CS 136 Project Proposal

Phillip Yu, Bovey Rao, Tiffany Yu

1 Idea One: Iterated Prisoner's Dilemma

Our first idea is to research existing literature on the current highest-performing iterated prisoners dilemma strategies (irregardless of any strategy limitations, such as the 5-node limitation given to us in PSET 0). We then plan to code up each of these strategies and run them against in each other in a large simulation, recording how each of these strategies perform in different populations. We also plan to change certain simulation parameters, such as the presence of noise and the potential payoffs, and compare how each strategy does in these modified games. Finally, we plan to summarize and compare the pros and cons of each strategy and analyze in depth the specific settings in which each given strategy performs well.

Additional notes/questions:

- In addition to summarizing strategies for the iterated/parameter-modified prisoner's dilemma, it might also be interesting to see which underlying strategy "intuitions" can be generalized to other prisoner's dilemma versions, possibly incorporating (for example) reputation systems.
- As someone suggested in the Piazza comments, it could also be interesting to run the results of our simulation against "common" iterated prisoner's dilemma strategies, and see which of the "elite" strategies perform best against a "common" population (rather than those who perform best against an "elite" population of other great strategies).

2 Idea Two: Restaurant Recommender System

Existing systems for restaurant searches are primarily reputation systems, while there are no popular systems for recommending restaurants to specific users based on their previous tastes and interests. We are thinking to build a hybrid recommender system based on user-user collaborative filtering, using Yelp data on restaurants and user reviews in addition to content-based user profiles based on a user's checked-in restaurants, reviews, and inputted restaurant preferences. We are also considering factoring in location and price into the user profile. We will likely need to include some hard-coded data about how restaurant cuisines may be similar to one another. To determine similarity between users, we will use machine-learning clustering methods based on the above information.

For a given Yelp user, our system will output a list of the top 10 recommended restaurants for that user that they have not checked into or reviewed. For users with only a few reviews, we will randomly generate a list of "inputted preferences"; for users with many reviews, reviews appear to be better representations of their preferences, so their review information will be weighted more heavily.