USING RANDOMIZATION TO BREAK THE CURSE OF DIMENSIONALITY

John Rust, Econometrica, May 1997

Presented by

Dinakar Jayarajan

RUST, 1987 APPROACH

- Single dimensional state (mileage)
- Discretize mileage into 90 grid pts in 5000 mile intervals
- Compute value function at these grid pts
- Round data to these grid pts and compute likelihood

In reality,

- Most problems have multidimensional state spaces leading to the *Curse of dimensionality*
- Consider a 3 dimensional state space discretized into 100 grid points each => 100^{3} states
- Not computationally feasible

KEANE & WOLPIN, 1994

- Discretize the states using a feasible number
- Compute value function at these states
- Use interpolation/function approximation to compute the value function at non-grid states
- Provides Monte Carlo evidence that the approach works
- Possibly the de-facto standard to estimate DDCs in Marketing and Labor Economics

This approach will face curse of dimensionality in the interpolation/approximation method

RUST, 1997

Use randomization to break the curse of dimensionality

Works for a subclass of MDPs where

- All state variables are continuous and evolve stochastically
- Actions are discrete and finite

This subclass is the Dynamic Discrete Choice problems commonly seen in marketing and economics

As we will see soon, it does not face CoD from interpolation/approximation algorithm

Fairly simple and straight forward to implement

BELLMAN OPERATOR

Bellman Operator is a mapping Γ: B -> B given by

$$(2.6) \qquad \Gamma(W)(s) \equiv \max_{a \in A(s)} \left[u(s,a) + \beta \int W(s') p(ds'|s,a) \right].$$

Decision Rule

$$(2.8) \qquad \alpha(s) = \operatorname*{argmax}_{a \in A(s)} \left[u(s,a) + \beta \int V(s') p(ds'|s,a) \right],$$

Value function is the solution to the Bellman Equation

$$(2.9) \qquad V(s) = \max_{a \in A(s)} \left[u(s,a) + \beta \int V(s') p(ds'|s,a) \right].$$

RANDOM BELLMAN OPERATOR

The Random Bellman operator (RBO) is also a mapping given by

$$(3.1) \qquad \tilde{\Gamma}_N(V)(s) \equiv \max_{a \in A}\left[u(s,a) + \frac{\beta}{N}\sum_{k=1}^{N} V(\tilde{s}_k)p(\tilde{s}_k|s,a)\right],$$

where s(tilde) are N randomly chosen states

The value function will converge to the true value function as N -> ∞ at the rate of √ N

This operator will be a contraction mapping only for large N

This is because the transition function p(.|s) may not sum to 1 for each s

CONVERGENCE OF RBO

Modify the transition function so it is well behaved

$$(3.5) \qquad p_N(s_k|s,a) = \frac{p(s_k|s,a)}{\sum_{i=1}^{N} p(s_i|s,a)}$$

The resulting RBO will be a contraction mapping for all N

$$(3.4) \qquad \hat{\Gamma}_N(V)(s) = \max_{a \in A} \left[u(s,a) + \beta \sum_{k=1}^{N} V(s_k) p_N(s_k|s,a) \right],$$

This operator is *self-approximating, i.e.*, for any s, the second term is an approximation to the expectation computed using the N randomly chosen states and *u(s,a)* is easily obtained

FINITE HORIZON PROBLEMS

Solved with Backward Induction

Draw N random state points and keep them fixed for the T iterations

In the final period T, the value function is given by

$$(4.1) \quad \hat{V}_T(\tilde{s}_i) = \underset{a \in A}{\operatorname{argmax}}\, u(\tilde{s}_i, a) \qquad (i = 1, \ldots, N).$$

For previous periods T-1,T-2,…0, apply the RBO

$$(4.2) \quad \hat{V}_{T-t}(\tilde{s}_i) = \hat{\Gamma}_N^t(V_T)(\tilde{s}_i) \qquad (t = 0, \ldots, T;\ s_i = 1, \ldots, N).$$

COMPLEXITY

Upper bound on worst case complexity for finite horizon problems is given by

$$(4.3) \qquad \text{comp}^{\text{wor-ran}}(\varepsilon, d) = O\left(\frac{Td^4|A|^5K_u^4K_p^4}{(1-\beta)^8\varepsilon^4}\right).$$

Upper bound for infinite problems is

$$(4.4) \qquad \text{comp}^{\text{wor-ran}}(\varepsilon, d) = O\left(\frac{\log(1/(1-\beta)\varepsilon)d^4|A|^5K_u^4K_p^4}{|\log(\beta)|(1-\beta)^8\varepsilon^4}\right).$$

Above holds for small ε and large β

We can improve this further by using a Multigrid algorithm for infinite horizon problems

RANDOM MULTIGRID ALGORITHM

Have an outer loop, in addition to the inner (successive approximations) loop, where the number of grid pts are varied

Start with a small number of states N_0 in outer iteration k=0

Init V_0 using max per period utility across all states and actions

Perform a series of outer iteration k=1,2,…

- Draw N_k uniform samples, where $N_k = 4 \times N_{k-1}$
- Draws are independent of draws from previous iterations
- Compute T(k) successive approximations using RBO
- Starting value function V_k in k^{th} iteration is the value function V_{k-1} obtained in $(k-1)^{th}$ iteration

RANDOM MULTIGRID ALGORITHM (CONT)

Stopping rule for T(k)

$$(4.8) \qquad E\left\{\left\|\hat{\Gamma}_{N_k}^{t}(\hat{V}_{k-1}) - \hat{\Gamma}_{N_k}^{t-1}(\hat{V}_{k-1})\right\|\right\} \le \frac{K}{\sqrt{N_k}\,\beta(1-\beta)}.$$

Stopping rule for Outer iterations

$$(4.9) \qquad N_{k^*}\frac{K^2}{(1-\beta)^4\varepsilon^2}.$$

where $$(2.17) \qquad K \equiv \sup_{s\in S}\sup_{a\in A(s)} |u(s,a)|.$$

Upper bound on worst case complexity is given by

$$\mathrm{comp}^{\text{wor-ran}}(\varepsilon, d) = O\left(\frac{|A|^5 d^4 K_u^4 K_p^4}{|\log(\beta)|(1-\beta)^8\varepsilon^4}\right).$$

This is an order better than the vanilla RBO algorithm

CONCLUSION

Possible to determine how many grid points and iterations are required to achieve a certain ε error

In reality, $K = 1$, $\varepsilon = 0.1$, $\beta = 0.995$ by Eqn 4.9,

$$N_k = 1.6 \times 10^6 \text{ states}$$

In practice, computing P_N for large N (=10,000) is very time consuming

Only known application in marketing is Gordon, 2009, Marketing Science

Additional reference – Rust, 1996, Handbook of Computational Economics, Vol 1