

# BUSN 37904 - Advanced Quantitative Marketing

## Bayesian Learning Assignment

Arjun Gopinath

We use the approach outlined in Erdem and Keane (1996) and Crawford and Shum (2005) to analyze and solve a learning model where consumers choose among  $J$  experience goods and learn about their match quality associated with each product. The code is provided in Python and requires the activation of a custom Conda environment.

### 1 Theory

#### The Model

A consumer (typically indexed by  $i$  but we ignore this notation) chooses among  $J$  experience goods,  $j = 0$  denotes the no-purchase option. The realized quality of product  $j$  is given by the sum of the intrinsic match value  $\vartheta_j$  and an idiosyncratic noise term,  $\nu_{jt}$ , which are i.i.d. draws across  $j$  and  $t$ .

$$\xi_{jt} = \vartheta_j + \nu_{jt}, \quad \nu_{jt} \sim \mathcal{N}(0, \sigma_\nu^2)$$

$\vartheta_j$  is the match value based on the product attributes and the consumer's idiosyncratic preferences over these attributes. Each consumer's match quality for product  $j$  represents the match value based on the product attributes and the consumer's idiosyncratic preferences over these attributes.

$$\vartheta_j \sim \mathcal{N}(\bar{\vartheta}_j, \tau_j^2)$$

However, consumers do not know their match quality. Instead, they have normal priors on the match values for each product, where at time  $t$ , the consumer believes that  $\vartheta_j$  is distributed as  $\pi_{jt}$  where

$$\pi_{jt} \equiv \mathcal{N}(\mu_{jt}, \sigma_{jt}^2)$$

We assume that the priors are independent across the product space and across consumers. Furthermore, under rational expectations, the initial prior is set such that  $\pi_{j0} \equiv \mathcal{N}(\bar{\vartheta}_j, \tau_j^2)$  wherein consumers begin by assuming the mean of their matching value is the average matching value.

After a purchase (which coincides with consumption) of product  $j$  at time  $t$ , the consumer observes the signal  $\xi_{jt}$  and updates their prior in a rational, Bayesian fashion. Given that the noisy shocks to the realized quality of the product are zero-mean, the expected value of the signal coincides with the expected

value of the match value associated with attribute  $j$  under the information set  $\mathbf{x}_{t-1}$ .

$$\mathbb{E}[\xi_{jt} \mid \mathbf{x}_{t-1}] = \mathbb{E}[\vartheta_j \mid \mathbf{x}_{t-1}] = \mu_{j,t-1}$$

Let  $C_{jt} = 1$  if the consumer chooses product  $j$  and 0 otherwise. Using standard Bayesian updating results given normality assumptions, the mean of the prior is updated from  $t - 1$  to  $t$  as shown below:

$$\begin{aligned} \mu_{jt} &:= \mu_{j,t-1} + C_{jt}\lambda_{j,t}(\xi_{jt} - \mu_{j,t-1}) \\ \text{with Kalman gain } \lambda_{jt} &= \frac{\sigma_{j,t-1}^2}{\sigma_{j,t-1}^2 + \sigma_\nu^2} \end{aligned} \tag{1}$$

The variance of the prior is updated deterministically as it does not depend on the random draw of the signal  $\xi_{j,t}$ .

$$\sigma_{j,t}^2 = \left( \frac{1}{\sigma_{j,t-1}^2} + \frac{C_{j,t}}{\sigma_\nu^2} \right)^{-1} \tag{2}$$

Since there is no correlated learning, i.e. the consumption of product  $j$  gives no new information about products  $k \neq j$ , the period- $t$  priors for the remaining  $J - 1$  products remain unchanged when product  $j$  is consumed in period  $t$ .

The realized utility (net of the latent utility draw  $\epsilon_{jt}$ ) conditional on the purchase of product  $j$  with price at time  $t$  denoted by  $p_{jt}$  is given by

$$u_{jt} = \gamma - \exp(-\rho\xi_{jt}) - \alpha p_{jt}$$

where  $\rho > 0$  captures the consumer's risk aversion.  $\gamma$  imposes restrictions on the market shares. Assume that the latent utility draws are Type I Extreme Value, i.i.d., and centered at 0. The price vectors  $\mathbf{p}_t$  are i.i.d. draws from a discrete distribution with a support of  $\{\mathbf{p}_1, \dots, \mathbf{p}_K\}$  with probability mass function given by  $\omega_k = \Pr\{\mathbf{p}_t = \mathbf{p}_k\}$ .

The state vector is given by  $\mathbf{x}_t = (\mathbf{p}_t, \boldsymbol{\pi}_t) = (p_{1t}, \dots, p_{Jt}, \mu_{1t}, \sigma_{1t}^2, \dots, \mu_{Jt}, \sigma_{Jt}^2)$ . The expected utility conditional on the purchase of  $j$  given state  $\mathbf{x}_t$  (which includes the consumer's prior belief about the match value of product  $j$ ) is

$$\begin{aligned} u_j(\mathbf{x}_t) &= \mathbb{E}[u_{jt} \mid \mathbf{x}_t] = \mathbb{E}[\gamma - \exp(-\rho\xi_{jt}) - \alpha p_{jt} \mid \mathbf{x}_t] \\ &= \gamma - \exp\left\{-\rho\mu_{jt} + \frac{\rho^2}{2}(\sigma_{jt}^2 + \sigma_\nu^2)\right\} - \alpha p_{kj} \end{aligned} \tag{3}$$

This uses the fact that from the consumer's perspective,  $\xi_{jt} \sim \mathcal{N}(\mu_{jt}, \sigma_{jt}^2 + \sigma_\nu^2)$ . We normalize the utility from the outside option such that  $u_0(\mathbf{x}_t) = 0$ .

## Characterizing the Bellman Equation

Consumers are forward-looking and discount the future using a discount factor  $\beta > 0$ . Optimal choices are characterized by the choice-specific value functions  $v_j(\mathbf{x})$ , such that consumers choose product  $j$  at time  $t$  ( $C_{jt} = 1$ ) if and only if

$$v_j(\mathbf{x}_t) + \epsilon_{jt} \geq v_k(\mathbf{x}_t) + \epsilon_{kt} \quad \forall k \neq j$$

Due to the Bayesian learning component, the consumer's choice problem is dynamic with two key considerations. The consumer's expected consumption utility in the current period and their learning process drive their intertemporal consumption choices. The Bellman equation facing a consumer at time  $t$  with state variable  $\mathbf{x}_t$  is given by

$$v(\mathbf{x}_t, \boldsymbol{\varepsilon}_t) = \max_j \left\{ \mathbb{E}_{\boldsymbol{\pi}_t} [\gamma - \exp(\rho \xi_{jt}) - \alpha p_{jt} + \varepsilon_{jt}] + \beta \mathbb{E}_{\mathbf{x}_{t+1}, \boldsymbol{\varepsilon}_{t+1}} [v(\mathbf{x}_{t+1}, \boldsymbol{\varepsilon}_{t+1})] \right\}$$

We can take expectations with respect to both  $\mathbf{p}_t$  and  $\boldsymbol{\varepsilon}_t$  in the above Bellman equation as they are both i.i.d. across  $t$ , and arrive at choice-specific value functions  $v_j$  in the expression on the RHS and the expected value function given beliefs  $\boldsymbol{\pi}_t$  on the LHS.

$$\bar{v}(\boldsymbol{\pi}_t) = \sum_{k=1}^K \omega_k \int \max_j \left\{ \mathbb{E}_{\boldsymbol{\pi}_t} [\gamma - \exp(\rho \xi_{jt}) - \alpha p_{kj} + \varepsilon_{jt}] + \beta \mathbb{E}_{\mathbf{x}_{t+1}, \boldsymbol{\varepsilon}_{t+1}} [v(\mathbf{x}_{t+1}, \boldsymbol{\varepsilon}_{t+1})] \right\} g(\boldsymbol{\varepsilon}) d\boldsymbol{\varepsilon} \quad (4)$$

$$= \sum_{k=1}^K \omega_k \int \max_j \{v_j(\mathbf{p}_k, \boldsymbol{\pi}_t) + \varepsilon_{jt}\} g(\boldsymbol{\varepsilon}) d\boldsymbol{\varepsilon} \quad (5)$$

We assume that the price vector is i.i.d. across time periods as it reduces the state space of our model. If we do not assume prices are i.i.d., the state space gets very large and the problem becomes quite computationally difficult. For instance, we will need to include lagged prices as a state variable as the probabilities  $\{\omega_{k,t}\}$  would now be a function of  $\mathbf{p}_{t-1}$ . This assumption is economically sound as we anticipate the consumer to fully account for prices in the utility function.

Given that the idiosyncratic shocks are i.i.d. Gumbel distributed, we can simplify the expression for the expected value function in terms of the choice-specific value functions.

$$\begin{aligned} \bar{v}(\boldsymbol{\pi}_t) &= \gamma + \mu + \sum_{k=1}^K \omega_k \log \left( \sum_{j=0}^J \exp \{v_j(\mathbf{p}_k, \boldsymbol{\pi}_t)\} \right) \\ &= \sum_{k=1}^K \omega_k \log \left( \sum_{j=0}^J \exp \{v_j(\mathbf{p}_k, \boldsymbol{\pi}_t)\} \right) \end{aligned} \quad (6)$$

We set the location parameter of the Gumbel distribution to the negative of the Euler-Mascheroni constant so that the latent utility draws are centered around zero. Note that the choice-specific value function is

now given by

$$v_j(\mathbf{p}_k, \boldsymbol{\pi}_t) = u_j(\mathbf{p}_k, \boldsymbol{\pi}_t) + \int_{\boldsymbol{\pi}} \bar{v}(\boldsymbol{\pi}_{t+1}) f(\boldsymbol{\pi}_{t+1} | \boldsymbol{\pi}_t, j) d\boldsymbol{\pi}_{t+1} \quad (7)$$

where  $f(\boldsymbol{\pi}_{t+1} | \boldsymbol{\pi}_t, j)$  denotes the state transition probability when the consumer chooses product  $j$  and updates their beliefs about the match value.

This implies that the Bellman equation that characterizes the expected value function  $\bar{v}(\cdot)$  is given by

$$\bar{v}(\boldsymbol{\pi}) = \sum_{k=1}^K \omega_k \log \left( \sum_{j=0}^J \exp \left\{ u_j(\mathbf{p}_k, \boldsymbol{\pi}) + \int_{\boldsymbol{\pi}'} \bar{v}(\boldsymbol{\pi}') f(\boldsymbol{\pi}' | \boldsymbol{\pi}, j) d\boldsymbol{\pi}' \right\} \right) \quad (8)$$

Note that the set of prior parameters  $\boldsymbol{\pi}$  has  $2J$  components as we assume there is no correlated learning and as such we can characterize the entire prior distribution using the first two moments associated with each product's perceived match quality. However, only one element of all  $2J$  elements evolves stochastically, which is the updated prior mean for the product chosen. The prior variance associated with that product decreases deterministically, and the prior moments of the remaining  $J - 1$  products remain unchanged.

We can characterize the conditional choice probabilities as a function of the state  $(\mathbf{p}_k, \boldsymbol{\pi}_t)$  as a result of the distributional assumption.

$$\Pr \{ \text{Consumer chooses } j | (\mathbf{p}_k, \boldsymbol{\pi}_t) \} = \frac{\exp \{ v_j(\mathbf{p}_k, \boldsymbol{\pi}_t) \}}{\sum_{\ell=0}^J \exp \{ v_{\ell}(\mathbf{p}_k, \boldsymbol{\pi}_t) \}} \quad (9)$$

Based on the approach outlined by Hartmann and Viard (2008), I compute the switching costs for a consumer from their previous choice  $j_-$  to their current choice  $j$  with current posterior beliefs  $\boldsymbol{\pi}^*$ , initial prior  $\boldsymbol{\pi}$  and price vector  $\mathbf{p}_k$ .

$$c(\boldsymbol{\pi}^*, \mathbf{p}_k) \equiv (v_j(\mathbf{p}_k, \boldsymbol{\pi}^*) - v_{j_-}(\mathbf{p}_k, \boldsymbol{\pi}^*)) - (v_j(\mathbf{p}_k, \boldsymbol{\pi}) - v_{j_-}(\mathbf{p}_k, \boldsymbol{\pi})) \quad (10)$$

## Numerical Approximations

We make use of Chebyshev interpolation and Gauss-Hermite quadrature to solve the Bellman equation in this assignment. First, given that the state variable  $\boldsymbol{\pi}$  consists of  $J$  means and  $J$  variances (no correlation terms as we are ruling out correlated learning), we have  $2J$  state variables. This implies that we can represent the expected value function through a system of  $2J$ -tensor product of Chebyshev polynomials with a given number of Chebyshev nodes and a specified degree of the polynomials used. If we use  $K$  nodes

$$\bar{v}(\boldsymbol{\pi}) \equiv \sum_{k=1}^K \theta_k T_k(\boldsymbol{\pi}) = \mathbf{T}(\boldsymbol{\pi}) \boldsymbol{\theta}$$

Furthermore, since  $f(\boldsymbol{\pi}_{t+1} \mid \boldsymbol{\pi}_t, j)$  corresponds to the p.d.f. of a multivariate normal distribution, we can use the Gauss-Hermite quadrature to evaluate the integral. This implies that the integral used to characterize the choice-specific value function can be rewritten as shown, assuming that we use  $Q$  quadrature nodes.

$$\begin{aligned}
\int_{\boldsymbol{\pi}} \bar{v}(\boldsymbol{\pi}_{t+1}) f(\boldsymbol{\pi}_{t+1} \mid \boldsymbol{\pi}_t, j) d\boldsymbol{\pi}_{t+1} &\approx \int_{\boldsymbol{\pi}} \sum_{k=1}^K \theta_k T_k(\boldsymbol{\pi}_{t+1}) f(\boldsymbol{\pi}_{t+1} \mid \boldsymbol{\pi}_t, j) d\boldsymbol{\pi}_{t+1} \\
&= \sum_{k=1}^K \theta_k \int_{\boldsymbol{\pi}} T_k(\boldsymbol{\pi}_{t+1}) f(\boldsymbol{\pi}_{t+1} \mid \boldsymbol{\pi}_t, j) d\boldsymbol{\pi}_{t+1} \\
&\equiv \sum_{k=1}^K \theta_k E_k(\boldsymbol{\pi}_t, j) \\
&\approx \sum_{k=1}^K \theta_k \sum_{q=1}^Q \psi_q(\boldsymbol{\pi}_t, j) T_k(\boldsymbol{\pi}_{t+1,q})
\end{aligned}$$

where we see that we now evaluate the expectation over the Chebyshev polynomials using Gauss-Hermite quadrature.

The numerical approximations outlined above will expedite the value function iteration. At iteration  $n$ , the Chebyshev approximation coefficients  $\boldsymbol{\theta}^{(n)}$  represent the optimal approximation of the expected value function  $\bar{v}$ . We can then evaluate each choice-specific value function as shown:

$$v_j^{(n)}(\mathbf{p}_{\mathbf{k}}, \boldsymbol{\pi}) \approx u_j(\mathbf{p}_{\mathbf{k}}, \boldsymbol{\pi}) + \beta \sum_{k=1}^K \theta_k^{(n)} \sum_{q=1}^Q \psi_q^{(j)} T_k(\boldsymbol{\pi}'_q) \quad (11)$$

These can be used to update the expected value function in the subsequent iteration

$$\bar{v}^{(n+1)}(\boldsymbol{\pi}) \leftarrow \sum_{k=1}^K \omega_k \log \left( \sum_j \exp v_j^{(n)}(\mathbf{p}_{\mathbf{k}}, \boldsymbol{\pi}) \right) \quad (12)$$

We obtain the updated Chebyshev coefficient vector  $\boldsymbol{\theta}^{(n+1)}$  based on the regression

$$\boldsymbol{\theta}^{(n+1)} = (\mathbf{T}(\boldsymbol{\pi})' \mathbf{T}(\boldsymbol{\pi}))^{-1} \mathbf{T}(\boldsymbol{\pi})' \bar{v}^{(n+1)}(\boldsymbol{\pi}) \equiv \tilde{\mathbf{T}}(\boldsymbol{\pi})' \bar{v}^{(n+1)}(\boldsymbol{\pi}) \quad (13)$$

The updated expected value function  $\bar{v}^{(n+1)}$  is represented as  $v^{(n+1)}(\boldsymbol{\pi}) \equiv \mathbf{T}(\boldsymbol{\pi}) \boldsymbol{\theta}^{(n+1)}$ . We can rewrite the Bellman equation (Equation 8) as it would be implemented using the Chebyshev polynomials and Gauss-Hermite quadrature:

$$\boldsymbol{\theta}^{(n+1)} \leftarrow \tilde{\mathbf{T}}(\boldsymbol{\pi})' \sum_{k=1}^K \omega_k \log \left( \sum_j \exp \left\{ u_j(\mathbf{p}_{\mathbf{k}}, \boldsymbol{\pi}) + \beta \sum_{q=1}^Q \psi_q^{(j)} \mathbf{T}(\boldsymbol{\pi}'_q) \boldsymbol{\theta}^{(n)} \right\} \right) \quad (14)$$

## 2 Computational Implementation

I created a `Python` class called `BayesianLearning` which included methods to compute choice specific value functions as in Equation 11, generate expected value functions as fixed point solutions to the Bellman Equation 14, and simulate the trajectory of a consumer with a specific prior belief set who Bayesian-learns about their match quality over a specified time horizon. This uses methods from the accompanying classes `ChebyshevApproximator` and `GaussHermiteQuadrature` to implement these via the use of numerical approximations.

Algorithm 1 shows the approach I used to compute the choice-specific value functions associated with  $J + 1$  products (including the outside option) for  $K$  different price vectors as a function of the consumer's current beliefs about their match qualities and the Chebyshev coefficient vector representing the current guess for the expected value function. In my code, I implement this in the class method `get_choice_specific_vf()`.

Algorithm 2 implements the Value Function Iteration procedure used to obtain a fixed point solution to Equation 8. This algorithm goes on until either the norm of the difference in Chebyshev coefficients from subsequent iterations is small enough or the iteration count has exceeded a specified maximum. This is implemented in class method `get_expected_vf()`.

Algorithm 3 simulates the learning pattern of a Bayesian consumer who chooses among a set of products to consume in each period over a time horizon. The method `bayesian_learning()` inside the eponymous class returns a list of the products chosen, the conditional choice probabilities and switching costs in each time period.

## References

- Crawford, Gregory S. and Matthew Shum (2005). “Uncertainty and Learning in Pharmaceutical Demand”. *Econometrica* 73.4, pp. 1137–1173.
- Erdem, Tülin and Michael P. Keane (1996). “Decision-Making under Uncertainty: Capturing Dynamic Brand Choice Processes in Turbulent Consumer Goods Markets”. *Marketing Science* 15.1, pp. 1–20.
- Hartmann, Wesley and V. Viard (June 2008). “Do frequency reward programs create switching costs? A dynamic structural analysis of demand in a reward program”. *Quantitative Marketing and Economics (QME)* 6.2, pp. 109–137.

## A Pseudocode for Algorithms

This appendix lists the pseudocode for each algorithm used in the computational implementation of my Bayesian learning model.

---

### Algorithm 1 Generating the Choice-Specific Value Functions

---

Takes a vector of prior mean parameters  $\boldsymbol{\mu}$  (dimension  $J$ ), prior variance matrix  $\boldsymbol{\Sigma}$  (dimension  $J \times J$ ), and Chebyshev coefficient vector  $\boldsymbol{\theta}$  (dimension  $(N+1)^{2J}$ ) representing the expected value function in terms of the posterior mean and variance of each product.

```

1: function GET_CHOICE_SPECIFIC_VF( $\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\theta}$ )
2:   for  $k = 1, \dots, K$  do ▷ Evaluate for each possible price vector.
3:     for  $j = 0, \dots, J$  do ▷ Evaluate CSVF for product  $j$ .
4:        $\boldsymbol{\xi}, \boldsymbol{\psi} \leftarrow Q$  Gauss-Hermite quadrature nodes and weights from  $\mathcal{N}(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_{jj} + \sigma_v^2)$ .
5:        $\tilde{\boldsymbol{\mu}}^{(j)} \leftarrow Q$  posterior means corresponding to signal vector  $\boldsymbol{\xi}$  for product  $j$  via Equation 1.
6:        $\tilde{\boldsymbol{\Sigma}}_{jj} \leftarrow$  Posterior variance for product  $j$  via Equation 2
7:        $(\boldsymbol{\mu}_{\text{post}}, \boldsymbol{\Sigma}_{\text{post}}) \leftarrow Q$  sets of posterior moments with those for products  $j' \neq j$  unchanged.
8:        $\mathbf{T}_j \leftarrow$  Matrix  $(Q \times (N+1)^{2J})$  of Chebyshev polynomial coefficients representing posterior
          means and variances after product  $j$  is chosen.
9:        $\mathbf{E}_j \leftarrow$  Vector  $((N+1)^{2J})$  containing the expected value of the Chebyshev polynomial
          coefficients using the GHQ weights, i.e.  $\pi^{-0.5} \boldsymbol{\psi}' \mathbf{T}_j$ 
10:       $u_{jk} \leftarrow$  Expected utility associated with product  $j$  and price vector  $k$ , equal to 0 for  $j = 0$ .
11:       $v_{jk} \leftarrow$  CSVF associated with product  $j$  and price vector  $k$ , i.e.  $u_{jk} + \beta \mathbf{E}_j \cdot \boldsymbol{\theta}$ .
12: return Choice-specific value function  $\mathbf{v}$ , matrix (dimension  $J+1 \times K$ ) for  $J+1$  products corresponding
    to  $K$  possible price vectors.

```

---



---

**Algorithm 2** Computing the Expected Value Function

---

Takes a vector of prior mean parameters  $\boldsymbol{\mu}$  (dimension  $J$ ), prior variance matrix  $\boldsymbol{\Sigma}$  (dimension  $J \times J$ ), and an initial guess of a Chebyshev coefficient vector  $\boldsymbol{\theta}_{\text{init}}$  (dimension  $(N+1)^{2J}$ ) for the expected value function.

```
1: function GET_EXPECTED_VF( $\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\theta}_{\text{init}}$ )
2:    $\boldsymbol{\Pi} \equiv \{\boldsymbol{\mu}_{(m)}, \boldsymbol{\Sigma}_{(m)}\} \leftarrow M^{2J}$  Chebyshev nodes generated from the prior beliefs.
3:    $\boldsymbol{\theta}_0 \leftarrow \boldsymbol{\theta}_{\text{init}}$ 
4:   while  $ite < ite\_max$  and  $error > tol$  do ▷ Exit conditions for the VFI.
5:     for  $m = 1, \dots, M^{2J}$  do ▷ Apply to each set of Chebyshev node prior beliefs.
6:        $\mathbf{v}_{(m)} \leftarrow \text{GET\_CHOICE\_SPECIFIC\_VF}(\boldsymbol{\mu}_{(m)}, \boldsymbol{\Sigma}_{(m)}, \boldsymbol{\theta}_{ite})$  CSVF (dimension  $(J+1) \times K$ ) for
       node  $m$  using Algorithm 1.
7:        $\bar{\mathbf{v}}_{ite+1} \leftarrow \{\mathbf{v}_{(m)}\}$  Obtain EVF (dimension  $M^{2J}$ ) using Equation 12.
8:        $\boldsymbol{\theta}_{ite+1} \leftarrow$  Chebyshev coefficient vector (dimension  $(N+1)^{2J}$ ) from Equation 13.
9:        $error = \|\boldsymbol{\theta}_{ite+1} - \boldsymbol{\theta}_{ite}\|$ 
10:     $ite = ite + 1$ 
11:   $\boldsymbol{\theta}^* \leftarrow \boldsymbol{\theta}_{ite+1}$ 
  return Vector  $\boldsymbol{\theta}^*$  of dimension  $(N+1)^{2J}$ , the fixed-point solution to the Bellman Equation 14
  given the inputs.
```

---

---

**Algorithm 3** Simulating the Bayesian Learning Experience of a Consumer

---

Takes the consumer's actual match quality  $\vartheta$  and a simulated trajectory of prices  $\mathbf{x}$  (of length  $T$  with integer values from 1 to  $K$ , where  $T$  is the time horizon over which the consumer's behavior is simulated and  $K$  is the number of possible price vector realizations) as inputs. The class object is initialized with the prior mean  $\boldsymbol{\mu}_{\text{init}}$  and variance  $\boldsymbol{\Sigma}_{\text{init}}$  upon creation.

```
1: function BAYESIAN_LEARNING( $\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\theta}_{\text{init}}$ )
2:    $\boldsymbol{\theta}_0 \leftarrow \mathbf{0}_{(N+1)^{2J}}$  Initial guess for the expected value function.
3:    $\boldsymbol{\mu}_0 \leftarrow \boldsymbol{\mu}_{\text{init}}$  and  $\boldsymbol{\Sigma}_0 \leftarrow \boldsymbol{\Sigma}_{\text{init}}$ .
4:   for  $t = 1, \dots, T$  do ▷ Span the time horizon.
5:      $\text{price} \leftarrow \text{price\_matrix}[\mathbf{x}[t], :]$  Obtain prices for products at time  $t$  based on simulated draws.
6:      $\boldsymbol{\theta}_t \leftarrow \text{GET\_EXPECTED\_VF}(\boldsymbol{\mu}_{t-1}, \boldsymbol{\Sigma}_{t-1}, \boldsymbol{\theta}_{t-1})$  Compute the EVF based on the time- $t$  beliefs,
       using the previous time period's Chebyshev coefficients as an initial guess.
7:      $\mathbf{v} \leftarrow \text{GET\_CHOICE\_SPECIFIC\_VF}(\boldsymbol{\mu}_{t-1}, \boldsymbol{\Sigma}_{t-1}, \boldsymbol{\theta}_t)$  Evaluate CSVF associatd with  $\text{price}$ .
8:      $\text{CCP}[t] \leftarrow$  Compute the conditional choice probabilities for the consumer at time  $t$  using
       Equation 9.
9:      $\boldsymbol{\varepsilon} \leftarrow$  Simulated  $J$  i.i.d. draws from a T1EV distribution centered at 0.
10:     $j^* \leftarrow \arg \max_{j=0, \dots, J} (\mathbf{v} + \boldsymbol{\varepsilon})_j$  Product choice corresponds to the maximum possible sum of the CSVF
       and the idiosyncratic preference shocks.
11:    if  $j^* \neq 0$  then A product is chosen and beliefs will be updated.
12:       $\xi \leftarrow$  Simulated signal realized by the consumer for product  $j$  from  $\mathcal{N}(\vartheta_j, \tau_j^2 + \sigma_\nu^2)$ .
13:       $(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t) \leftarrow$  Update beliefs for product  $j^*$  using Equations 1 and 2 based on signal  $\xi$ .
14:    else The outside option is chosen and beliefs remain unchanged.
15:       $(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t) \leftarrow (\boldsymbol{\mu}_{t-1}, \boldsymbol{\Sigma}_{t-1})$ 
16: return Product choice vector (length  $T$  with integer values from 0 to  $J$ ) and  $\text{CCP}$  (matrix of size
        $T \times (J + 1)$ ) containing conditional choice probabilities at each time period.
```

---