The Problem of Coincidences

JOHN DICKERSON

15-859D

The Gambling Scholar

- Gerolamo Cardano, Italian Renaissance physician and mathematician, on gambling:

  "...In times of great anxiety and grief, it is considered to be not only allowable, but even beneficial."

- Written in his *Liber de ludo aleae*—"Book on Games of Chance"—an early systematic treatment of probability.

A Simple Game of Cards

- Take a French deck of 52 cards:
  - 4 suits (spades, hearts, diamonds, clubs)
  - 13 ranks, {A=1, 2, …, 10, J=11, Q=12, K=13}
- One card at a time:
  - Player picks number x in {1,…,13}
  - Turn over card
- If the card matches the player's card, loses.
- If the player survives *all* (13) rounds, wins.
- What is the probability of winning?

The Game of Thirteen

- This is called the "Game of Thirteen"
  - Proposed by Pierre Montmort in 1708.

- Lots of famous people involved:
  - Solved by Montmort in 1710—in the non-general case—in a letter to Johann Bernoulli.
  - Nikolaus Bernoulli solved the general case in 1711.
  - Leonard Euler provides an alternate proof in 1751.
  - Eugène Catalan prove results for a similar game in 1837.
  - …
  - Mathematical Games-2010 has elegant, open discussion!

1. Simplest Model

- All cards the same suit.
- Cards {1,…,n} in deck.
- Cards drawn without replacement.

- A coincidence occurs when the i^{th} card has value i.

- *P(n,k)* – probability that n drawings yields at least k coincidences.

Examples: $P(n,k)$

- $D(n)$ – number of permutations with at least 1 coincidence
- $Q(n)$ – number of permutations with 0 coincidences

- $P(n,1) = D(n) / n!$
  - "Probability that n drawings yields at least 1 coincidence."

- $1 - P(n,1) = Q(n) / n!$
  - "Probability that n drawings yields no coincidences."

Calculating $Q(n)$

- Let A_i be the event of a coincidence on the i^{th} drawing
- ξ_n is total number of coincidences
  - $\xi_0 = 0$

- $P(\xi_n = k) = \binom{n}{k} Q(n-k) / n!$
  - Define $Q(0) = 1$.
  - Find the distribution of ξ_n – just find all the $Q(i)$s

- We'll create a recurrence relation to do this.

Calculating *Q(n)*

- We know: $\sum_{k=0}^{n} P(\xi_n = k) = 1$

- Expanding the equation from last slide:
  - $\sum_{k=0}^{n} \binom{n}{k} Q(n-k) \,/\, n! = 1$
  - $\sum_{k=0}^{n} Q(n-k) \,/\, k!(n-k)! = 1$
  - $\frac{Q(n)}{n!} = 1 - \sum_{k=1}^{n} \frac{Q(n-k)}{k!(n-k)!}$

- Recurrence form, but not so easy to compute
  - Especially by hand in the 1700s . . .

Calculating *Q(n)*

- Recall geometric & Taylor series:
  - $e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!}$ and $\frac{1}{1-x} = \sum_{n=0}^{\infty} x^n$ for |x|<1
- From last slide, have $\frac{Q(n)}{n!} = 1 - \sum_{k=1}^{n} \frac{Q(n-k)}{k!(n-k)!}$
  - $\sum_{n=0}^{\infty} \frac{Q(n)}{n!} x^n = e^{-x}/_{1-x}$
  - $e^{-x}/_{1-x} = \sum_{n=0}^{\infty} \left(\left[1 - \sum_{k=1}^{n} \frac{Q(n-k)}{k!(n-k)!} \right] x^n \right)$
- Remove the geometric series, get explicit form:
  - $\frac{Q(n)}{n!} = \sum_{k=0}^{n} \frac{(-1)^k}{k!}$ the first n terms of e^{-1}
  - Still, not so easy to compute . . .

Calculating $P(\xi_n = k)$ from $Q(n)$

- [Jordan 1867]: $A_1, A_2, \ldots, A_n$ random events then
  - $P(\xi_n = k) = \sum_{r=k}^{n} \binom{r}{k} (-1)^{r-k} B_r$ for $0 \le k \le n$
  - $P(\xi_n \ge k) = \sum_{r=k}^{n} \binom{r-1}{k-1} (-1)^{r-k} B_r$ for $1 \le k \le n$
- $B_r = \sum_{1 \le i_1 < \ldots < i_r \le n} P(A_{i_1}, A_{i_2}, \ldots, A_{i_r})$
  - So, e.g., B_2 is the sum of the probabilities of 2 events firing
  - Generally, this is the binomial moment of ξ_n:
    - $B_r = \mathrm{E}\left\{\binom{\xi_n}{r}\right\}$

- We can use this to get a better explicit form.

Calculating $P(\xi_n = k)$ from $Q(n)$

- For our game:
  - $P\left(A_{i_1}, A_{i_2}, \cdots, A_{i_r}\right) = \frac{(n-r)!}{n!}$
  - For example, when r=1, there's a $\frac{1}{n}$ chance that I choose value *i* when the banker shows value *i*
- Summing, yields $B_r = \frac{1}{r!}$
- Plug in, simplify from last slide:
  - $P(\xi_n = k) = \frac{1}{(k-1)!} \sum_{r=k}^{n} \frac{(-1)^{r-k}}{r(r-k)!}$ (1)

Calculating $P(\xi_n = k)$ from $Q(n)$

- But remember!
  - $P(n,k) = P(\xi_n \geq k) = \binom{n}{k}\frac{Q(n-k)}{n!}$ (2)
  - $1 - P(n,1) = \frac{Q(n)}{n!}$
- We derived:
  - $\frac{Q(n)}{n!} = \sum_{k=0}^{n}\frac{(-1)^k}{k!}$ (3)
  - $\lim_{n\to\infty}\frac{Q(n)}{n!} = e^{-1}$ by expanding to full Taylor series
- Combine (1), (2), and (3) to get:
  - $\lim_{n\to\infty} P(\xi_n = k) = \frac{e^{-1}}{k!}$

Summary of Simple Model Results

- We can now quickly (kind of) compute the probability of winning in the general case:
  - $1 - P(n,1) = \sum_{k=0}^{n} \frac{(-1)^k}{k!}$
- We also came up with a neat result:
  - The probability of exactly k coincidences converges to $\frac{e^{-1}}{k!}$
- In short, don’t play: you’ll win ~36.79% of the time.

2. Slightly Less Simple Model

- The last model did not take suits into account
- Let the deck contain *s* sets of cards
  - Each set has cards with values {1,…,n}.
  - Only *n* cards drawn without replacement.
- Total number of possible outcomes:
  - $ns(ns-1)(ns-2)\cdots(ns-n+1)$

- *P(n,s,k)* – probability that at least *k* coincidences occur
- Montmort's game: solve P(13,4,1)
- Simplified model from before: solve P(13,1,1) = P(13,1)

Calculating $P(\xi_n(s) = k)$

- Let A_i be the event of a coincidence on the i^{th} drawing
- $\xi_n(s)$ is total number of coincidences for s-suit game
  - $\xi_o(s) = o$

- Note for this model:
  - $P\left(A_{i_1}, A_{i_2}, \cdots, A_{i_r}\right) = \frac{s^r}{ns(ns-1)(ns-2)\cdots(ns-r+1)}$

- Can use similar techniques (using [Jordan 1867]) to determine:
  - $\lim_{n\to\infty} P(\xi_n(s) = k) = \frac{e^{-1}}{k!}$ for any s = 1,2,3,...

Don’t Play this Game, Either

- Intuitively, including suits doesn’t really matter.
- Again, don’t play—even if you have an arbitrarily large deck of cards:
  - You’ll still win at most ~36.79% of the time!

How Did They Do It?

- We used techniques from 1867+ to solve these games!
  - Bernoulli, Montmort, Euler, and Catalan did not have these techniques.

- Read Lajos Tákacs' *The Problem of Coincidences* for the story of how this problem was solved . . .
  - . . . and how studying it advanced probability theory!
  - Both N. Bernoulli and De Moivre showed specialized versions of the general results of [Jordan 1867] nearly 150 years earlier.

Skip to the 21st Century . . .

- Game Theory exists now, for better or for worse
- The earlier results tell us, given optimal play, the limit of the probability of winning:
  - But how do we play optimally in more general games?
- First known treatment in [Litwack *et al*. 2008].
  - Analyzes the game "Dundee"

3. A More General Model

- Cards in the deck assume values in $\{1,2,\ldots,v\}$.
- Let s_i be the number of cards with value i.
  - E.g., if $s = (4,4,\ldots,4)$, with thirteen 4s, we have a standard deck.
- Let m be the number of rounds.
  - Must have: $m \leq s_1 + \ldots + s_v$
- Same win/loss model:
  - If any coincidence occurs across m rounds, player loses.
  - If no coincidences occur, player wins.

Two Types of Strategies

- Adaptive strategy:
  - The player's bid at round i can depend on values that appeared in previous rounds.
- Advance strategy:
  - The player's bid at round i cannot depend on values that appeared in previous rounds; it is chosen in advance.
  - This is the strategy used in the Montmort's game.

Greedy Adaptive Strategy

- Adaptive—the player knows what cards have been dealt, and can adjust his strategy in real-time:
  - Thus, knows what cards are left, too.
- We call an adaptive strategy greedy if, for each bid, the player names a value that appears the least number of times in the remaining deck.
- We can actually call this *the* greedy strategy.

Greedy Strategy is Unique

- Nondeterminism in selection: if values i and j remain in the same amount, greedy could choose either.
  - This does not matter.
- Assume k cards have been dealt, out of $\sum(s)$ total:
  - The remaining $\sum(s) - k$ are still uniformly distributed.
  - Choosing either leads to the same* game tree . . .
  - . . . at least the same branching probabilities + symmetry.
- In short, any "two" greedy strategies have the same chance of winning any m-round game.

Greedy Strategy is Optimal (?)

- The greedy strategy has the highest chance of surviving the next step in a game:
  - It's the greedy strategy.
- Still, there could be some strategy that starts out performing poorly, but finds (significantly) better positions later.

Greedy Strategy Can Tie for Optimality

- In Dundee played with only two values ($v = 2$), the greedy strategy does not strictly dominate.
- Informal Proposition:

  "The greedy strategy is at least as good as any other strategy when $v = 2$."

  - To be proved on the board.

Greedy Strategy is Optimal

- In Dundee played with at least three values ($v \geq 3$), the greedy strategy strictly dominates.
- Theorem:

  "Let $v \geq 3$ and $s = (s_1, s_2, \cdots, sv)$ be an arbitrary vector whose entries are non-negative integers. Let $m \leq \sum(s)$. Then the greedy strategy is the unique optimal strategy for the m-round s-game."

  - To be proved on the board.

Don’t Play Dundee

- The authors computed the probability of winning Dundee with a normal French deck of playing cards:
  - Approximately 27.019%.
  - Not too bad!

“I have tried to do this and have not yet managed to deal right through the pack; it is quite amazing how impossible it is.”

-- R. Harbin, inventor of Dundee

- There’s always Solitaire . . .

References

- Cardano. *De Ludo Aleae*. 1663.
- Catalan. *Solution d'un problème de probabilité*. 1837.
- Euler. *Calcul de la probabilitié dans le jeu de recontre*. 1753.
- Jordan. *De quelques formules de probabilité*. 1867.
- Litwack, Pikhurko, and Pongnumkul. *How to Play Dundee*. 2008.
- McFadyen. *Review of Liber De Ludo Aleae*. 2003.
- McFadyen. *An Historical Survey of the Development of Probability and Statistics*. 2003.
- Takács. *The Problem of Coincidences*. 1980.

Solitaire Win Probabilities

- Standard (52, infinite deals, draw 3):
  - Roughly 10% of games cannot be won.
  - 0.25% of games cannot even play one move!
- 82+% of games have a solution:
  - But you'd have to find it . . .
  - . . . which is still a common problem in AI, solved:
    - Heuristically
    - Poorly
    - Computationally intensively

- http://web.engr.oregonstate.edu/~afern/papers/solitaire.pdf