Interpretation of Neural Network Players for a Generalized Divide the Dollar Game Using SHAP Values

Garrison Greenwood [1] Hussein Abbass [2] Aya Hussein [2]

[1]Portland State University, Portland, OR USA

[2]University of New South Wales, Canberra, Australia

Explainable AI (XAI)

Machine learning (ML) algorithms often operate as black-box function-approximators. They receive inputs and render output predictions.

Unfortunately how their function approximation abilities work is not well known.

XAI attempts to get some answers, but not much prior work in multi-agent systems. One of our aims is to propose a technique for explaining behaviors in multi-agent systems using XAI.

Divide the Dollar Game

Divide the Dollar is a simple game designed to study how individuals act in bargaining situations. The 2-player version works as follows:

\$1 dollar is split among two players. The players simultaneously submit a **bid** indicating how much of the dollar they are willing to accept.

If the sum of the bids is $\leq \$1$, the players get their bids as a payoff. However, it the sum of the bids $> \$1$, both players get nothing.

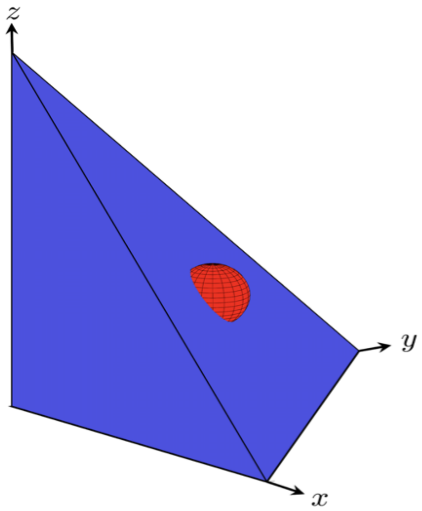
Generalized Divide the Dollar

In 2018 Dan Ashlock and I extended Divide the Dollar to the Generalized Divide the Dollar (GDD) game. It differs in the following ways:

1. it is a $N > 2$ player game
2. small subsidizes (≈\$0.05–\$0.10) are possible

Note: subsidizes only apply if all bids are “fair”—i.e., roughly equal.

Coordinated Subspace



Each player's bid is a coordinate in 3-dim space. Associated with each point is the bid total.

Bids are coordinated if the bid total $\leq \$1$; otherwise they are uncoordinated.

The hemisphere is the subsidy region.

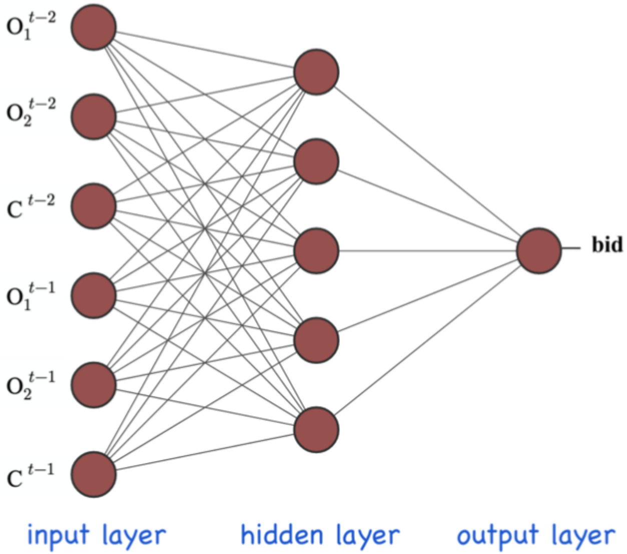
NN players

In 2022 Dan Ashlock and I evolved NN players for the GDD game*. The goal: evolve three players that would consistently offer fair bids in order to exploit the subsidy.

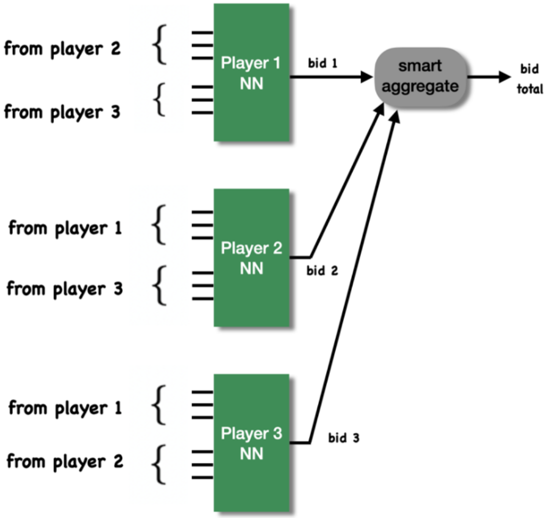
Input features were the bids of a player's two opponents from the previous two rounds. The NN regression output was a player's bid.

* G. Greenwood and D. Ashlock. *Evolving neural networks for a generalized divide the dollar game.* In 2022 IEEE Cong. on Evol. Comput. (CEC), pages 1—8, 2022.

Player NN Archtecture



3-player GDD Framework



Interpreting Behavior

NN players were evolved using a CMA-ES algorithm. 3-player tournaments were used to evaluate the NN behavior.

Three players were successively evolved using CMA-ES. However, NNs are black boxes, so it was not clear how the predictions were made.

That's where XAI methods proved to be useful.

Background: Shapley's Problem

Lloyd Shapley[1] was interested in the following coalition problem:

Problem Statement

A coalition $C \subset N$ cooperates to complete some activity.

When the activity is finished, there is a payoff to distribute among the $|C|$ coalition members.

What a fair distribution of the total payoff among the coalition members?

[1] 2012 recipient of the Nobel prize in economics sciences

What is a fair distribution?

Intuitively, a fair distribution would be $\frac{v(N)}{N}$. But is that really fair?

Example

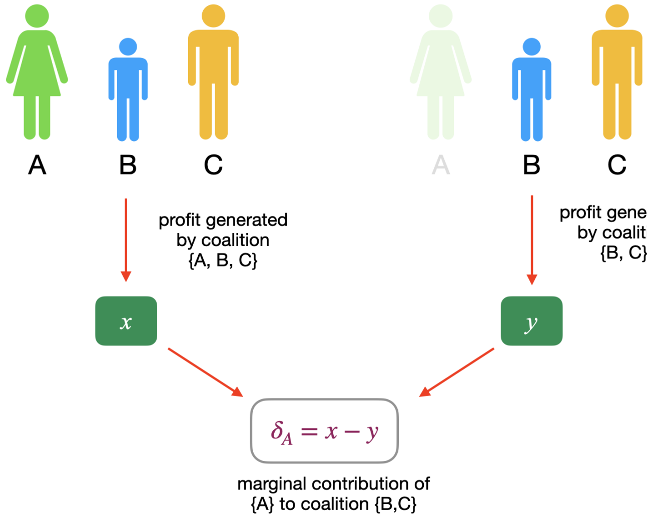
Three people want to build a house, sell it, and split the profit.

- one person does all of the plumbing and internal wiring
- one person installs the doors and windows
- one person lays the foundation, puts up the walls and roof

Clearly the 3rd person did most of the work and therefore should get a greater share the profit.

Question: How would you determine a fair distribution of the profit in this example?

Marginal Contribution



A
B
C
A
B
C
profit generated by coalition {A, B, C}
profit generated by coalition {B, C}
x
y
$\delta_A = x - y$
marginal contribution of {A} to coalition {B,C}

Shapley Value (cont.)

The Shapley value for player i is a weighted sum of the marginal contributions of i over all possible coalitions.

Let $C \subseteq N$ be a coalition of players with $n = |N|$.

Definition

The Shapley value of player $i \in C$ is

$$\varphi_i = \sum_{C \subseteq N \setminus \{i\}} \underbrace{\frac{|C|! \times (n - |C| - 1)!}{n!}}_{\text{weight}} \underbrace{\left(v\left(C \cup \{i\}\right) - v\left(C\right)\right)}_{\text{marginal contribution}}$$

$v(C)$ is a characteristic function which gives the value of the coalition C (problem dependent).

SHapley Additive exPlanations (SHAP)

Lundberg and Lee introduced SHAP to interpret ML black box models through Shapley values. Games now become ML models, players now become input features and characteristic functions now become model predictions.

Definition

The SHAP value ϕ_i of feature $i \in C$ is

$$\phi_i = \sum_{z' \subseteq x'} \underbrace{\frac{|z'|! \times (M - |z'| - 1)!}{n!}}_{\text{weight}} \underbrace{\left(f_x\left(z'\right) - f_x\left(z'/i\right)\right)}_{\text{marginal contribution}}$$

where M is the number of features and $x' \in \{0,1\}^M$. $x_j = 1$ if feature j is present in the feature vector and $x_j = 0$ if missing. f_x is the ML model prediction for input feature vector x.

SHAP (con't.)

The **prediction** of a ML model f to input feature x is given by

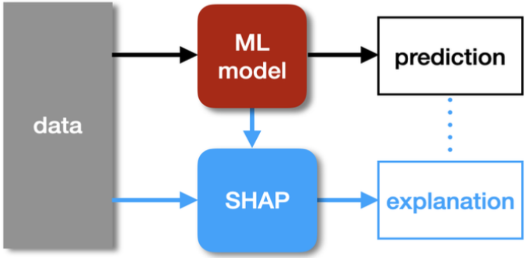
$$f(x) = \phi_0 + \sum_{i=1}^{M} \phi_i x'_i$$

where $\phi_0 = E[f(x)]$ (avg over all predictions in the database)

SHAP computations are available in an open-source library that efficiently computes Shapley values in ML domains.

Available in several programming languages (MATLAB, R, python)

The SHAP Framework



The ML model is first fit to a database. Then observations from the database and the model are fed to SHAP, which computes the SHAP value ϕ_i for each feature i.

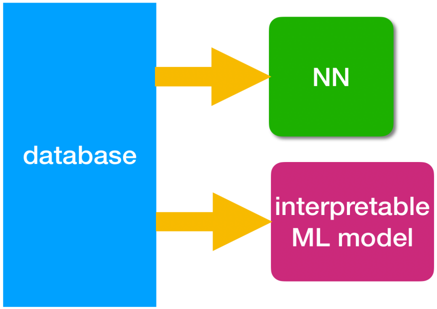
$$f(x) = \phi_0 + \sum_{i=1}^{M} \phi_i x_i'$$

then provides the explanation. $f(x)$ is the player's bid for the input feature vector.

1st Problem

PROBLEM: NNs aren't interpretable. . .

SOLUTION: Train a ML model that *is* interpretable using the same database used to train the NN.

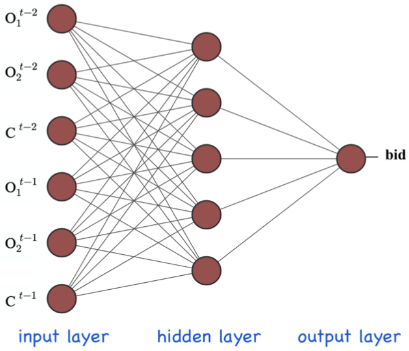


Examples of interpretable ML models include XGBoost and random forrests.

2nd Problem

PROBLEM: We don't have a database to train an interpretable ML model. . .

SOLUTION: Create one from the NN players we previously evolved. (Similar to creating surrogate models.)



Take a known coordinated input feature vector, and randomly perturb the input features with a small, normally-distributed random variable. Record the feature vector and NN output.

Repeat for known subsidized and uncoordinated feature vectors. We created a database of 2000 feature vectors using this method.

Comparing the NN with XGBoost

The R^2 metric measures database fit; the closer to 1.0, the better the fit. In the ML community R^2 is frequently used to compare different regression models on the same database.

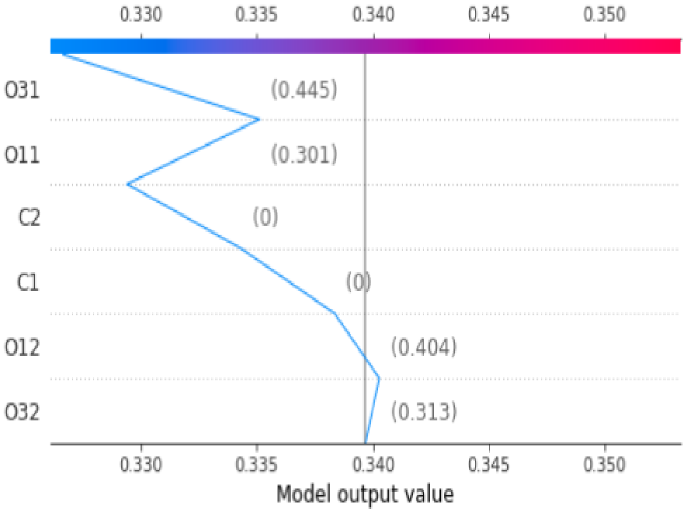
$$R^2 \ = \ 1 - \frac{\sum(y_{\text{pred}} - y_{\text{mean}})^2}{\sum(y_{\text{actual}} - y_{\text{mean}})^2}$$

y_{pred} is from the XGBoost model; y_{actual} is from the NN model.

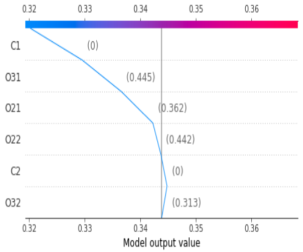
The XGBoost model had $\mathrm{R}^2 = 0.996$. Thus, any assumptions made by the XGBoost ML model holds for the NN model. Interpreting the XGBoost model also interprets the NN model.

Apply SHAP routines to the XGBoost models and extract the SHAP values to conduct the interpretation.

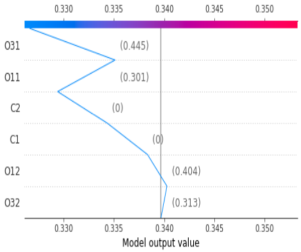
Decision Plots



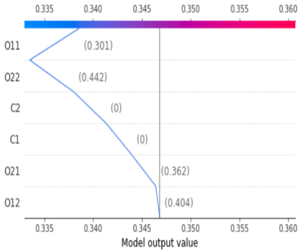
Decision Plot (coordinated bid total w/o subsidy)



(a) NN1

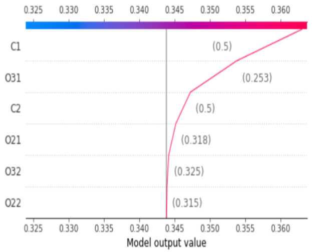


(b) NN2

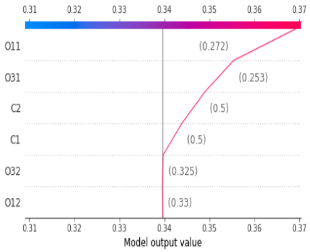


(c) NN3

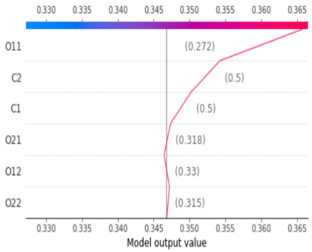
Decision Plot (uncoordinated bid total)



(a) NN1

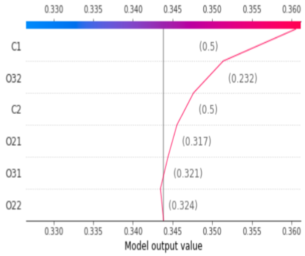


(b) NN2

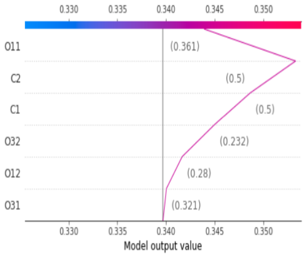


(c) NN3

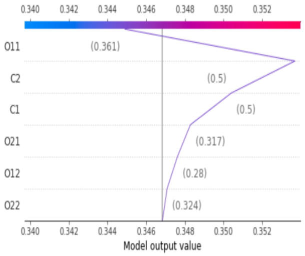
Decision Plot (coordinated bid total w/subsidy)



(a) NN1

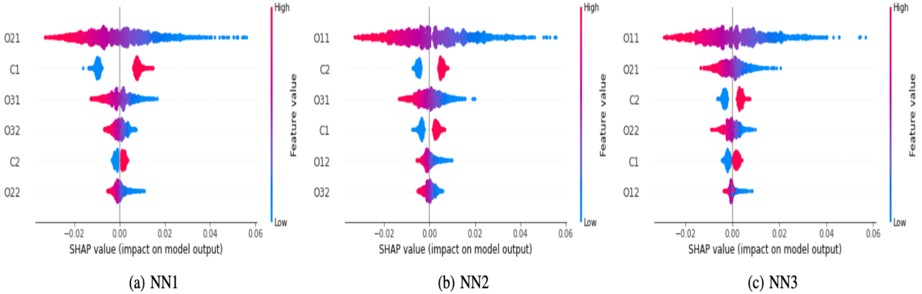


(b) NN2



(c) NN3

Global Interpretation



(a) NN1 (b) NN2 (c) NN3

Final Remarks

- The NN players can exhibit greedy and altruistic behavior when subsidies are possible.
- The beeswarm plot indicates opponent's previous round bid had the most impact on future bids
- SHAP helps identify the most important features. Removing unimportant features reduces training times and helps prevent overfitting.
- Follow-on work: Eliminate the round $(t-2)$ inputs and re-evolve the GDD NN players. Will the smaller NN ML models still exhibit greedy/altruistic behavior?

Questions