Replication Code for "Nonparametric Estimation of the Random Coefficients Model: An Elastic Net Approach"

Florian Heiss, Stephan Hetzenecker, and Maximilian Osterhaus

November 2020

This readme file describes how to run the programs provided in this folder. The folders *Monte_Carlos* and *Application* contain the R code for the replication of the results presented in Section 4 and Section 5 of the paper.
The code for the Monte Carlo studies is structured such that each evaluation of the functions *MC_Continuous* and *MC_Discrete* returns the results of one Monte Carlo replicate for different number of grid points and observations. We ran the Monte Carlo experiments on a high performance cluster and sent the files *MC_Discrete_Parallel.R* and *MC_Continuous_Parallel.R* 200 times through a batch file to the cluster.
The R code to the application imports the data from the *mlogit* package. So no additional data is needed. The code estimates the random coefficients distribution with the semiparametric estimator, FKRB estimator, and the generalized estimator and calculates the corresponding own- and cross-travel time elasticities.

The folder *Functions* contains the functions that are used inside the code of the Monte Carlo studies and the application. The following functions are only relevant for the Monte Carlo setups:

- *cdfTrue* calculates the empirical joint CDF from the random draws provided to the function. We only use the function to calculate the joint CDF of the true discrete distribution in the discrete Monte Carlo experiment.
- *gendata* generates multinomial choice data for the Monte Carlo experiments. The function returns a matrix of covariates and a vector of binary dependent variables.
- *gmdist* generates random draws from a mixture of multivariate normals.

The subsequent functions are not specific to the Monte Carlo experiments and our application and can also be used for other applications:

- *cdfEst* calculates the estimated joint CDF as described in Equation (5) in the paper.

- *crossValidation* conducts a k-fold cross validation to find the optimal tuning parameter μ. For the random shuffling of the data, the function assumes that the regressor matrix and the dependent variable is balanced and ordered in the sense that the number of alternatives (j) is the same for every observation unit (n) and that the first j observations belong to observation unit 1, the next j observations to observation unit 2, and so forth. The function returns the tuning parameter that minimizes the out-of-sample mean squared error, maximizes the out-of-sample log-likelihood, minimizes the out-of-sample prediction error, and the tuning parameter according to the one-standard-error rule for the mean squared error.

- *estimator* takes the regressor matrix, the dependent variable, and the ridge tuning parameter μ as input and returns the estimated weight at every grid point. The function conducts a least-squares estimation subject to the constraints that the weight at every grid point is nonnegative, and that the weights sum to one. Providing the argument $\mu = 0$ returns the weights estimated with the FKRB estimator.

- *halton_draws* generates a d-dimensional Halton sequence using the prime numbers provided as input.

The functions *estimator*, *crossValidation*, *cdfEst*, and *halton_draws* are programmed flexible in terms of the number of observation units, alternatives, and grid points so that they can be easily applied to alternative data sets. To use the function for the estimation of one-dimensional distributions, it is necessary that the grid used for the estimation and evaluation is provided as a matrix and not as a vector.

In addition to the R code, we provide Matlab code for the continuous Monte Carlo experiment presented in Section 4. The code is similarly structured as the R code. The file *MC_Continuous.m* is the main file which conducts the Monte Carlo experiment for different number of observations and grid points. The functions that are used inside this file have the same names as the corresponding R files. All results presented in the paper were produced with the R code.

If you have any questions or problems with the code, do not hesitate to write to stephan.hetzenecker@vwl.uni-due.de or m.osterhaus@hhu.de. We are happy to help.