Deal or No Deal: Modeling a Game Show with Utility Theory and Machine Learning

Austin Biondi - Gonzaga University
Dylan Jamner - UCLA
Ashley Mullan - University of Scranton
Michael Wise - Marist College
Advisor: Dr. Sooie-Hoe Loke

CC-REU

March 11, 2024

Table of Contents

The Setup


$0.01
$1
$5
$10
$25
$50
$75
$100
$200
$300
$400
$500
$750
$1,000
$5,000
$10,000
$25,000
$50,000
$75,000
$100,000
$200,000
$300,000
$400,000
$500,000
$750,000
$1,000,000

Round 1


$25,521.51
$0.01
$1
$5
$10
$25
$50
$75
$100
$200
$300
$400
$500
$750
$1,000
$5,000
$10,000
$25,000
$50,000
$75,000
$100,000
$200,000
$300,000
$400,000
$500,000
$750,000
$1,000,000

Round 2


$31,617.91
$0.01
$1
$5
$10
$25
$50
$75
$100
$200
$300
$400
$500
$750
$1,000
$5,000
$10,000
$25,000
$50,000
$75,000
$100,000
$200,000
$300,000
$400,000
$500,000
$750,000
$1,000,000

Round 3


$70,336.80
$0.01
$1
$5
$10
$25
$50
$75
$100
$200
$300
$400
$500
$750
$1,000
$5,000
$10,000
$25,000
$50,000
$75,000
$100,000
$200,000
$300,000
$400,000
$500,000
$750,000
$1,000,000

Round 4


$187,661.90
$0.01
$1
$5
$10
$25
$50
$75
$100
$200
$300
$400
$500
$750
$1,000
$5,000
$10,000
$25,000
$50,000
$75,000
$100,000
$200,000
$300,000
$400,000
$500,000
$750,000
$1,000,000

Round 5


$279,990.97
$0.01
$1
$5
$10
$25
$50
$75
$100
$200
$300
$400
$500
$750
$1,000
$5,000
$10,000
$25,000
$50,000
$75,000
$100,000
$200,000
$300,000
$400,000
$500,000
$750,000
$1,000,000

Round 6


$146,986.90
$0.01
$1
$5
$10
$25
$50
$75
$100
$200
$300
$400
$500
$750
$1,000
$5,000
$10,000
$25,000
$50,000
$75,000
$100,000
$200,000
$300,000
$400,000
$500,000
$750,000
$1,000,000

Round 7


$91,962.48
$0.01
$1
$5
$10
$25
$50
$75
$100
$200
$300
$400
$500
$750
$1,000
$5,000
$10,000
$25,000
$50,000
$75,000
$100,000
$200,000
$300,000
$400,000
$500,000
$750,000
$1,000,000

Round 8


$100,024.67
$0.01
$1
$5
$10
$25
$50
$75
$100
$200
$300
$400
$500
$750
$1,000
$5,000
$10,000
$25,000
$50,000
$75,000
$100,000
$200,000
$300,000
$400,000
$500,000
$750,000
$1,000,000

Round 9: The Deal

Problem Setup

We aimed to answer a few questions about the show.

1. Given that there is no publicly available framework, how can we accurately model the banker's offer system?
2. What do we notice about past players' gameplay?
3. How can we replicate empirical gameplay data and describe it in terms of utility theory?
4. What methods are best to generate an optimal player strategy?

Expected Utility Theory

- A **utility theory** contains a binary **preference relation** on a set of X elements that represent choices or actions[2].
  - $x \preccurlyeq y \iff$ "x is not preferred to y"
  - not $x \preccurlyeq y \iff$ "it is false that x is not preferred to y"
- Utility functions characterize the behavior of $\preccurlyeq$ via assigning values to outcomes that respect the preference relation.
- **Expected Utility Theory** is a method that identifies optimal decisions when presented with a risky choice. It is based off of expected value of the choice and individuals' utility functions.

	Option A	**Option B**
Probability	30% chance	70% chance
Prize	\$5	\$3
Expected Utility	\$1.50	\$2.10

Example Player

Player	Education	Stop	Their Case	Winnings
Bezos	High	9	$750	$202,281.22

Round	Expected Value	Banker Offer
1	$168,368.30	$25,521.51
2	$169,089.40	$31,617.91
3	$212,348.20	$70,336.80
4	$290,718.80	$187,661.90
5	$337,625.00	$279,990.97
6	$205,150.00	$146,986.90
7	$156,437.50	$91,962.48
8	$175,250.00	$100,024.67
9	$250,375.00	$202,281.22

Modeling the Banker's Offer: Linear Regression and Other Candidates

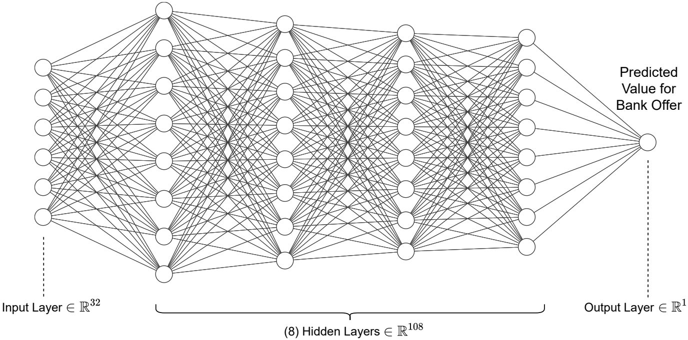
- Modeling the banker's offer is not a simple task.
- We partitioned our data to develop a model (75% training, 25% test).
- A log linear regression seemed fairly accurate at first. Could we do better?

$$\ln(\widehat{Bank\ Offer}) = \beta_0 + \beta_1 \ln(\widehat{Expected\ Value}) + \beta_2 \widehat{Remaining\ Cases}$$

- How did other machine learning methods compare?

Model	Mean Absolute Error	Root Mean Square Error	R^2
Log Linear Regression	13977.0295	22824.6396	0.9171
Random Forest	15372.0245	23571.227	0.9116
kNN ($k = 8$)	16181.0126	22654.5502	0.9183
Deep Neural Network (Sequential, 8 hidden layers)	11368.5809	16849.2989	0.9548

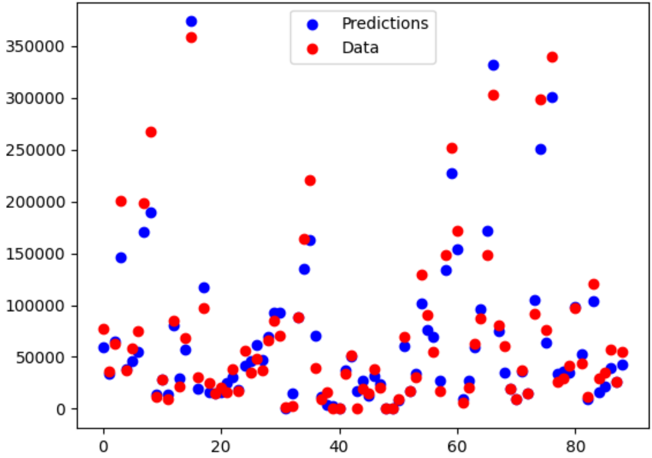
Modeling the Banker's Offer: Neural Network



Modeling the Banker's Offer: Neural Network (cont.)

- Inputs into the neural net:
  - Round number
  - Status of each case
  - Number of remaining cases
  - Bank offer of the previous round
  - Expected value
- Activation Functions:
  - Sigmoid for input layer
  - Rectified Linear Unit (ReLU) function for hidden layers
- Measures of Fit:
  - Mean Absolute Error: approximately 11000-13000
  - Mean Squared Error: approximately 16000-19000
  - R^2: approximately 0.94-0.957

Bank Offer: Neural Network Visual



Modeling the Player: Neural Network

- Initially, we tried to implement a neural network to model whether or not the player decided to take the deal in each round.

75/25	80/20	90/10
86.5%	88.7%	80%
89.9%	91.5%	82.9%
88.8%	94.4%	94.3%
84.3%	88.7%	97.1%

Modeling the Player: Logistic Regression

- The model is trained on the dataset from Post et al.
- It outputs probability of the player taking the deal.
- A cutoff of 50% probability is used to generate the decision of the player.

	Deal	**No Deal**
Deal	26	11
No Deal	15	303

This model predicted the player's decision with 92.7% accuracy. It was used to generate decisions for the 1000 simulated players.

Modeling the Player: Expo-Power Utility Function

- Our first attempt at a utility function was an expo-power function [1]:

$$u(w) = \frac{1 - e^{\frac{-\beta \cdot w^{1-\alpha}}{1-\alpha}}}{\beta}$$

- The alpha parameter controls the absolute risk aversion.
  - Used value from Blavatskyy [1]: -0.3567
- The beta parameter controls the relative risk aversion.
  - Found a value experimentally: $1.0 \cdot 10^{-7}$

Modeling the Player: Inputs into Utility Function

- In each round contestant has a choice between two distinct outcomes: taking the deal or refusing the deal and advancing to the next round.
- Our first attempt at the utility for taking the deal was simply plugging in the value of the bank offer into the utility function.
- Our first attempt at the utility for selecting No Deal was to pass all potential states for the next round through our bank offer neural net and take the average.
  - Intuitively, this is the average predicted bank offer of the following round.

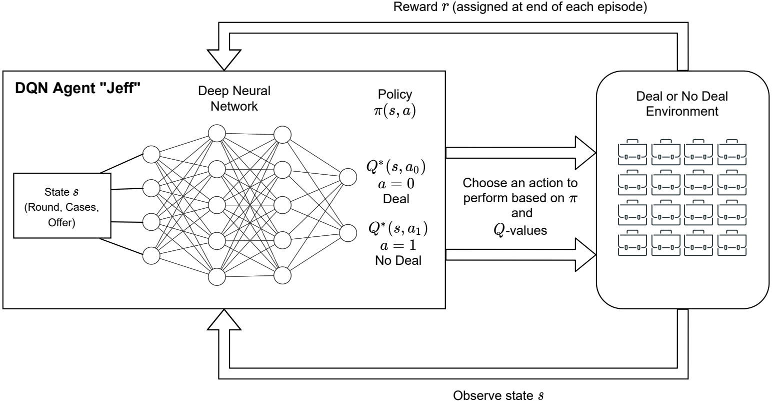
Modeling the Player: Final Utility configuration

- After trying a multitude of inputs, we settled on the following:
  - Utility for a deal is the bank offer plus the scaled difference between the expected value of the round and the expected value that had been predicted for that round:

    $$u(offer) + 0.1 \cdot (EV - PEV)$$

    - This is inspired by the path-dependent model given by Kőszegi and Rabin [3].
  - Utility for a no deal is the the average expected utility over all possible states of the next round, plus a scaled average predicted banker offer.
- This resulted in an accuracy of up to 92.12% on the empirical data and 86.74% on the simulation data.

Modeling the Player: Reinforcement Learning



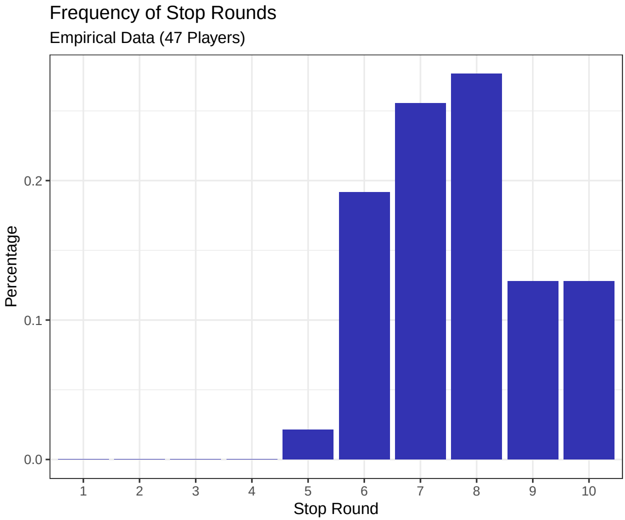
- ϵ-Greedy Policy for action a at state s for $\epsilon \in [0, 1]$

$$a_s = \begin{cases} \max(Q^*(s, a)) & \text{with prob.} \quad 1 - \epsilon \\ \text{choose random } a & \text{with prob.} \quad \epsilon \end{cases}$$

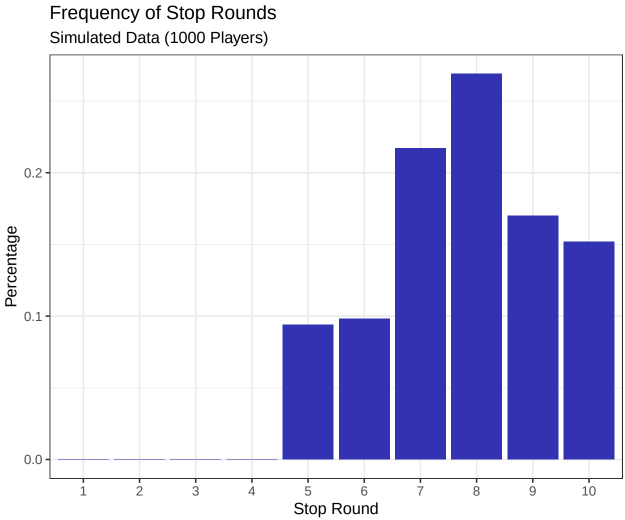
Modeling the Player: RL Results and Implications

- Jeff tends to end the game early, often going with safer offers without maximizing winnings compared to human players.
- On the other hand, the DQN agent [4] occasionally prioritizes playing through all 9 rounds because of times it remembers constantly saying "no deal" and walking away with a big case (eg. $1,000,000). This creates a sporadic distribution of ending rounds.
- Discount factor $\gamma \in [0, 1]$, which accounts for how events in the distant future are weighted less than events in the immediate future ($\gamma = 0.95$).
    - Take the deal given offers higher than the expected value of all 26 briefcases, $131,477.54.
- Performance compared to simulation data:
    - Jeff went home with more money than the simulated contestants 42.3% of the time (lots of room for improvement).

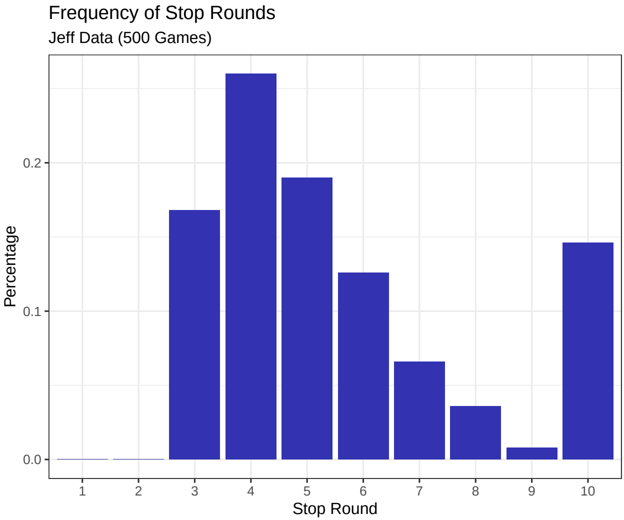
Where to Quit?



Where to Quit?



Where to Quit?



Future Investigations and Improvements

- We can incorporate stochastic choice into our player utility model.
  - Utility theory models assume that players will choose the option with the greatest utility.
  - People don't always act in this fashion, and they sometimes choose plays with lower utilities.
  - This is defined as a **tremble**.
  - By nesting our utility function in a model of stochastic choice, we can incorporate this behavior into our model and possibly improve the accuracy of our predictions.
- We can modify or improve the policy and reward system of the DQN agent to better maximize winning strategies.

References I

Pavlo R Blavatskyy and Ganna Pogrebna.
Models of stochastic choice and decision theories: Why both are important for analyzing decisions.
Journal of Applied Econometrics, 25(6):963–986, 2010.

Peter C Fishburn.
Utility theory.
Management science, 14(5):335–378, 1968.

Botond Kőszegi and Matthew Rabin.
Reference-dependent consumption plans.
American Economic Review, 99(3):909–36, 2009.

References II

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al.
Human-level control through deep reinforcement learning.
Nature, 518(7540):529–533, 2015.

Thank you!

This presentation summarizes the results of the CC-REU NSF summer REU experience (DMS-2050692) where these questions were explored.