The infinite horizon value function:

$$v_j(X_i) = u_j(X_i) + \beta \int_{X_i'} V(X_i') dF_j(X_i'|X_i)$$

The infinite horizon value function:

$$v_j(X_i) = u_j(X_i) + \beta \int_{X_i'} V(X_i') dF_j(X_i'|X_i)$$

Expanding the continuation value:

$$v_j(X_i) = u_j(X_i) + \beta \int_{X_i'} E_{\epsilon'} \left(\max_k v_k(X_i') + \epsilon_{ik}' \right) dF_j(X_i'|X_i)$$

For T1EV ϵ's, the expectation of the maximum has a closed form:

$$v_j(X_i) = u_j(X_i) + \beta \int_{X_i'} \log\left(\sum_{k=1}^{J} \exp[v_k(X_i')]\right) dF_j(X_i'|X_i) + \beta c$$

For T1EV ϵ's, the expectation of the maximum has a closed form:

$$v_j(X_i) = u_j(X_i) + \beta \int_{X_i'} \log\left(\sum_{k=1}^{J} \exp[v_k(X_i')]\right) dF_j(X_i'|X_i) + \beta c$$

Key insight: The v's appear on both sides of the equation

This creates a system that requires solving for a fixed point

For T1EV ϵ's, the expectation of the maximum has a closed form:

$$v_j(X_i) = u_j(X_i) + \beta \int_{X_i'} \log \left(\sum_{k=1}^{J} \exp[v_k(X_i')] \right) dF_j(X_i'|X_i) + \beta c$$

Key insight: The v's appear on both sides of the equation

This creates a system that requires solving for a fixed point

(This works because it is a contraction mapping)

Let $\mathcal{X}$ denote the number of states X can take on

Stack the conditional value functions for each possible state and choice:

$$\begin{bmatrix} v_1(X_1) \\ v_1(X_2) \\ \vdots \\ v_1(X_{\mathcal{X}}) \\ \vdots \\ v_J(X_{\mathcal{X}}) \end{bmatrix} = \begin{bmatrix} u_1(X_1) + \beta \int_{X'} \log\left(\sum_{k=1}^{J} \exp[v_k(X')]\right) dF_1(X'|X_1) + \beta c \\ u_1(X_2) + \beta \int_{X'} \log\left(\sum_{k=1}^{J} \exp[v_k(X')]\right) dF_1(X'|X_2) + \beta c \\ \vdots \qquad\qquad \vdots \qquad\qquad \vdots \\ u_1(X_{\mathcal{X}}) + \beta \int_{X'} \log\left(\sum_{k=1}^{J} \exp[v_k(X')]\right) dF_1(X'|X_{\mathcal{X}}) + \beta c \\ \vdots \qquad\qquad \vdots \qquad\qquad \vdots \\ u_J(X_{\mathcal{X}}) + \beta \int_{X'} \log\left(\sum_{k=1}^{J} \exp[v_k(X')]\right) dF_J(X'|X_{\mathcal{X}}) + \beta c \end{bmatrix}$$

Let $\mathcal{X}$ denote the number of states X can take on

Stack the conditional value functions for each possible state and choice:

This system of equations can be written compactly as:

$$\mathbf{v} = \mathbf{u} + \beta \mathbf{T}(\mathbf{v})$$

where $\mathbf{T}(\cdot)$ represents the expectation operator

Solution algorithm:

- Plug in values for the parameters and take a guess at the v's

Solution algorithm:

- Plug in values for the parameters and take a guess at the v's
- Substitute in for the v's on the right hand side which gives us a new set of v's

Solution algorithm:

- Plug in values for the parameters and take a guess at the v's
- Substitute in for the v's on the right hand side which gives us a new set of v's
- Repeat until convergence

Solution algorithm:

- Plug in values for the parameters and take a guess at the v's
- Substitute in for the v's on the right hand side which gives us a new set of v's
- Repeat until convergence

Formally:

Iterate

$$\mathbf{v}^{(n+1)} = \mathbf{u} + \beta \mathbf{T}(\mathbf{v}^{(n)})$$

until

$$\|\mathbf{v}^{(n+1)} - \mathbf{v}^{(n)}\| < \delta$$

The complete estimation algorithm has a similar nested structure as finite horizon case:

Outer loop (parameter estimation):

- Guess parameter values $\boldsymbol{\theta}^{(0)}$ (utility coefficients, β, etc.)

The complete estimation algorithm has a similar nested structure as finite horizon case:

Outer loop (parameter estimation):

- Guess parameter values $\boldsymbol{\theta}^{(0)}$ (utility coefficients, β, etc.)

Inner loop (value function solution):

- For given $\boldsymbol{\theta}^{(m)}$, solve fixed point: $\mathbf{v}^{(n+1)} = \mathbf{u}(\boldsymbol{\theta}^{(m)}) + \beta \mathbf{T}(\mathbf{v}^{(n)})$
- Continue until $\|\mathbf{v}^{(n+1)} - \mathbf{v}^{(n)}\| < \delta$

The complete estimation algorithm has a similar nested structure as finite horizon case:

Outer loop (parameter estimation):

- Guess parameter values $\boldsymbol{\theta}^{(0)}$ (utility coefficients, β, etc.)

Inner loop (value function solution):

- For given $\boldsymbol{\theta}^{(m)}$, solve fixed point: $\mathbf{v}^{(n+1)} = \mathbf{u}(\boldsymbol{\theta}^{(m)}) + \beta \mathbf{T}(\mathbf{v}^{(n)})$
- Continue until $\|\mathbf{v}^{(n+1)} - \mathbf{v}^{(n)}\| < \delta$

Back to outer loop:

- Calculate choice probabilities using solved $\mathbf{v}(\boldsymbol{\theta}^{(m)})$
- Evaluate likelihood $\mathcal{L}(\boldsymbol{\theta}^{(m)})$
- Update parameters to $\boldsymbol{\theta}^{(m+1)}$ using nonlinear optimizer (e.g. L-BFGS)
- Repeat until likelihood convergence