeaway that, with underlying distributions that closely resemble each other (i.e. the U.S. and North Korea are similarly matched), it requires a higher number of n interactions for the sequence of posteriors to converge to certainty. In real life, since such interactions and confrontations are risky and costly, it makes sense to minimize n . As much information as possible should be extracted from the resulting p_n , not only in terms of whether p_n is closer to 0 or to 1 (which informs which country is stronger), but also how far it is from 0 or 1 (the farther away it is, the more likely that the sequence would take lots of interactions to converge, suggesting that the two countries are more evenly matched).

Appendix

Simulating the War Card Game

We are not the first to undertake this exercise, and we started from the R source available from PremierSoccerStats blog. But we soon ran into a problem, that we addressed in two ways.