SCILAB *à l'École nationale des ponts et chaussées*

http://cermics.enpc.fr/scilab

Halmstad 2006

Dynamic programming and Markov chains

Jean-Philippe CHANCELIER

22 novembre 2006 (dernière date de mise à jour)

Table des matières

1 Finite horizon problem

Let $(X_n)_{n\in\mathbb{N}}$ be a finite Markov chain with transition matrix $M^{(n)}$. We want to recursively compute :

$$v_n(x) = \mathbb{E}\left[\sum_{k=n}^{T-1} \frac{1}{(1+r)^{k-n}} c_k(X_k) + \frac{1}{(1+r)^{T-n}} f(T, X_T) | X_n = x\right],$$

Question 1 *Write a first routine which returns a random stochastic matrix M of size* `NxN`*. In scilab there is a function called* `genmarkov` *that you can also check if you want.*

```
function M=my_gen_markov(n);
  M=rand(...)
  ...
endfuncion
```

Question 2 *Choose a state size, generate a stochastic matrix M, then generate and plot some samples of the homogeneous Markov chain with states in $[1, N]$ described by M (using* `grand` *in Scilab).*

```
n=10;
M=my_gen_markov(n)
// generate and draw some typical samples
T=4;
m=5;// number of samples
Y=grand(...)
Y=[ones(m,1),Y]; // add the first state to the sampled trajectories
// draw the first trajcetory
plot(Y(1,:))
```

Question 3 *Choosing an instantaneous cost c and a final cost f recusively compute the value function $v^n(x)$ and draw the result as a surface (i.e $v(t, x)$).*

```
function y=c(x); y=... endfunction
function y=f(x); y=... endfunction

states=... // possible states
Cv=c(states) // vector of c possible values
```

```
Kv=f(states) // vector of f possible values
r=0.05; // a discount factor
V=...  // initialize V to fix dimensions
V(:,T) = ...  // the final value of V for time T
for i=T-1:-1:1
  V(:,i) = ...
end
```

Question 4 *Using samples of the markov chain for a given starting state evaluate by Monte Carlo the cost function* $v_0(1)$ *and compare the results with previous question.*

```
m= ... // number of Monte Carlo
X0=ones(m,1); // initial state for all samples
X=grand(...
X=[X0,X];
// fix n
// Approximate the cost at time n  by Monte Carlo
n=1;
Cm= mean(.... f(X(:,T)));
for i=(T-1):-1:n;
  Cm = mean(...
end
Cm - C(1,n)
```