1 Data

The data are from two sources. Both are proprietary, and are therefore not posted here. The first is the Superpanel which can be purchased from Kantar Worldpanel. The second is the Store Dataset which can be purchased from the Institute for Grocery Distribution (http://www.igd.com/). From the raw data, we aggregate products into categories, create price indices and draw a sample of consumers as discussed in the paper. For each consumer in the sample we create store choice sets from the Store Dataset (also as discussed in the paper). Once this is done, the data are read into Matlab and calculations performed by the files described in this document. The model can be estimated and all results obtained by running the script file `TSS_script.m`.

2 Main files used for estimation

1. `TSS_script.m` is the script file from which all other files are run. It
   (a) gets the data ready by calling `TSS_input.m`,
   (b) creates the first- and second-stage weighting matrices,
   (c) minimizes the GMM objective function by running the script file `TSS_optimization.m`
   (d) calculates standard errors, fit measures (out-of-sample predictions etc.), margins, profit derivatives for consumer types, and confidence intervals.
2. `TSS_input.m`
   (a) reads the .csv files that contain the data into Matlab,
   (b) creates all the variables needed for estimation etc. and stores them in the structures `inp` and `inp2`.
3. `TSS_W_second_stage.m` calculates the variance matrix of the moments.
4. `TSS_optimization.m` creates an initial population, chooses settings and minimizes the objective function using Matlab's `ga.m`.
5. `TSS_gmm.m` calculates the value of the objective function.
6. `TSS_moments.m` calculates the value of the empirical moments.
7. `TSS_nonlin_moments.m` calculates the values of the cross-period and cross-category moments.
8. `TSS_quantities.m` calculates the model's predicted quantities and shopping choices.
9. `QP_nonneg.m` solves the constrained quadratic programming problem in variable utility.

3 Main files used for post-estimation calculations

10. `TSS_acov.m` calculates the estimated covariance matrix of the estimator.

11. `TSS_print_estimates.m` prints the estimates and standard errors to Excel.

12. `TSS_print_predictions.m` prints to Excel within-sample predictions, out-of-sample predictions, and other measures of fit.

13. `TSS_predictions.m` calculates these predictions.

14. `TSS_pairs.m` also calculates these predictions.

15. `TSS_markups_weekly.m` calls functions that calculate and print margins and price derivatives.

16. `TSS_weekly_input_struct.m` creates an input structure similar to `inp` but only for the middle week of the sample.

17. `TSS_input_week.m` imports data for all weeks and creates variables used by `TSS_weekly_input_struct.m`.

18. `TSS_print_elasticities_markups.m` prints elasticities and margins etc. to Excel.

19. `TSS_multipledraws_markups.m` calculates elasticities and margins.

20. `TSS_Qfirmcat.m` calculates derivatives of quantities and choice probabilities with respect to prices.

21. `TSS_counterfactual_exog_dist.m` calculates and prints derivatives of visits and profits from different consumer types with respect to price.

22. `TSS_profit_visits.m` calculates profits and visits at different prices.

23. `TSS_markups_counterf_distribution.m` calculates margins and price derivatives for multiple draws from the estimated asymptotic distribution of the estimator. Quantiles of the resulting distribution are used to calculate confidence intervals.

4 Other files

24. `TSS_distance_histogram.m` produces a histogram of observed and predicted distances travelled.

25. `TSS_mutationgaussian.m` modifies the mutation process of `ga.m` to allow us to save intermediate results.

26. `TSS_simplechoicelikelihood.m` calculates the likelihood for the simple logit model used to assign stores when location is unobserved (discussed in Section I).

27. `beta.mat` contains the estimates from that simple logit model.

28. `TSS_simplechoiceprob.m` calculates the choice probabilities in that simple logit model.

29. `yearindices.m` auxiliary function (see explanatory comments in code).

30. `TSS_notin.m` auxiliary function (see explanatory comments in code).

31. `hh_code_estimation_sample.mat` contains a list of identifiers of the households used for estimation.

32. `hh_code_validation_sample.mat` similarly for the validation sample.

33. `nu2000_estimation_sample.mat` contains the random taste draws used for the estimation sample.

34. `nu2000_validation_sample.mat` contains the random taste draws used for the validation sample.

35. `first_stage_estimates.mat` contains the estimates from the first stage of estimation.