Documentation of Input Files and Code

Peng Shi, July 2019

If use the data files or code in anyway, please cite my paper "Optimal Priority-Based Allocation Mechanisms."

1. Description of Input Files

A neighborhood is represented by a geocode between 0 and 867, and a school is represented by a 4-digit school ID.

The following input files are in the directory `data/raw_inputs`:

- **capacity_schools.csv**: a mapping of geocode to the ID of its default school, which is the closest capacity school.
- **capacity_schools_3zone.csv**: the analogous mapping for the 3-Zone plan, in which the default school is the closest capacity school within the zone-based menu.
- **geocode_characteristics.csv**: the expected number of applicants and the area of each neigborhood (geocode). The expected number of applicants is based on the location of applicants in assignment year 2010 through 2013. Note that certain geocodes rae not included because there were zero applicants from the location in the four yeras.
- **geocode_location.csv**: the longitude and latitude of the centroid of each geocode.
- **geocode_sampling_weights.csv**: the weights used when sampling students from within each geocode of a particular region, used by the simulation engine. This is based on data from 2010-2013.
- **geocode_sampling_weights_2014.csv**: updated sampling weights based on the applicants from 2014.
- **geocode_to_geocode_distances.csv**: a file containing Google Maps walking distance from the centroid of each geocode to each other geocode.
- **geocode_to_neighborhood_mapping.csv**: a mapping of each geocode to each of the 14 regions.
- **geocode_to_school_distances.csv**: the Google Maps walking distance from the centroid of each geocode to each school, in meters.
- **miles_of_busing_for_optimization.csv**: equal to the above except it is converted to miles, and the schools withtin the walk-zone of each geocode are given a value of zero. This matrix is used to compute busing distances, which is only relevant for schools outside of the walk-zone.
- **regional_population.csv**: estimated proportion of applicants coming from each of the 14 regions.
- **school_characteristics.csv**: the inferred qualities and capacities of each school. The inferred qualities are based on submitted choice rankings from 2013 under the 3-Zone plan. Note that the normalization in this data file is different from that of the paper. The inferred quality Q_j defined in Appendix D.2 of the paper is equal to the value here divided by 0.531. The parameter β in Appendix D.2 is equal to $1/0.531$, and the parameter γ is equal to $0.456/0.531$.
- **school_characteristics_2014.csv**: a similar file to the above except that the inferred quality are based on rankings from 2014 under the Home-Based plan. The inferred quality Q_j for 2014 defined in Appendix D.2 of the paper is equal to the value here divided by 0.610. The parameter β is equal to $1/0.610$, and the parameter γ is equal to $0.223/0.610$.
- **walk_zone_table.csv**: the set of schools within the walk-zone of each geocode.

The other input files are:

- **data/assignment_plans/plans_for_simulation/3ZoneParams.csv**: an encoding of the 3-Zone plan used in Boston in 1988-2013. The column "Priority" is 1 if the school is in the walk-zone of the geocode, and 0 if the school is in the choice menu but not in the walk-zone.
- **data/assignment_plans/plans_for_simulation/HomeBasedParams.csv**: an encoding of the Home-Based plan implemented in Boston in 2014. The column "Priority" is 1 if the school and the geocode are both in East Boston or both outside of East Boston, and 0 if there are on opposite sides of the bridge linking East Boston with the rest of Boston.
- **data/assignment_plans/plans_for_simulation/AshlagiShi2015Params.csv**: an encoding of the plan from Ashlagi & Shi (Management Science 2015). The column "Priority" contains the priority boost b_{tj} from neighborhood t to school j.

2. Running the Code

The code is written using Python 3.7 (from the Anaconda 3 distribution), and uses Gurobi 8.1. One should also have installed the packages: `ipython`, `numpy`, `matplotlib`. See the docstrings in each file and class for descriptions of each component.

To generate all of the computational results for the paper, go to the directory immediately above `data` and execute the following lines

```
ipython peng/assignment_plans/main/compute_plans.py
ipython peng/assignment_plans/main/tabulate_outcomes.py
ipython peng/assignment_plans/main/make_figures.py
```

The first line solves the LP for the Boston case study with parameters $(A, B_1, B_2, B_3) = (0.5, 0.6, 8.5, 6.2)$, which is the basis of the optimized plan in this paper. It also solves the LP with parameters $(A, B_1, B_2, B_3) = (0.5, 0.64, 8.52, 8.18)$ to derive the theoretical upper bound for the finite market stochastic model. The computed plans are stored in the folder `data/assignment_plans/large_market_plans` and `data/assignment_plans/plans_for_simulation`. The summary of the theoretical upper bound in the continuum model is written to the file `data/results/other/theoretical_upper_bound.csv`. (This represents the upper bound in the finite market stochastic model of `(average utilitarian welfare + lowest expected utility of any neighborhood)/2`. (Note that the plan from Ashlagi & Shi (2015) can also be obtained by solving $(A, B_1, B_2, B_3) = (0.5, 0.5, 100, 100)$.)

The second line simulates the various assignment plans under the finite market stochastic model (Table 1 of the paper), as well as under the model with updated demand estimates (Table 4). It also compares the predictions of the continuum model for the optimized plan with discrete simulations under minimal randomness of the applicant population and the quotas prescribed by the continuum model (Table 5). The details of the simulations are written to the folder `data/logs/simulations`. The final results are written to the folder `data/results/tables`.

The last line plots all of the figures in the paper. It must be run after `tabluate_outcomes.py` because the expected utility plots are based on the intermediate outputs of the simulations.