PS1 report

Lana Gorlinski, Yuto Yagi

March 31, 2020

1 Approach

We used machine learning approach loosely based on Hartford et. al.[1] paper. We added max welfare, maxmax, minimax and unique NE as features and apply a multi-layer perceptron regressor. We do this largely because the lack of repeated play here: as we saw in Kolombus et. Noti, when history is sufficiently short, behavioral based features increase prediction accuracy over algorithms that examine play history alone. Since we only have a single-iteration game here, we decided that a mix of behavioral features with machine learning modeling would be the optimal approach.

2 Description

We constructed our model as follows:

1. **Add features to the data**: here, we chose Nash Equilibrium, altruism, maxmax, and minimax. We thought NE was the canonical game theory model, so its inclusion would be beneficial. Maxmax, or the action that results in the best possible outcome, we hoped would highlight the human tendency towards high numbers. Minimix, or the action whose worst payout is the highest, would look at pessimism. Altruism refers to the action maximizing the total welfare of both players in the best case scenario, which again highlights the tendency towards large numbers as well as highlighting fairness, something found in the course to be salient in behavioral prediction. Each feature was represented as a set of dummy binary variables, with each feature split into three dummies by action. The number of nash equilibria was represented as a scalar.

2. **Train and cross-validate** First we standardized the features. Then we cross-validated the net on the training data and ensured that our features were sufficient to get a good prediction

3. **Train on full dataset** We then fit our cross-validated net to the full training data set; since we have so few samples, we thought using all of them in our training would give us the best predictive power.

3 Interesting discovery

Interestingly enough, the single biggest increase in accuracy occurred when incorporating Nash equilibrium as features; previously, the model was only doing slightly better than random chance, yet including the NE as features shot accuracy up to around 50% , depending on the specifc training/test split. We've read a lot about how nash equilibrium may not be reliable in behavioral settings because of the degree of contingent reasoning required, so it would be interesting to figure out why it was so salient here and why the model didn't train to factor in Nash Equilibrium on its own.

4 Results

Q and A scores of our final approach on the training data.
Below is the scores of a model which learned from the training data applied to the exact same data.
A = 0.968
Q = 0.007
Below is the scores of a model which learned from "training" portion of training data applied to separate "test" portion of the training data. (This gives different results depending on the stochastic different train/test split)
A = 0.650
Q = 0.081

5 Future works

Tuning hyperparameters of the multi-layer perceptron regressor (number of iteration, reglarization term etc.) Also, examining the importance of the different features (perhaps running iterations of fitting the net with dropout of certain features in between each iteration).

6 Relevant works

See the list of references.

References

[1] *Hartford et. al.*.
J.Hartford, J.R.Wright, K.Leyton-Brown *Deep Learning for Predicting Human Strategic Behavior*

[2] *Kolombus et. Noti*.
Y. Kolombus, G. Noti *Neural Networks for predicting Human Interactions in Repeated Games*