Dynamic Programming

Introduction

- The imperialism of Dynamic Programming (Ljunqvist & Sargent)

Markov chain and Markov process

- Stochastic process: family of random variables indexed by time
- A stochastic process has the Markov property if its future evolution depends only on its current state.
- Special cases:

	Discrete States	Continuous States
Discrete Time	Discrete Markov Chain	Continuous Markov Chain
Continuous Time	Markov Jump Process	Markov Process

Stochastic matrices

- a $n \times n$ matrix is said to be **stochastic** if all the lines lines sum to 1
- let's a vector $\mu_t \in R^n$ with positive components denote a distribution of mass between n different states
- if $\sum_{i=1}^{n} \mu_{i,t} = 1$ then μ_t is a probability density

Simulation

- Consider: $\mu'_{i,t+1} = \mu'_t P$
- We have $\mu_{i,t+1} = \sum_{k=1}^{n} \mu_{k,t} P_{k,i}$
- And: $\sum_i \mu_{i,t+1} = \sum_i \mu_{i,t}$
- Postmultiplication by a stochastic matrix preserves the mass.
- Interpretation: P_{ij} is the fraction of the mass initially in state i which ends up in j

Example

$$\underbrace{\begin{pmatrix} ? & ? & ? \end{pmatrix}}_{\mu'_{t+1}} = \underbrace{\begin{pmatrix} 0.5 & 0.3 & 0.2 \end{pmatrix}}_{\mu'_t} \begin{pmatrix} 0.4 & 0.6 & 0.0 \\ 0.2 & 0.5 & 0.3 \\ 0 & 0 & 1.0 \end{pmatrix}$$

Representation as a graph

{% dot attack_plan.svg digraph G { rankdir=LR A B C A -> B [label=0.4] A -> C [label=0.6] C -> C [label=1.0] B -> A [label=0.2] B -> B [label=0.5] B -> C [label=0.5] } %}

Probabilistic interpretation

- Denote by $S = (s_1, ...s_n)$ a finite set with n elements ($|S| = n$).
- A **Markov Chain** with values in S and with transitions given by a stochastic matrix $P \in R^n \times R^n$ is a *stochastic process* $(X_t)_{t\geq 0}$ such that

$$P_{ij} = Prob(X_{t+1} = s_j | X_t = s_i)$$

- In words, line i describes the conditional distribution of X_{t+1} conditional on $X_t = s_i$.

What about longer horizons?

- It is easy to show that for any k, P^k is a stochastic matrix.
- P^k_{ij} denotes the probability of ending in j, after k periods, starting from i
- Given an initial distribution $\mu_0 \in R^n$
  - Which states will visited with positive probability between t=0 and t=k?
  - What happens in the very long run?
- We need to study a little bit the properties of Markov Chains

Connectivity

- Two states s_i and s_j are connected if $P_{ij} > 0$
- We call incidence matrix: $\mathcal{I}(P) = (\delta_{P_{ij}>0})_{ij}$
- Two states i and j communicate with each other if there are k and l such that: $(P^k)_{i,j} > 0$ and $(P^l)_{j,i} > 0$
  - it is an equivalence relation
  - we can define equivalence classes
- A stochastic matrix P is irreducible if all states communicate
  - there is a unique communication class

Connectivity and irreducibility (example from QE)

Connectivity and irreducibility (example from QE)

Aperiodicity

- Are there cycles? Starting from a state i, how long does it take to return to i?
- The **period** of a state is defined as

$$gcd(k \geq 1 | (P^k)_{i,i} > 0)$$

- If a state has a period d>1 the chain returns to the state only at dates multiple of d.

Aperiodicity

example

Stationary distribution

- μ is a **stationary** distribution if $\mu' = \mu' P$
- Theorem: there always exists such a distribution
  - proof: Brouwer theorem
  - $f : \mu \to (\mu' P)'$
- Theorem:
  1. if P is irreducible the fixed point $\mu^\star$ is unique
  2. if P is irreducible and aperiodic $|\mu_0' P^k - \mu^\star| \underset{k \to +\infty}{\longrightarrow} 0$ for any initial distribution μ_0
- We then say the Markov chain is **ergodic**.

Stationary distribution (proof)

- **Brower's theorem**: Let $\mathcal{C}$ be a compact convex subset of R^n and f a continuous mapping $\mathcal{C} \to \mathcal{C}$. Then there exists a fixed point $x_0 \in \mathcal{C}$ such that $f(x_0) = x_0$

Stationary distribution?

How do we compute the stationary distribution?

- Simulation
- Linear algebra
- Decomposition

Simulating a Markov Chain

- Very simple idea: start with μ_0 and compute the iterates recursively
  - $\mu'_{n+1} = \mu'_n P$
  - convergence is linear

Using Linear Algebra

- Find the solution of $\mu'(P - I) = 0$?
  - not well defined, 0 is a solution
  - we need to incorporate the constraint $sum(\mu_i) = 1$
- Define $M_{ij} = (P - I)_{ij}$ if $j > 1$, 1 if $j = 1$
- Look for a solution μ of $\mu' M = (\delta_{i=1})$ with a linear algebra solver
  - if the solution completes, there is a unique solution

Further comments

- Knowledge about the structure of the Markov Chain can help speedup the calculations
- There are methods for potentially very-large linear system
  - Newton-Krylov based methods, GMRES
- Basic algorithms are easy to implement by hand
- QuantEcon toolbox has very good methods to study markov chains

Dynamic Programing: notations

General Formulation

Markov Decision Problem

- states: $s \in S$
- actions: $x \in X(s)$
- transitions: $\pi(s'|s, x)$
  - probability of going to s' in state s, given action x

Objective (finite horizon)

- policy: $x() : s \to x \in S(x)$
  - deterministic policy
  - given $x()$, the evolution of s' is a Markov process.
- reward: $r(s, x)$
  - felicity, intratemporal utility
- expected lifetime reward: starting from s
  - $R(s; x()) = E_0 \sum_t^T \delta^t [r_t]$
  - $\delta \in [0, 1[$: discount factor, $T \in \{N, \infty\}$: horizon

Classes of Dynamic Optimization

The formulation so far is very general. It encompasses several variants of the problem:

- finite horizon vs infinite horizon
- discrete-space problem vs continuous-state space problem

There are also variants not included: - non time-separable problems - non time homogenous problem - learning problems (bayesian updating, reinforcement learning)

Finite horizon vs infinite horizon

- Recall objective: $V(s;x()) = \max E_0 \sum_{t=0}^{T} \delta^t \left[r(s_t, x_t)\right]$
- If $T < \infty$, the decision in the last periods, will be different from before
  - one must find a decision rule $\pi_t()$ per period
  - or add t to the state space: $\tilde{S} = S \times [0, T]$
- If $T = \infty$, the continuation value of being in state s_t is independent from t

$$V(s;x()) = E_0 \max \sum_{t=0}^{T_0} \delta^t \left[r(s_t, x_t)\right] + \delta^{T_0} E_0 \sum_{t=T_0}^{\infty} \delta^t \left[r(s_t, x_t)\right]$$

$$= E_0 \left[\max \sum_{t=0}^{T_0} \delta^t \left[r(s_t, x_t)\right] + \delta^{T_0} V(s_{T_0}; x()) \right]$$

Continuous vs discrete

- State space S:
  - continuous: $\subset R^n$
  - discrete: $S = (s_1, ...s_n)$ (today)
- General approach is the same but implementation is different:
- Continous problem:
  - $x(s)$, $V(s;\pi)$ require an infinite number of coefficients
  - one must discretize the initial problem and solve an approximate version
- Discrete problem:
  - there is a finite number of policies, the can be represented exactly
  - unless $|S|$ is very large (cf go game)

Non time separable

- For instance Epstein-Zin preferences:

$$\max V(;c())$$

where $V_t = (1-\delta)\frac{c_t^{1-\sigma}}{1-\sigma} + \delta \left[E_t V_{t+1}^{\alpha}\right]^{\frac{1}{\alpha}}$

- Why would you do that?
  - to disentangle risk aversion and elasticity of intertemporal substitution
  - robust control
- You can still use ideas from Dynamic Programming.

Non homogenous preference

- Look at the $\alpha - \beta$ model.

$$V_t = \max \sum_t^{\infty} \beta_t U(c_t)$$

where $\delta_0 = 1$, $\delta_1 = \alpha$, $\delta_k = \alpha\beta^{k-1}$

- Makes the problem time-inconsistent: the optimal policy you would choose for the continuation value after T is not the same if you maximize it in expectation from 0 or at T.

Learning problems

- Bayesian learning: Uncertainty about some model parameters
  - ex: variance and return of a stock market
  - agent models this uncertainty as a distribution
  - agent updates his priors after observing the result of his actions
  - actions are taken optimally taken into account the revelation power of some actions
- Is it good?
  - clean: the *rational* thing to do with uncertainty
  - super hard: the state-space should contain all possible priors
  - mathematical cleanness comes with many assumptions
- Used to estimate rather big (mostly linear) models

Learning problems (2)

- Reinforcement learning
  - model can be partially or totally unknown
  - decision rule is updated by observing the reward from actions
    - no priors
  - solution does not derive directly from model
    - can be used to solve dynamic programming problems
- Good solutions maximize a criterion similar to lifetime reward but are usually not optimal:
  - usually evaluated by replaying the game many times
  - tradeoff exploration / exploitations

Examples

Examples (2)

Finite horizon DMDP

Finite horizon DMDP

When $T < \infty$. With discrete action the problem can be represented by a tree.

[Graph]

Finite horizon DMDP

- Intuition: backward induction.
  - Find optimal policy $x_T(s_T)$ in all terminal states s_T. Set $V_T(s_T)$ equal to $r(s_T, \pi_T)$
  - For each state $s_{k-1} \in S$ find $x_{k-1} \in X(s_{k-1})$ which maximizes

$$V_{k-1}(s_{k-1}) = \max_{x_{k-1}(s_{k-1}) \in X(s_{k-1})} r(s_{k-1}, x_{k-1}) + \delta \underbrace{\sum_{s_k \in S} p(s_k | s_{k-1}, x_{k-1}) V_k(s_k)}_{\textit{expected continuation value}}$$

- Policies $x_0(), ...x_T()$ are "Markov-perfect": they maximize utility on all subsets of the "game"
  - also from t=0

Remarks

- Can we do better than this naive algorithm?
  - not really
  - but we can try to limit S to make the maximization step faster
  - exclude a priori some branches in the tree using knowledge of the problem

Infinite horizon DMDP

Infinite horizon DMDP

- Horizon is infinite:

$$V(s; x()) = \max E_0 \sum_{t=0}^{\infty} \delta^t r(s_t, x_t)$$

- Intuition:
  - let's consider the finite horizon version $T < \infty$ and $T >> 1$
  - compute the solution, increase T until the solution doesn't change
  - in practice: take an initial guess for V_T then compute optimal V_{T-1}, V_{T_2} and so on, until convergence of the Vs

Infinite horizon DMDP (2)

- This is possible, it's called *Successive Approximation* or *Value Function Iteration*
  - how fast does it converge? *linearly*
  - can we do better? yes, *quadratically*

Successive Approximation

- Consider the decomposition:

$$V(s;x()) = E_0 \sum_{t=0}^{\infty} \delta^t r(s_t, x_t) = E_0 \left[r(s, x(s)) + \sum_{t=1}^{\infty} \delta^t r(s_t, x_t) \right]$$

or

$$V(s;x()) = r(s, x(s)) + \delta \sum_{s'} p(s'|s, x(s)) V(s'; x())$$

Successive Approximation (2)

- Taking continuation value as given we can certainly improve the value in every state $\tilde{V}$ by choosing $\tilde{x}()$ so as to maximze

$$\tilde{V}(s; x(), \tilde{x}()) = r(s, \tilde{x}(s)) + \delta \sum_{s'} \pi(s'|s, \tilde{x}(s)) V(s'; x())$$

- By construction: $\forall s, \tilde{V}(s, \tilde{x}(), x()) > V(s, x())$
  - it is an "improvement step"
- Can $V(s, \tilde{x}())$ be worse for some states than $V(s, \tilde{x}())$?
  - actually no

Bellman equation

Idea: it should not be possible to improve upon the optimal solution. Hence the optimal value V and policy $x^\star$ should satisfy:

$$\forall s \in S, V(s) = \max_{y(s)} r(s, y(s)) + \delta \sum_{s' \in S} \pi(s'|s, y(s)) V(s')$$

with the maximum attained at $x(s)$. This is referred to as the **Bellman equation**.

Conversely, it is possible to show that a solution to the Bellman equation is also an optimal solution to the initial problem.

Bellman operator

- The function $G : V \to \max_{y(s)} r(s, y(s)) + \delta \sum_{s' \in S} \pi(s'|s, y(s)) V(s')$ is known as the **Bellman operator**.
- Optimal value if a fixed point of G
- Can we show it's a contraction mapping ?

Blackwell's theorem

- Let $X \subset R^n$ and let $\mathcal{C}(X)$ be a space of bounded functions $f : X \to R$, with the sup-metric. $B : \mathcal{C}(X) \to \mathcal{C}(X)$ be an operator satisfying two conditions:
  1. (monotonicity) if $f, g \in \mathcal{C}(X)$ and $\forall x \in X, f(x) \leq g(x)$ then
  $\forall x \in X (Bf)(x) \leq (Bg)(x)$
  2. (discounting) there exists some $\delta \in]0, 1[$ such that:
     $B.(f + a)(x) \leq (B.f)(x) + \delta a, \forall f \in \mathcal{C}(X), a \geq 0, x \in X$
- Then B is a contraction mapping with modulus δ.

Successive Approximation

- Using the Blackwell's theorem, we can prove the Bellman operator is a contraction mapping (do it).
- This justifies the Value Function Iteration algorithm:
  - choose an initial V_0
  - given V_n compute $V_{n+1} = G(V_n)$
  - iterate until $|V_{n+1} - V_n| \leq \eta$
- Policy rule is deduced from V as the maximand in the Bellman step

Successive Approximation (2)

- Not that convergence of V_n is geometric
- But x_n converges in finite time (X is finite)
  - surely the latest iterations are suboptimal
  - they serve only to evaluate the value of $x^\star$
- In fact:
  - V_n is never the value of $x_n()$
  - should we try to keep both in sync?

Policy iteration for DMDP

- Choose initial policy $x_0()$
- Given initial guess $x_n()$
  - compute the value function $V_n = V(; x_n)$ which satisfies $\forall s, V_n(s) = r(s, x_n(s)) + \delta \sum_{s'} \pi(s'|s, x_n(s)) V_n(s')$
  - improve policy by maximizing in $x_n()$

$$\max_{x_n()} r(s, x_n(s)) + \delta \sum_{s' \in S} \pi(s'|s, x_n(s)) V_{n-1}(s')$$

- Repeat until convergence, i.e. $x_n = x_{n+1}$
- One can show the speed of convergence (for V_n) is *quadratic*
  - it corresponds the the Newton-Raphson steps applied to $V \to G(V) - V$

How do we compute the value of a policy?

- Given x_n, goal is to find $V_n(s)$ in $\forall s, V_n(s) = r(s, x_n(s)) + \delta \sum_{s'} \pi(s'|s, x_n(s)) V_n(s')$
- Two(three) approaches:
  1. simulate the policy rule and compute $E\left[\sum_t \delta^t r(s_t, x_t)\right]$ with Monte-Carlo draws
  2. successive approximation: - put V_k in the rhs and recompute the lhs V_{k+1}, replace V_k by V_{k+1} and iterate until convergence
  3. solve a linear system in V_n
- For 2 and 3 it helps representing a linear operator M such that $V_{n+1} = R_n + \delta M_n.V_n$

Another approach consist: compute $\left(\sum_{t\geq 0} \delta^t M_n^t\right)$