Chapter 4. Conditional Choice Probability (CCP) estimation

Joan Llull

Advanced Econometric Methods II
Master in Economics and Finance
Barcelona GSE

I. Introduction

The estimation of dynamic discrete choice models using the approaches described in the previous chapter is often challenging computationally, and sometimes even unfeasible. In this chapter, we review alternative estimation techniques that dramatically reduce the computational burden of the estimation. These methods avoid having to solve the dynamic programming problem (DP) in each iteration of the estimation algorithm. They are not as efficient as full solution methods, but they are much faster to implement. Additionally, the process of deriving the specific representation that allows estimation without solving the DP provides more transparent insights of the sources of variation in the data that identify the parameters of the model. Finally, the faster estimation algorithms allow the researcher to carry robustness checks. These techniques allow the estimation of problems that would be otherwise out of reach, like dynamic games or non-stationary environments in which the full time horizon is not covered in the data and the researcher is unwilling to make assumptions regarding how expectations are formed outside of the sample.

The methods that we discuss here build on the seminal work of Hotz and Miller (1993). These authors noted that individual choices contain rich information on individual expectations about future outcomes. Intuitively, the individuals have already solved their optimization problem, so their decisions, as reflected in their CCPs, are informative of their value functions. Thus, they show that there exists a mapping between conditional choice probabilities (CCPs) and alternative-specific value functions $v_{jt}(\boldsymbol{x}_t)$ that, under general conditions, can be inverted. Representing the mapping as a function of the CCPs instead of the value functions, and armed with nonparametric estimates of the CCPs avoids the need for a solution of the DP in estimation.

II. Conditional choice probability (CCP) representation

A. *Models requiring only one-period-ahead choice probabilities*

We start analyzing the CCP representation in the easiest possible case, in which the conditional value functions are expressed only as a function of one-period-ahead conditional choice probabilities. We begin with the Rust example, introduced in the previous chapter, and then we generalized to the subclass of models that share this type of representation.

The Rust example To illustrate how this method works in practice, consider the fixed point in the Rust engine replacement example in the previous chapter:

$$v_j(x_t) = u_j(x_t) + \beta \sum_{x \in X} \ln \left(\sum_{h \in \mathcal{D}} \exp\{v_h(x)\} \right) F^j_{x,x_t} + \beta\gamma. \tag{1}$$

Given the Type-I extreme value assumption, the CCPs are of logit type:

$$p_{jt}(\boldsymbol{x}_t) = \frac{e^{v_{jt}(\boldsymbol{x}_t)}}{\sum_{h \in \mathcal{D}} e^{v_{ht}(\boldsymbol{x}_t)}}. \tag{2}$$

Without loss of generality, we can rewrite them in terms of a base category. In this context, it is convenient to use replacement as the base category:

$$p_1(x) = \frac{1}{1 + e^{v_0(x) - v_1(x)}} \quad \text{and} \quad p_0(x) = \frac{e^{v_0(x) - v_1(x)}}{1 + e^{v_0(x) - v_1(x)}}. \tag{3}$$

Given this:

$$-\ln p_1(x) = \ln\left(1 + e^{v_0(x) - v_1(x)}\right) = \ln \left(\sum_{h \in \mathcal{D}} \exp\{v_h(x) - v_1(x)\} \right). \tag{4}$$

Rewriting Equation (1):

$$\begin{aligned} v_j(x_t) &= u_j(x_t) + \beta \sum_{x \in X} \ln \left(e^{v_1(x)} \sum_{h \in \mathcal{D}} \exp\{v_h(x) - v_1(x)\} \right) F^j_{x,x_t} + \beta\gamma \\ &= u_j(x_t) + \beta \sum_{x \in X} \left(v_1(x) - \ln p_1(x)\right) F^j_{x,x_t} + \beta\gamma \\ &= u_j(x_t) + \beta v_1(0) - \beta \sum_{x \in X} \ln p_1(x) F^j_{x,x_t} + \beta\gamma. \end{aligned} \tag{5}$$

The last step uses the fact that $v_1(x) = v_1(0)\ \forall x$, as the utility of replacement does not depend on mileage. Noting that $\sum_{x \in X} \ln p_1(x) F^1_{x,x_t} = \ln p_1(0)$, as F^1_{x,x_t} is degenerate, we can finally write:

$$\begin{aligned} v_0(x_t) - v_1(x)_t &= u_0(x_t) - u_1(x_t) + \beta \left(\ln p_1(0) - \sum_{x \in X} \ln p_1(x) F^0_{x,x_t} \right) \\ &= \theta_R - \theta_M x_t + \beta \left(\ln p_1(0) - \sum_{x \in X} \ln p_1(x) F^0_{x,x_t} \right). \end{aligned} \tag{6}$$

We can obtain non-parametric estimates of $p_1(x)$ from the data, for instance from relative frequencies:

$$\hat{p}_1(x) = \frac{\sum_{\mathrm{i}=1}^{\mathrm{N}} \sum_{t=1}^{T} d_{1it}\, \mathbb{1}\{x_{it} = x\}}{\sum_{\mathrm{i}=1}^{\mathrm{N}} \sum_{t=1}^{T} \mathbb{1}\{x_{it} = x\}}. \tag{7}$$

Finally, we can replace $p_1(x)$ by $\hat{p}_1(x)$ in Equation (6), and replace this expression into Equation (3), which describes a straightforward binary logit (as in the two-step Rust algorithm, we typically estimate F^0_{x,x_t} separately in a first stage).

General representation Similar ideas can be implemented for finite horizon, and with different assumptions on the residuals, as long as a mapping between CCP and conditional value functions exists and it is invertible. In particular:

$$\begin{aligned} p_{Jt}(\boldsymbol{x}_t) &\equiv \int d^*_{Jt}(\boldsymbol{x}_t, \boldsymbol{\varepsilon}_t) dF_\varepsilon(\boldsymbol{\varepsilon}_t) \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\varepsilon_{Jt} + v_{Jt}(\boldsymbol{x}_t) - v_{J-1t}(\boldsymbol{x}_t)} \dots \int_{-\infty}^{\varepsilon_{Jt} + v_{Jt}(\boldsymbol{x}_t) - v_{1t}(\boldsymbol{x}_t)} dF_\varepsilon(\boldsymbol{\varepsilon}_t) \\ &\equiv Q_J(v_{Jt}(\boldsymbol{x}_t) - v_{1t}(\boldsymbol{x}_t), ..., v_{Jt}(\boldsymbol{x}_t) - v_{J-1t}(\boldsymbol{x}_t)). \end{aligned} \tag{8}$$

We can write analogous mappings $Q_j(.)$ for $j = 1, ..., J-1$. The main theorem in Hotz and Miller (1993) specifies that if we define $\boldsymbol{p}_t$ such that $p_{jt} > 0\ \forall j$, then there exists a real-valued function $\psi_j(\boldsymbol{p})$ for every $j \in \mathcal{D}$ such that:

$$\psi_j(\boldsymbol{p}_t(\boldsymbol{x}_t)) \equiv V_t(\boldsymbol{x}_t) - v_{jt}(\boldsymbol{x}_t). \tag{9}$$

This formulation of the theorem is drawn from Arcidiacono and Miller (2011). The original formulation specifies $\psi_j(\boldsymbol{p}) \equiv Q_j^{-1}(\boldsymbol{p})$ as a function of differences in conditional value functions. In our example $\psi_j(\cdot)$ has the following form:

$$\psi_j(\boldsymbol{p}(x_t)) = -\ln p_j(x_t) + \gamma, \tag{10}$$

where γ is the Euler-Mascheroni constant.

Note that Equation (9) can be rewritten as:

$$V_t(\boldsymbol{x}_t) = v_{jt}(\boldsymbol{x}_t) + \psi_j(\boldsymbol{p}_t(\boldsymbol{x}_t)). \tag{11}$$

This expression is closely connected to the selection models reviewed in the Microeconometrics course. For example, in the Heckman selection model we first estimated the probability to be in the sample and then we computed the $\lambda(\cdot)$ correction as a function of the parameters governing this probability. The term $\psi_j(\boldsymbol{p}_t(\boldsymbol{x}_t))$ plays exactly the same role here. Thus, (11) tells us that the ex-ante value function can be expressed as the sum of the deterministic part of the utility associated to one of the choices plus a correction factor that accounts for the fact that $\mathbb{E}[\varepsilon_{jt}|d_{jt} = 1]$ (i.e. the expectation of the random component given that option j is selected) is not zero. Intuitively, the less likely is alternative j to be chosen (the lower the CCP) the more positive the draw of ε_{jt} needs to be for j to be the optimal choice.

This mapping can help us in rewriting the DP into a known function of data, parameters, and CCPs. In the previous chapter, we defined conditional value functions as:

$$v_{jt}(\boldsymbol{x}_t) \equiv u_{jt}(\boldsymbol{x}_t) + \beta \int V_{t+1}(\boldsymbol{x}_{t+1}) dF_x(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, j). \tag{12}$$

Substituting for $V_{t+1}(x_{t+1})$ using Equation (11) (for any alternative k) yields:

$$v_{jt}(\boldsymbol{x}_t) \equiv u_{jt}(\boldsymbol{x}_t) + \beta \int \left[v_{kt+1}(\boldsymbol{x}_{t+1}) + \psi_k(\boldsymbol{p}_{t+1}(\boldsymbol{x}_{t+1}))\right] dF_x(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, j). \tag{13}$$

We can repeat this procedure ad infinitum, substituting $v_{kt+1}(\boldsymbol{x}_{t+1})$ using the above expression, and then appealing to the inversion theorem of Hotz and Miller (1993)/Arcidiacono and Miller (2011) in Equation (11). This would lead to a representation that would not require solving for the value functions, provided we have nonparametric estimates of the CCPs.

In the next section, we show that, for a large class of problems, dealing with the remaining conditional value function is surprisingly straightforward. The key to the argument is that the researcher can choose to which choice k to make the future term relative. Because the choices are made by differencing each possible pair of alternatives, a clever choice of alternatives k in the recursive forward substitution can allow the difference in future utility terms across two choices to be equal and, hence be differenced out.

Taking this idea to the extreme, the Rust example belongs to a class of models in which this is so straightforward that only one-period-ahead CCPs are needed in the estimation. In particular, this class of models include those in which either there is a *terminal action* (no further decisions are made after this decision), or a *renewal action* (state variables are re-initiated after this action, implying that the choice in the previous period becomes irrelevant). An example of the first would be an occupational choice model featuring a retirement choice, and the Rust model is an example of the second. Let R denote the terminal/renewal action. In the terminal action, we only require one-period-ahead CCPs because $v_{Rt+1}(\boldsymbol{x}_t)$ does not have a continuation value. Thus, replacing $V_{t+1}(\boldsymbol{x}_{t+1})$ by $v_{Rt}(\boldsymbol{x}_{t+1}) + \psi_R(\boldsymbol{p}_{t+1}(\boldsymbol{x}_{t+1}))$

we already eliminate the need to solve for the value functions in estimation.

The renewal action has the property that the state vector is reset in such a way that the previous choices become irrelevant. Specifically, it implies that, for any pair of choices d_t and d_t':

$$\int f_x(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, d_t) f_x(\boldsymbol{x}_{t+2}|\boldsymbol{x}_{t+1}, R) d\boldsymbol{x}_{t+1} = \int f_x(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, d_t') f_x(\boldsymbol{x}_{t+2}|\boldsymbol{x}_{t+1}, R) d\boldsymbol{x}_{t+1}. \tag{14}$$

In words, the distribution of $\boldsymbol{x}_{t+2}$ after choosing renewal in $t+1$ is the same regardless of whether the individual initially chose d_t or d_t' in t. This implies that, when comparing two choices, we can replace the ex-ante value function at $t+2$ by its representation for the choice R, such that the resulting terms are common to all choices and cancel out when writing the CCPs. Substituting (12) into (13), and evaluating the second period at $k = R$, we obtain:

$$\begin{aligned}
v_{jt}(\boldsymbol{x}_t) &= u_{jt}(\boldsymbol{x}_t) + \beta \int \left[v_{Rt+1}(\boldsymbol{x}_{t+1}) + \psi_R(\boldsymbol{p}_{t+1}(\boldsymbol{x}_{t+1})) \right] dF_x(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, j) \\
&= u_{jt}(\boldsymbol{x}_t) + \beta \int \left[\begin{array}{l} u_{Rt+1}(\boldsymbol{x}_{t+1}) \\ +\beta \int V_{t+2}(\boldsymbol{x}_{t+2}) dF_x(\boldsymbol{x}_{t+2}|\boldsymbol{x}_{t+1}, R) \\ +\psi_R(\boldsymbol{p}_{t+1}(\boldsymbol{x}_{t+1})) \end{array} \right] dF_x(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, j) \\
&= u_{jt}(\boldsymbol{x}_t) + \beta \int \left[u_{Rt+1}(\boldsymbol{x}_{t+1}) + \psi_R(\boldsymbol{p}_{t+1}(\boldsymbol{x}_{t+1})) \right] dF_x(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, j) \qquad (15) \\
&\quad + \beta^2 \int \int V_{t+2}(\boldsymbol{x}_{t+2}) f_x(\boldsymbol{x}_{t+2}|\boldsymbol{x}_{t+1}, R) f_x(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, j) d\boldsymbol{x}_{t+2} d\boldsymbol{x}_{t+1}.
\end{aligned}$$

Given the renewal property described in (14), the last term of (15) is invariant to the choice j. Let alternative 1 denote the base category. The probability of choosing option j, depends on the pairwise comparisons $v_{jt}(\boldsymbol{x}_t) - v_{1t}(\boldsymbol{x}_t)$ for each j:

$$p_{jt}(\boldsymbol{x}_t) = \frac{\exp(v_{jt}(\boldsymbol{x}_t) - v_{1t}(\boldsymbol{x}_t))}{\sum_{h \in \mathcal{D}} \exp(v_{ht}(\boldsymbol{x}_t) - v_{1t}(\boldsymbol{x}_t))}. \tag{16}$$

Hence:

$$\begin{aligned}
& v_{jt}(\boldsymbol{x}_t) - v_{1t}(\boldsymbol{x}_t) \qquad (17) \\
&= (u_{jt}(\boldsymbol{x}_t) - u_{1t}(\boldsymbol{x}_t)) + \beta \left\{ \begin{array}{l} \int \left[u_{Rt+1}(\boldsymbol{x}_{t+1}) + \psi_R(\boldsymbol{p}_{t+1}(\boldsymbol{x}_{t+1})) \right] dF_x(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, j) \\ - \int \left[u_{Rt+1}(\boldsymbol{x}_{t+1}) + \psi_R(\boldsymbol{p}_{t+1}(\boldsymbol{x}_{t+1})) \right] dF_x(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, 1) \end{array} \right\},
\end{aligned}$$

for every $j \in \mathcal{D}$, which only requires the one-period-ahead CCPs.

B. Finite dependence

The ideas presented above for the case of renewal action are generalizable to a larger class of problems, except that multiple-periods-ahead CCPs as opposed to one-period-ahead are needed. In these models, ρ periods (and particular sequences of choices) are needed to reset the system. After the ρ periods, the specified combination of actions across the two paths undo the dependence on the initial choice. In that case, we say that the problem exhibits *finite dependence.* First introduced in Altuğ and Miller (1998), this idea is generalized by Arcidiacono and Miller (2011). As in the renewal/termination action case, we begin illustrating the idea with an example and then generalize.

Occupational choice example Consider a simplified version of the occupational choice model described in Keane and Wolpin (1997). In this version, individuals choose whether to say home $d_t = H$ or to work $d_t = W$. The per-period utility function for working depends on the wage, which is a function of experience (the flow of utility from the home alternative is normalized to zero). The only (observable) state variable is experience, which is degenerate, and does not depreciate. Therefore, if an individual does not work in the current period and works in the next period, or she works in this period and does not work in the next period, two periods from now, her level of experience. This is exactly what we exploit to establish, in this case, still one-period ahead finite dependence.

Substituting (12) into (13), and evaluating the second period at $k = H$, we express the working utility as:

$$v_{Wt}(x_t) = u_{Wt}(x_t) + \beta \left[u_{Ht+1}(x_t+1) + \psi_H(\boldsymbol{p}_{t+1}(x_t+1)) \right] + \beta^2 V_{t+2}(x_t+1). \tag{18}$$

Likewise, the utility of home can be expressed as:

$$v_{Ht}(x_t) = u_{Ht}(x_t) + \beta \left[u_{Wt+1}(x_t) + \psi_W(\boldsymbol{p}_{t+1}(x_t)) \right] + \beta^2 V_{t+2}(x_t+1). \tag{19}$$

The CCPs depend on the difference between the two conditional value functions, which can be expressed as:

$$v_{Wt}(x_t) - v_{Ht}(x_t) = u_{Wt}(x_t) - u_{Ht}(x_t) + \beta \left[\begin{array}{l} u_{Ht+1}(x_t+1) - u_{Wt+1}(x_t) \\ +\psi_H(\boldsymbol{p}_{t+1}(x_t+1)) - \psi_W(\boldsymbol{p}_{t+1}(x_t)) \end{array} \right], \tag{20}$$

and only depends on the ρ-periods ahead CCPs, where, in this case, $\rho = 1$.

General representation In the example above, finite dependence is achieved exactly, as the state vector two periods ahead coincides under both paths. This is a sufficient but not necessary condition for finite dependence, which can be established for stochastic state vectors as long as the same state results in expectation. To see it, we define an expression for the cumulative probability of being in a particular state given a particular decision sequence and an initial state. Let $\{d'_t, d'_{t+1}, ..., d'_{t+\rho}\}$ define a sequence of decisions from t to $t+\rho$ (no need to be optimal). For each $\tau \in \{t, ..., t+\rho\}$, denote $\kappa'_\tau(\boldsymbol{x}_{\tau+1}|\boldsymbol{x}_t)$ as the cumulative probability of being in state $\boldsymbol{x}_{\tau+1}$ given the specified decision sequence, recursively defined as:

$$\kappa'_\tau(\boldsymbol{x}_{\tau+1}|\boldsymbol{x}_t) \equiv \begin{cases} f(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, d'_t) & \text{if } \tau = t \\ \int f(\boldsymbol{x}_{\tau+1}|\boldsymbol{x}_\tau, d'_\tau)\kappa'_{\tau-1}(\boldsymbol{x}_\tau|\boldsymbol{x}_t)d\boldsymbol{x}_\tau & \text{if } \tau > t. \end{cases} \tag{21}$$

Using this expression, we can telescope (13) ρ-periods ahead to obtain:

$$\begin{aligned} v_{\{d'_t\}t}(\boldsymbol{x}_t) = u_{\{d'_t\}t}(\boldsymbol{x}_t) + \sum_{\tau=t+1}^{t+\rho} \beta^{\tau-t} \int \left[u_{\{d'_\tau\}\tau}(\boldsymbol{x}_\tau) + \psi_{\{d'_\tau\}}(\boldsymbol{p}_\tau(\boldsymbol{x}_\tau))\right] \kappa'_{\tau-1}(\boldsymbol{x}_\tau|\boldsymbol{x}_t)d\boldsymbol{x}_\tau \\ + \beta^{\rho+1} \int V_{t+\rho+1}(\boldsymbol{x}_{t+\rho+1})\kappa'_{t+\rho}(\boldsymbol{x}_{t+\rho+1}|\boldsymbol{x}_t)d\boldsymbol{x}_{t+\rho+1}. \end{aligned} \tag{22}$$

Now we can define an alternative sequence of decisions $\{d''_t, d''_{t+1}, ..., d''_{t+\rho}\}$ that lead the individual to the same state in expectation, or, equivalently, such that:

$$\kappa'_{t+\rho}(\boldsymbol{x}_{t+\rho+1}|\boldsymbol{x}_t) = \kappa''_{t+\rho}(\boldsymbol{x}_{t+\rho+1}|\boldsymbol{x}_t) \text{ for all } \boldsymbol{x}_{t+\rho+1}. \tag{23}$$

Clearly, in the difference $v_{\{d''_t\}t}(\boldsymbol{x}_t) - v_{\{d'_t\}t}(\boldsymbol{x}_t)$, the last term of (22) cancels, and the resulting expression only depends on flow payoffs and CCPs up to period $t+\rho$.

C. *Infinite-horizon stationary settings*

Recovering a representation of the value functions based on the CCPs is still very straightforward in stationary infinite-horizon settings in the absence of finite dependence. To do so, it is convenient to follow the matrix representation of Aguirregabiria and Mira (2002), which we also use below to explain their nested-fixed point algorithm. Define $\varepsilon^*_{jt}(\boldsymbol{x}_t) \equiv \mathbb{E}[\varepsilon_{jt}|d^*_t = j, \boldsymbol{x}_t]$, where d^*_t denotes the optimal choice. The ex-ante value function can be written as:

$$\begin{aligned} V(\boldsymbol{x}_t) &= \mathbb{E}[\max_d\{v_j(\boldsymbol{x}_t) + \varepsilon_{jt}\}] \\ &= \sum_{j\in\mathcal{D}} p(j|\boldsymbol{x}_t) \left[v_j(\boldsymbol{x}_t) + \varepsilon^*_{jt}(\boldsymbol{x}_t)\right] \\ &= \sum_{j\in\mathcal{D}} p(j|\boldsymbol{x}_t) \left[u_j(\boldsymbol{x}_t) + \beta \sum_{\boldsymbol{x}\in\mathcal{X}} V(\boldsymbol{x})f(\boldsymbol{x}|\boldsymbol{x}_t, j) + \varepsilon^*_{jt}(\boldsymbol{x}_t)\right], \end{aligned} \tag{24}$$

where $\mathcal{X}$ denotes the set that includes the X possible values that $\boldsymbol{x}_{t+1}$ can take. Note that we do not use subscript in $V(\cdot)$ to reflect the infinite-horizon stationary nature of the setting. Now we can express each of the components of the previous equation in matrix form:

$$\boldsymbol{V} \equiv \begin{bmatrix} V(\boldsymbol{x}^{(1)}) \\ \vdots \\ V(\boldsymbol{x}^{(X)}) \end{bmatrix}, \qquad \boldsymbol{u}_j \equiv \begin{bmatrix} u_j(\boldsymbol{x}^{(1)}) \\ \vdots \\ u_j(\boldsymbol{x}^{(X)}) \end{bmatrix}, \qquad \boldsymbol{\varepsilon}^*_{jt} \equiv \begin{bmatrix} \varepsilon^*_{jt}(\boldsymbol{x}^{(1)}) \\ \vdots \\ \varepsilon^*_{jt}(\boldsymbol{x}^{(X)}) \end{bmatrix} \tag{25}$$

$$\boldsymbol{p}_j \equiv \begin{bmatrix} p(j|\boldsymbol{x}^{(1)}) \\ \vdots \\ p(j|\boldsymbol{x}^{(X)}) \end{bmatrix}, \qquad F_j \equiv \begin{bmatrix} f(\boldsymbol{x}^{(1)}|\boldsymbol{x}^{(1)}, j) & \dots & f(\boldsymbol{x}^{(X)}|\boldsymbol{x}^{(1)}, j) \\ \vdots & \ddots & \vdots \\ f(\boldsymbol{x}^{(1)}|\boldsymbol{x}^{(X)}, j) & \dots & f(\boldsymbol{x}^{(X)}|\boldsymbol{x}^{(X)}, j) \end{bmatrix},$$

so that (24) reads as:

$$\boldsymbol{V} = \sum_{j \in \mathcal{D}} \boldsymbol{p}_j \circ [\boldsymbol{u}_j + \beta F_j \boldsymbol{V} + \boldsymbol{\varepsilon}^*_{jt}], \tag{26}$$

where $\circ$ denotes the Hadamard product (or element-by-element multiplication). Solving for $\boldsymbol{V}$ in the above expression yields:

$$\boldsymbol{V} = \left(I_X - \beta \sum_{j \in \mathcal{D}} \boldsymbol{p}_j \iota'_X \circ F_j \right)^{-1} \left(\sum_{j \in \mathcal{D}} \boldsymbol{p}_j \circ [\boldsymbol{u}_j + \boldsymbol{\varepsilon}^*_{jt}] \right), \tag{27}$$

where I_X denotes the size X identity matrix and ι_X denotes the size X vector of ones. This matrix expression is operational, in practice, in any standard matrix software, and avoids the solution of the DP in estimation. Furthermore, it is useful in the estimation of dynamic games, as we review later in the course.

III. Estimation methods

So far we have analyzed the case in which $\boldsymbol{\varepsilon}_t$ is i.i.d. across time, and there is no persistent unobserved heterogeneity. The estimation methods that we describe in this section are applicable to this type of models. Estimation occurs in two stages. In the first stage, we estimate CCPs and transition functions for the state variables. In the second stage, we form the value functions using the CCPs estimated in the first stage and estimate the structural parameters. Despite the CCPs are only approximated, CCP estimators of the structural parameters are $\sqrt{N}$-consistent and asymtotically normal under standard regularity conditions. However, the approximation error of the CCPs can introduce small sample bias in the structural parameter estimates. This bias can be mitigated by updating the initial nonparametric estimates with the CCPs implied by the structural model, as we discuss in Section III.D below.

A. CCPs and transition functions

With unlimited data, both CCPs and transition functions can be estimated nonparametrically using simple bin estimators. For example, the CCPs would be estimated as:

$$\hat{p}(d_t = d|\boldsymbol{x}_t = \boldsymbol{x}) = \frac{\sum_{\mathrm{i}=1}^{\mathrm{N}} \sum_{t=1}^{T} \mathbb{1}\{d_{it} = d\}\, \mathbb{1}\{\boldsymbol{x}_{it} = \boldsymbol{x}\}}{\sum_{\mathrm{i}=1}^{\mathrm{N}} \sum_{t=1}^{T} \mathbb{1}\{\boldsymbol{x}_{it} = \boldsymbol{x}\}}. \tag{28}$$

However, in reality data limitations and continuous state variables prevent us from doing it this way. In this case, some smoothing is needed, either through nonparametric kernels, basis functions, or flexibly specified logits or probits.

B. Estimating the structural parameters

The estimation of the structural parameters under finite dependence, infinite-horizon stationary settings, or shortly-lived finite horizon settings is straightforward. The CCP representation that we discussed in the previous section together with the CCP estimates obtained in the first stage lead to simple expressions that can either be included in a likelihood estimation or in a GMM estimation. To illustrate this, take the Rust example. Estimation by maximum likelihood would follow a standard logit approach. This is, following with the expression obtained in (6) the likelihood would be expressed as:

$$\mathcal{L}_{\mathrm{N}} = \sum_{i=1}^{N} \sum_{t=1}^{T} d_{1it} \ln p_1(x_{it}) + (1 - d_{1it}) \ln[1 - p_1(x_{it})], \tag{29}$$

where:

$$p_1(x_{it}) = \frac{1}{1 + \exp\left\{\theta_R - \theta_M x_{it} + \beta\left(\ln \hat{p}_1(0) - \sum_{x \in X} \ln \hat{p}_1(x) \hat{F}^0_{x,x_t}\right)\right\}}. \tag{30}$$

Alternatively, we can use a GMM (in this case simple regression) approach. From (4) we know:

$$\begin{aligned} -\ln p_1(x_{it}) &= \ln\left[1 + \exp(v_0(x_{it}) - v_1(x_{it}\right] \\ &= \ln\left[1 + \exp\left\{\theta_R - \theta_M x_{it} + \beta\left(\ln \hat{p}_1(0) - \sum_{x \in X} \ln \hat{p}_1(x) \hat{F}^0_{x,x_t}\right)\right\}\right]. \end{aligned} \tag{31}$$

Rearranging the terms in the above expression yields:

$$\ln\left(\frac{1 - \hat{p}_1(x_{it})}{\hat{p}_1(x_{it})}\right) = \theta_R - \theta_M x_{it} + \beta\left(\ln \hat{p}_1(0) - \sum_{x \in X} \ln \hat{p}_1(x) \hat{F}^0_{x,x_t}\right). \tag{32}$$

Even though this expression describes a relation that should hold exactly if $\hat{p}_1(\cdot)$ and $\hat{F}$ where replaced by population values, there is an econometric error in sample because of the estimation of the CCPs and the transition functions. One can exploit a large set of orthogonality conditions that this econometric error satisfies and estimate this expression using GMM. As noted in Llull (2018b) if one have access to estimates of the CCPs obtained with a different dataset, those serve as good instruments for the expressions in the right hand side of the expression that are computed with the CCPs, like the one in parenthesis in (32), to correct for the attenuation bias given by the measurement error in the (right-hand-side) CCPs, and the potential biases that may arise from the correlation between the measurement error of the term in the left hand side.

C. Forward simulation methods

In problems that do not exhibit finite dependence and are either long finite horizon or short horizon but with a large state space, even computing the value functions using the CCP representation can be cumbersome. Hotz, Miller, Sanders, and Smith (1994) propose an alternative method that avoids having to visit all future states for all individuals. This method is based on forward simulation, based on the transition functions and, potentially, the CCPs.

The idea is the following. The continuation value $\int V(\boldsymbol{x}_{t+1})dF_x(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, j)$ involves integration over all possible future states of the world (paths) and all possible choices. This is costly, even with the CCP representation if there are many paths for which we need to compute $u_j(\cdot) + \psi_j(\cdot)$. For example, in the Rust example, even though there is only one state variable, next period's state can take one of three values, two periods ahead it can take six values, and so on. For each individual, we should compute the value function in each of these possible paths and then integrate over the probabilities of taking each of these paths. Alternatively, forward simulation methods only compute them for a finite number of draws of these future paths for each individual. Then, when averaging across individuals, we obtain an approximation of the future expectation.

There are two versions of the method. In the most basic one, given the individual's current state, we simulate a choice using the CCPs (pile up the CCPs into a sort of cdf, draw a uniform $U[0, 1]$, and pick the choice based on this "cdf"). Given this choice, use the transition function to draw a value for the state next period. Then, proceed recursively until T in finite horizon problems or until the future is discounted enough to be unimportant in infinite horizon problems. The second version selects the path of choices by strategically choosing k in the CCP representation $V_{t+1}(\boldsymbol{x}_{t+1}) = v_{kt}(\boldsymbol{x}_{t+1}) + \psi_k(p_{t+1}(\boldsymbol{x}_{t+1}))$, and the path for the state variables is drawn from the estimated transition function for the state variables.

D. Aguirregabiria and Mira's iterative approach

Aguirregabiria and Mira (2002) propose nested algorithm that embeds Rust's nested fixed point (NFXP) and Hotz-Miller's CCP estimation as special cases. This method is a nested algorithm that swaps the order of the nest. In particular, it uses the CCP representation and non-parametric estimates of the CCPs (as in Hotz-Miller) to start the algorithm, and to recover a first estimation of the parameters. For these parameter estimates, which are consistent estimates, we then solve the model to calculate new estimates of CCPs, also consistent. Then we feed the new CCPs back into the first step and repeat the process K times, until reaching convergence in parameters and CCPs. This estimator delivers Hotz-Miller's CCP estimation when $K = 0$, and Rust's nested fixed point algorithm when $K \rightarrow \infty$ (i.e., when we iterate until convergence). Note that estimates are consistent at any $k = 0, ..., K$ iteration, so technically, there is no need to iterate until convergence.

This is a swapping of the nested fixed point algorithm. To see it, rewrite (26) to emphasize the dependence of the CCPs on the value functions, $\boldsymbol{p}_j(\boldsymbol{V})$:

$$\boldsymbol{V} = \sum_{j \in \mathcal{D}} \boldsymbol{p}_j(\boldsymbol{V}) \circ [\boldsymbol{u}_j + \beta F_j \boldsymbol{V} + \boldsymbol{\varepsilon}^*_{jt}]. \tag{33}$$

Rust's nested fixed point describes an algorithm with an inner loop that first finds V as a fixed point in this equation, and an outer loop updates the parameters. The Aguirregabiria and Mira's Nested Pseudo-Likelihood algorithm (NPL) reverts the order of the nesting. The CCPs are mappings of the value function:

$$\boldsymbol{p} = \Lambda(\boldsymbol{V}). \tag{34}$$

Rewriting (27) as:

$$\boldsymbol{V}(\boldsymbol{p}) = \left(I_X - \beta \sum_{j \in \mathcal{D}} \boldsymbol{p}_j \iota'_X \circ F_j \right)^{-1} \left(\sum_{j \in \mathcal{D}} \boldsymbol{p}_j \circ [\boldsymbol{u}_j + \boldsymbol{\varepsilon}^*_{jt}] \right), \tag{35}$$

which implies that $\boldsymbol{p} = \Lambda(\boldsymbol{V}(\boldsymbol{p})) \equiv \Psi(\boldsymbol{p})$ describes a fixed point in the CCPs. Therefore, Aguirregabiria and Mira swap the order of the nesting, with the fixed point (in CCPs) in the outside loop and the update of the parameters in the inner loop.

E. Results in the Rust example

We now update the estimates of the Rust example from previous chapter to include estimates with the different methods described in this chapter. For the estimation, we fix $\beta = 0.99$. The results obtained using each of the methods are:

Method	Parameter	Group 1, 2, 3	Group 4	Group 1, 2, 3, 4
NFXP	θ_R	11.87	10.12	9.75
		(1.95)	(1.36)	(0.89)
	θ_M	5.02	1.18	1.37
		(1.40)	(0.28)	(0.24)
CCP	θ_R	11.76	10.21	9.62
		(0.91)	(0.71)	(0.45)
	θ_M	-4.99	1.16	-3.95
		(22.49)	(13.48)	(9.35)
NPL	θ_R	11.72	10.19	9.59
		(0.91)	(0.71)	(0.45)
	θ_M	-2.20	1.43	-2.11
		(22.48)	(13.46)	(9.35)

Courtesy of José García-Louzao, Sergi Marin Arànega, Alex Tagliabracci, and Alessandro Ruggieri, who replicated Rust's paper for the replication exercise in the Microeconometrics IDEA PhD course in Fall 2014.

As discussed above, the CCP estimation can save important amounts of time, but this comes at the cost of losing precision. In this example, this seems to be not very relevant in the estimation of θ_R, but quite important for the estimation of θ_M. Importantly, the estimation of the CCPs includes estimation error, and, if this is substantial, it can create small sample biases in the estimation. The following table shows a Monte-Carlo simulation in which we check the amount of time used by each estimator (only NFXP and CCP), and the estimated parameters with different sample sizes. The exercise is based on 50 replications, and the running time is cumputed based on a MacBookPro with 2.5GHz Intel Core i5, and 8GB RAM:

N.obs.	Algorithm	θ_R		θ_M		Time (sec.)
	DGP	9.74		2.69		
$N = 100$	NFXP	10.07	(0.90)	2.83	(0.39)	157.56
	CCP	10.48	(0.40)	-2.17	(6.91)	0.56

$N = 1,000$	NFXP	9.78	(0.28)	2.67	(0.12)	160.57
	CCP	10.04	(0.14)	2.19	(2.55)	47.84
$N = 5,000$	NFXP	9.93	(0.11)	2.79	(0.05)	400.31
	CCP	10.09	(0.05)	2.92	(0.93)	59.11
$N = 10,000$	NFXP	9.85	(0.08)	2.73	(0.04)	1,030.70
	CCP	9.99	(0.04)	2.64	(0.66)	121.38
$N = 25,000$	NFXP	9.78	(0.05)	2.67	(0.02)	1,070.20
	CCP	9.92	(0.02)	2.74	(0.41)	133.38

Courtesy of José García-Louzao, Sergi Marin Arànega, Alex Tagliabracci, and Alessandro Ruggieri, who replicated Rust's paper for the replication exercise in the Microeconometrics IDEA PhD course in Fall 2014. Simulations executed with a MacBookPro, 2.5GHz Intel Core i5, 8GB RAM (50 replications, asymptotic standard errors from the first iteration are in parentheses).

Clearly, the results show very important gains by CCP in computation time, and the loss of precission diminishes substantially as sample size increases.

IV. Extensions: unobserved heterogeneity and competitive equilibrium models

A. Unobserved heterogeneity

The expectation-maximization (EM) algorithm. As discussed in the previous chapter, the standard approach to introduce unobserved heterogeneity in dynamic discrete choice models uses mixture distributions, as proposed by Heckman and Singer (1984). Given our assumptions, the log-likelihood can be written as:

$$\mathcal{L}_{\mathrm{N}}(\boldsymbol{\theta}, \boldsymbol{\pi}) = \sum_{\mathrm{i}=1}^{\mathrm{N}} \ln \left[\sum_{k=1}^{K} \pi_{k|\boldsymbol{x}_{i1}} \prod_{t=1}^{T} f_t(d_{it}, \boldsymbol{x}_{it+1} | \boldsymbol{x}_{it}, \boldsymbol{\omega}^k; \boldsymbol{\theta}) \right]. \tag{36}$$

Note that this log-likelihood function is no longer additively separable, and we cannot directly estimate it in two stages as before. The first order condition that stems from maximizing the above likelihood is:

$$\begin{aligned} 0 &= \sum_{\mathrm{i}=1}^{\mathrm{N}} \frac{\sum_{k=1}^{K} \pi_{k|\boldsymbol{x}_{i1}} \sum_{t=1}^{T} \left[\frac{\partial f_t(d_{it}, \boldsymbol{x}_{it+1}|\boldsymbol{x}_{it}, \boldsymbol{\omega}^k; \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \prod_{t' \neq t} f_{t'}(d_{it'}, \boldsymbol{x}_{it'+1}|\boldsymbol{x}_{it'}, \boldsymbol{\omega}^k; \boldsymbol{\theta}) \right]}{\sum_{k=1}^{K} \pi_{k|\boldsymbol{x}_{i1}} \prod_{t=1}^{T} f(d_{it}, \boldsymbol{x}_{it+1}|\boldsymbol{x}_{it}, \boldsymbol{\omega}^k; \boldsymbol{\theta})} \\ &= \sum_{\mathrm{i}=1}^{\mathrm{N}} \frac{\sum_{k=1}^{K} \pi_{k|\boldsymbol{x}_{i1}} \sum_{t=1}^{T} \left[\frac{\partial \ln f_t(d_{it}, \boldsymbol{x}_{it+1}|\boldsymbol{x}_{it}, \boldsymbol{\omega}^k; \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \prod_{t'=1}^{T} f_{t'}(d_{it'}, \boldsymbol{x}_{it'+1}|\boldsymbol{x}_{it'}, \boldsymbol{\omega}^k; \boldsymbol{\theta}) \right]}{\sum_{k=1}^{K} \pi_{k|\boldsymbol{x}_{i1}} \prod_{t=1}^{T} f(d_{it}, \boldsymbol{x}_{it+1}|\boldsymbol{x}_{it}, \boldsymbol{\omega}^k; \boldsymbol{\theta})}, \end{aligned} \tag{37}$$

where the second expression is obtained by replacing $\partial f_t(\cdot)/\partial \boldsymbol{\theta} = f_t(\cdot)\partial \ln f_t(\cdot)/\partial \boldsymbol{\theta}$. This first order condition is easily reinterpreted using the Bayes' Rule because the term:

$$\frac{\pi_{k|\boldsymbol{x}_{i1}} \prod_{t'=1}^{T} f_{t'}(d_{it'}, \boldsymbol{x}_{it'+1}|\boldsymbol{x}_{it'}, \boldsymbol{\omega}^k; \boldsymbol{\theta})}{\sum_{k=1}^{K} \pi_{k|\boldsymbol{x}_{i1}} \prod_{t=1}^{T} f(d_{it}, \boldsymbol{x}_{it+1}|\boldsymbol{x}_{it}, \boldsymbol{\omega}^k; \boldsymbol{\theta})} \equiv \varpi(k|\boldsymbol{d}_i, X_i; \boldsymbol{\theta}, \boldsymbol{\pi}), \tag{38}$$

where $\boldsymbol{\pi}$ is the vector including all the type probabilities $\pi_{k|\boldsymbol{x}_{i1}}$ for all k and i, and $\boldsymbol{d}_i \equiv (d_{i1}, ..., d_{iT})'$ and $X_i \equiv (\boldsymbol{x}_{i1}, ..., \boldsymbol{x}_{iT})'$, represent the probability of being of type k conditional on the observed choices and state variables, and, therefore, the FOC implies:

$$0 = \sum_{\mathrm{i}=1}^{\mathrm{N}} \sum_{k=1}^{K} \sum_{t=1}^{T} \varpi(k|\boldsymbol{d}_i, X_i; \boldsymbol{\theta}, \boldsymbol{\pi}) \frac{\partial \ln f_t(d_{it}, \boldsymbol{x}_{it+1}|\boldsymbol{x}_{it}, \boldsymbol{\omega}^k; \boldsymbol{\theta})}{\partial \boldsymbol{\theta}}. \tag{39}$$

Dempster, Laird, and Rubin (1977) note that the following maximization problem delivers exactly the same first order conditions:

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \sum_{\mathrm{i}=1}^{\mathrm{N}} \sum_{t=1}^{T} \sum_{k=1}^{K} \varpi(k|\boldsymbol{d}_i, X_i; \hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\pi}}) \ln f_t(d_{it}, \boldsymbol{x}_{it+1}|\boldsymbol{x}_{it}, \boldsymbol{\omega}^k; \boldsymbol{\theta}), \tag{40}$$

for $\hat{\boldsymbol{\pi}}$ that satisfies:

$$\hat{\pi}_{k|\boldsymbol{x}_1} = \frac{\sum_{\mathrm{i}=1}^{\mathrm{N}} \varpi(k|\boldsymbol{d}_i, X_i; \hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\pi}}) \, \mathbb{1}\{\boldsymbol{x}_{i1} = \boldsymbol{x}_1\}}{\sum_{\mathrm{i}=1}^{\mathrm{N}} \mathbb{1}\{\boldsymbol{x}_{i1} = \boldsymbol{x}_1\}}. \tag{41}$$

Given this premise, they propose the EM algorithm, that is an iterative procedure that goes as follows. For any given guess $\boldsymbol{\theta}^{(m)}$ and $\boldsymbol{\pi}^{(m)}$ the expectation (E) step updates $\boldsymbol{\pi}^{(m+1)}$ as follows. First update the conditional probabilities of being of each step as $\varpi(k|\boldsymbol{d}_i, X_i; \boldsymbol{\theta}^{(m)}, \boldsymbol{\pi}^{(m)})$ evaluating (38) at the guessed parameters $\boldsymbol{\theta}^{(m)}$ and $\boldsymbol{\pi}^{(m)}$. Then update $\boldsymbol{\pi}^{(m+1)}$ averaging the new prediction across individuals for each initial state value as in (41). Using these two, the maximization (M) step updates $\boldsymbol{\theta}^{(m+1)}$ using (40). Iterating until convergence leads to consistent estimates of $\boldsymbol{\theta}$ and $\boldsymbol{\pi}$. Note that the M-step of the algorithm reintroduces additive separability (we are back to sum of logs), which, as shown in Arcidiacono and Jones (2003), implies that the updating of transition functions and utility parameters can be done separately.

Arcidiacono and Miller's approach. The EM algorithm described so far makes no use of CCP estimation. In fact, the application we just described implicitly predicates on the estimation of the M-step using full solution methods. Using CCP estimation in this context has a non-trivial difficulty: the CCPs and transition functions should be estimated conditional on the unobserved type, which is difficult given that, by construction, types are unobserved. Arcidiacono and Miller (2011) propose an implementation of the EM algorithm that allows the estimation of dynamic discrete choice models with unobserved heterogeneity using CCP estimation methods. Arcidiacono and Miller's approach is also well suited for time-varying persistent unobserved heterogeneity, but we are going to focus on permanent unobserved heterogeneity.

Using again the Bayes' Rule (and the law of iterated expectations), the CCPs can be expressed as:

$$\begin{aligned}\Pr(j|\boldsymbol{x}_{it},k) = \frac{\Pr(j,k|\boldsymbol{x}_{it})}{\Pr(k|\boldsymbol{x}_{it})} &= \frac{\mathbb{E}[\mathbb{1}\{d_{it}=j\}\,\mathbb{1}\{k_{it}=k\}|\boldsymbol{x}_{it}]}{\mathbb{E}[\mathbb{1}\{k_{it}=k\}|\boldsymbol{x}_{it}]} \\ &= \frac{\mathbb{E}\left[\mathbb{1}\{d_{it}=j\}\,\mathbb{E}[\mathbb{1}\{k_{it}=k\}|\boldsymbol{d}_i,X_i]|\boldsymbol{x}_{it}\right]}{\mathbb{E}\left[\mathbb{E}[\mathbb{1}\{k_{it}=k\}|\boldsymbol{d}_i,X_i]|\boldsymbol{x}_{it}\right]} \\ &= \frac{\mathbb{E}[\mathbb{1}\{d_{it}=j\}\varpi(k|\boldsymbol{d}_i,X_i;\boldsymbol{\theta},\boldsymbol{\pi})|\boldsymbol{x}_{it}]}{\mathbb{E}[\varpi(k|\boldsymbol{d}_i,X_i;\boldsymbol{\theta},\boldsymbol{\pi})|\boldsymbol{x}_{it}]}. \qquad (42)\end{aligned}$$

The Arcidiacono and Miller (2011) approach expands the E step of the EM algorithm to also update the CCPs computing the sample analog of the above expression using the updated $\varpi(k|\boldsymbol{d}_i,X_i;\boldsymbol{\theta}^{(m)},\boldsymbol{\pi}^{(m)})$.

B. Competitive equilibrium models and aggregate shocks

The extension of CCP estimation to competitive equilibrium models under this setting is straightforward in a world in which there are no aggregate shocks. As individuals are price-takers, aggregate equilibrium conditions in the baseline economy are just treated as nuisance (and non-fundamental) parameters of the model, and its estimation is often enriched with aggregate data or individual pay-off data, like wages (e.g. see Traiberman, 2018, for a recent example).

However, this becomes more complicated in the presence of aggregate shocks. To the best of my knowledge, only Altuğ and Miller (1998) and Llull (2020) deal with CCP estimation in the presence of aggregate shocks. The key difficulty for implementing this approach is that one needs to integrate over future counterfactual paths for the aggregate conditions, for which CCPs cannot be (directly) recovered from the data.

To be more explicit, expand the value functions above to account for an aggregate random variable r_t:

$$v_{jt}(\boldsymbol{x}_t,r_t) = u_{jt}(\boldsymbol{x}_t,r_t) + \beta\int\int V_{t+1}(\boldsymbol{x}_{t+1},r_{t+1})dF_x(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t,j)dF_r(r_{t+1}|r_t,\Omega_t). \qquad (43)$$

where Ω_t is the relevant information at time t that is useful to predict r_{t+1}. The CCP representation would imply replacing $V_{t+1}(\boldsymbol{x}_{t+1}, r_{t+1}) = v_{kt+1}(\boldsymbol{x}_{t+1}, r_{t+1}) + \psi_k(p_{t+1}(\boldsymbol{x}_{t+1}, r_{t+1}))$ for any possible value of $(\boldsymbol{x}_{t+1}, r_{t+1})$, factual or counterfactual. For example, consider a labor market in which the wage rate can be high or low. In the data, we observe a given sequence of wages for a given time period (say, e.g. H,H,L). However, we would need to compute the CCPs for any possible sequence of wages (i.e., H,H,H; L,H,H; H,L,H; and so on).

The solutions to this problem used in Altuğ and Miller (1998) and Llull (2020) share some similarities and also have important differences. In Altuğ and Miller (1998), the discrete choice problem is associated to a continuous choice on consumption. Armed with consumption data, they can use, say, rich households observed in bad times as good counterfactuals for poor households had they lived in good times. In a more standard dynamic discrete choice environment, Llull (2020) exploits the stationarity of his model besides the aggregate conditions. This allows him to use calendar time as a sufficient statistic for baseline aggregate conditions. Combined with wage data, this allows him to recover estimates of skill prices r_t in each of the years in the sample. Having estimated them, he uses, say, good years as counterfactuals for bad years had they been good. In the next section, we review Llull (2020) in detail as the application for this chapter.

V. Application: Llull (2020)

See the paper.