Codes to Estimate the Random Parameter Model

Jose Apesteguia[*], *Miguel A. Ballester*[†] & *Angelo Gutierrez*[‡§]

Description

The accompanying files provide Stata and Matlab routines to estimate the Random Parameter Model (RPM) following Apesteguia and Ballester (Forthcoming). Angelo Gutierrez provided outstanding research assistance in the development of these codes. The *MAIN* file in each folder contains detailed comments explaining the codes for each program. We illustrate their use with data from Andersen, Harrison, Lau, and Rutström (2008). However, the codes can be easily adapted to other datasets. We've made an effort to make the codes flexible and easy to understand.

Data

The codes take as input a dataset with information of different risk-aversion choice tasks in the style of the multiple-price lists of Holt and Laury (2002). An observation i consists in two lotteries, X_i and Y_i, that are presented to an individual. Lottery X_i pays $x_{1,i}$ with probability p_i and $x_{2,i}$ with probability $1 - p_i$. Lottery Y_i pays $y_{1,i}$ with probability p_i and $y_{2,i}$ with probability $1 - p_i$. The individual can express preference for one of the lotteries or indifference between the two. We define an indicator variable C_i and set $C_i = 1$ if lottery Y_i is chosen and set $C_i = 0$ if X_i is chosen instead. If the individual is indifferent between both lotteries, we set $C_i = -1$. Let id_i denote the individual who is presented with the lottery in observation i. The following table illustrates the database used in the codes by showing 7 typical observations in the dataset used by the codes.

Tab. 1: Example of the data used as input in the codes

obs	*id*	x_1	x_2	y_1	y_2	p	C
1	1	2000	1750	4000	150	0.3	0
2	1	2000	1750	4000	150	0.7	0
3	1	2000	1750	4000	150	0.5	0
4	2	2250	1500	4000	500	0.9	1
5	3	2000	1600	3850	100	0.9	1
6	4	2500	1000	4500	50	0.7	0
7	4	2250	1500	4000	500	0.3	1

A key variable for the estimation of the RPM is the risk aversion level that equalizes the expected value of lotteries X_i and Y_i. This value is denoted as ω_i and, adopting a CRRA formulation, is defined implicitly by the following equation.

$$p_i \left(\frac{x_{1,i}^{1-\omega_i}}{1-\omega_i} \right) + (1-p_i) \left(\frac{x_{2,i}^{1-\omega_i}}{1-\omega_i} \right) - p_i \left(\frac{y_{1,i}^{1-\omega_i}}{1-\omega_i} \right) - (1-p_i) \left(\frac{y_{2,i}^{1-\omega_i}}{1-\omega_i} \right) = 0.$$

[*] ICREA, Universitat Pompeu Fabra and Barcelona GSE
[†] University of Oxford
[‡] Universitat Pompeu Fabra and Barcelona GSE
[§] November 2016

Our code provides routines to compute ω_i by solving numerically this non-linear equation using the information of the lottery provided in each observation. For the case of Matlab, we make use of the function **fsolve** in the Optimization Toolbox. For Stata, we use the Mata programming language embedded in Stata.[1]

We use data from Andersen, Harrison, Lau, and Rutström (2008) field experiment to illustrate how to use the codes. Their experiment compromised 253 individuals chosen to get a representative sample of the adult Danish population. In their experiment, there were four different risk-aversion choice tasks in the style of the multiple-price lists of Holt and Laury (2002). Each task comprised ten pairs of nested gambles. For every pair of gambles, subjects could either choose one of the gambles, or express indifference between the two. In the latter case, they were told that the experimenter would settle indifference by tossing a fair coin. To test the codes, we pool the observations from all individuals to get a dataset with a sample of 7928 observations.

Technical Details

The point estimators in both Matlab and Stata are computed by maximizing the conditional log-likelihood of the model. In estimation of the RPM, the probability of choosing lottery X_i over lottery Y_i is defined as:

$$P_i(\theta;\, \omega_i) = (1-\kappa)\, F(\lambda(r-\omega_i)) + \kappa\,(1 - F(\lambda(r-\omega_i))).$$

Where r is the population risk-aversion level, λ is a precision parameter, κ is a tremble probability, $\theta = [r,\, \lambda,\, \kappa]$ denotes the vector of parameters in the model and F is a cumulative distribution function.[2] It follows that the log-likelihood function of observation i is defined as

$$\log f(C_i|\omega_i;\, \theta_i) = \begin{cases} \log(P_i) & \text{if } C_i = 0 \\ \log(1-P_i) & \text{if } C_i = 1 \\ \frac{1}{2}\log(P_i) + \frac{1}{2}\log(1-P_i) & \text{if } C_i = -1. \end{cases}$$

Let $\log \mathcal{L}(\theta) = \sum_{i=1}^{n} \log f(C_i|\omega_i;\, \theta_i)$ denote the log-likelihood function of the Random Parameter Model, given our sample of n observations. As usual, the Maximum Likelihood Estimator of θ is given by

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}} \log \mathcal{L}(\theta).$$

In the Stata codes, we use the **ml** command to get $\hat{\theta}$ after specifying the $\log f(C_i|\omega_i;\, \theta_i)$ manually. This powerful command computes the standard deviation of the estimates and allows to easily add explanatory variables to our model, as well as cluster the standard errors by individual.

In the Matlab codes, we maximize $\log \mathcal{L}(\theta)$ numerically using the command **fminunc** of the Optimization Toolbox to get $\hat{\theta}$. The estimated standard deviations are computed using a consistent estimator of the asymptotic covariance matrix of the MLE estimator $\hat{\theta}$. Under some regularity conditions, we have:[3]

$$\sqrt{n}\left(\hat{\theta} - \theta_0\right) \xrightarrow{d} \mathcal{N}\left(0,\, AVAR\left(\hat{\theta}\right)\right),$$

[1] It is important to note that ω_i is not well defined if one of the two lotteries stochastically dominates the other. As an example, there are several observations where lottery Y_i dominates lottery X_i in the dataset used to illustrate the codes. For estimation of the RPM, it is enough to set ω_i to be an arbitrarily large value whenever Y_i dominates X_i. As we will see later, this makes the probability of choosing lottery X_i tend to 0 (plus a tremble probability), giving us the correct log-likelihood for this observation. In the codes provided, the numerical solvers of the nonlinear equation set automatically a large value for ω_i whenever Y_i dominates X_i.

[2] In the codes, we use the CDF of a Logistic distribution (Logit model) and the Normal distribution (Probit model). Furthermore, we estimate $\log\lambda$ instead of λ to keep this parameter positive and use the delta method to recover the standard deviation of $\hat{\lambda}$.

[3] See Section 7.3 of Hayashi (2000) or Section 5.8 in Hamilton (1994).

where θ_0 is the population value of θ and the asymptotic variance covariance matrix $AVAR\left(\hat{\theta}\right)$ is given by

$$AVAR\left(\hat{\theta}\right) = J_1^{-1} J_2 J_1^{-1}$$
$$J_1 = E\left[H\left(C_i;\, \omega_i,\, \theta_0\right)\right] = E\left[\frac{\partial^2 \log f\left(C_i|\omega_i;\, \theta_0\right)}{\partial\theta\partial\theta^T}\right]$$
$$J_2 = E\left[s\left(C_i;\, \omega_i,\, \theta_0\right) s\left(C_i;\, \omega_i,\, \theta_0\right)^T\right] = E\left[\left[\frac{\partial \log f\left(C_i|\omega_i;\, \theta_0\right)}{\partial\theta}\right]\left[\frac{\partial \log f\left(C_i|\omega_i;\, \theta_0\right)}{\partial\theta^T}\right]\right].$$

Under further regularity conditions, each of these matrices can be estimated consistently using their sample counterparts:

$$\widehat{AVAR}\left(\hat{\theta}\right) = \hat{J}_1^{-1} \hat{J}_2 \hat{J}_1^{-1}$$
$$\hat{J}_1 = \frac{1}{n}\sum_{i=1}^{n} \frac{\partial^2 \log f\left(C_i|\omega_i;\, \hat{\theta}\right)}{\partial\theta\partial\theta^T}$$
$$\hat{J}_2 = \frac{1}{n}\sum_{i=1}^{n} \left[\frac{\partial \log f\left(C_i|\omega_i;\, \hat{\theta}\right)}{\partial\theta}\right]\left[\frac{\partial \log f\left(C_i|\omega_i;\, \hat{\theta}\right)}{\partial\theta^T}\right].$$

The standard deviations are computed as the finite sample approximation $\hat{\sigma}\left(\hat{\theta}_k\right) = \sqrt{\frac{1}{n}\left[\widehat{AVAR}\left(\hat{\theta}\right)\right]_{kk}}$. In applications, it is usual to cluster the standard errors to take into account potential correlation on the responses of the same individual. Assume that there are m clusters of observations, denoted as $G_1, G_2, \ldots, G_m$. We can compute the clustered-robust standard errors using the same formula as before but replacing $\hat{J}_2$ with

$$\hat{J}_2^* = \frac{1}{n}\sum_{s=1}^{m} \left[\sum_{j\in G_s} \frac{\partial \log f\left(C_i|\omega_i;\, \hat{\theta}\right)}{\partial\theta}\right]\left[\sum_{j\in G_s} \frac{\partial \log f\left(C_i|\omega_i;\, \hat{\theta}\right)}{\partial\theta^T}\right].$$

The Matlab codes compute the clustered-robust standard errors of the RPM using the previous formulas.

References

Andersen, S., G. W. Harrison, M. I. Lau, and E. E. Rutström (2008): "Eliciting Risk and Time Preferences," *Econometrica*, 76(3), 583–618. 1, 2

Apesteguia, J., and M. A. Ballester (Forthcoming): "Monotone Stochastic Choice Models: The Case of Risk and Time Preferences," *Journal of Political Economy*. 1

Hamilton, J. (1994): *Time Series Analysis*. Princeton University Press. 2

Hayashi, F. (2000): *Econometrics*. Princeton University Press. 2

Holt, C. A., and S. K. Laury (2002): "Risk Aversion and Incentive Effects," *American Economic Review*, 92(5), 1644–1655. 1, 2