Le vendeur de journaux sur un horizon T

Jean-Philippe Chancelier

March 23, 2018

Contents

1 Le problème du vendeur de journaux à un pas de temps

On reprend ici le code du T.P précédent et on va utiliser la loi de demande discrète sur trois points que l'on avait considéré sur le problème à un pas de temps.

```
// -*- Mode:scilab -*-

// une loi discrète
wi=[30,50,80];
pi=[1/2,1/4,1/4];
titre=sprintf("loi discrète sur %d valeurs",size(wi,'*'));
mu=pi*wi';// la moyenne
// un echantillong de taille N
N=100000;
P=ones(size(wi,'*'),1)*pi;
Y=grand(N,'markov',P,1);
W=wi(Y);

if %f then
  function graphique_loi(titre,valeurs,probas)
    // dessin de la loi
    xset('window',1);
    xbasc();
    plot2d3(wi,pi,rect = [0,0,max(valeurs)+10,min(max(probas)+0.2,1)],style = 2);
```

```
    xtitle(titre);
    // dessin de la fonction de repartition
    xset('window',2);
    xbasc();
    plot2d2([0,wi],[0,cumsum(pi)],rect = [0,0,max(valeurs)+10,1.1],style = 2)
    plot2d([0,wi],[0,cumsum(pi)],rect = [0,0,max(valeurs)+10,1.1],style = -3)
    xtitle(sprintf('Fonction de repartition de la %s',titre));
  endfunction

  graphique_loi(titre,wi,pi);
  xclick();
end

// paramètres de la fonction cout

c=10,cm=20,cs=5;cf=200;

// la fonction coût j(u,w)

function y=j(u,w)
  y=c*u+cs*max(u-w,0)+cm*max(w-u,0)
endfunction

// La fonction J(u) pour la loi binomiale

function y=J(u)
  y=c*u+cs*pi*max((u-wi'),0)+cm*pi*max((wi'-u),0);
endfunction

// graphique de la fonction J(u) et recherche du minimum

if %t then
  U=-10:100;
  Ju=[];
  for u=U do Ju=[Ju,J(u)];end
  xset('window',1);
  xbasc();
  plot2d(U,Ju,style = 2);
  xtitle("La fonction J");
  // recherche du minimum
  [m,kmin]=min(Ju);
  uopt=U(kmin);
```

```
  S=uopt;
  printf("Nombre optimal de journaux a commander %f\n",U(kmin));
  printf("Moyenne de la demande %f\n",mu);
  plot2d(U(kmin),m,style = -4);// dessin
  plot2d3(U(kmin),m);
  xclick();
end

// On regarde maintenant un cas ou le vendeur à déjà des journaux
// et ou il paye un coup fixe si il commande des journaux
// on voit une stratégie [s,S]

function y=Jtilde(u,x)
  y=cf*(u > 0)+J(u+x)-c*x
endfunction

// On calcule pour x variant de 0 à 2*max(wi)
// la commande optimale de journaux correspondante

xuopt=[];
xv=0:1:2*max(wi);
for x0=xv do
  U=0:2*max(wi);
  Ju=[];for u=U do Ju=[Ju,Jtilde(u,x0)];end
  g=[];
  [m,kmin]=min(Ju);
  xuopt=[xuopt,U(kmin)];
end

I=find(xuopt==0);
s=xv(I(1)-1);

// vérifier que la stratégie optimale est [s,S];

xset('window',1);
xbasc();
plot2d2(xv,xuopt,style = 2);
plot2d2(xv(1:s),xv(1:s)+xuopt(1:s),style = 1);
xtitle(sprintf("Valeur de s=%d et de S=%d",xv(s),xv(s-1)+xuopt(s-1)));

//
```

```
// trouver s, S=uopt (calculé avant)
x=0:2*max(wi);
Jv=[];
for i=1:size(xv,'*') do Jv=[Jv,J(x(i))];end
costs=Jv-(cf+J(uopt));
I=find(costs <= 0);
if isempty(I) then
  s=0;
else
  s=x(I(1)-1)
end

printf("On trouve s=%d et S=%d\n",s,S);
```

Question 1 *Faites éxecuter le code précédent dans scicoslab et vérifiez que la stratégie optimale est bien de la forme* $[s, S]$.

1.1 On cherche maintenant à résoudre le problème du vendeur de journaux sur un horizon T

On propose ici un squelette de programme qui implémente l'équation de Programmation dynamique. Il faut compléter ce programme et résoudre le problème sur un horizon T. À l'instant $t = 1$ le vendeur de journaux fait sa première commande et à l'instant T il ne commande plus. Le cas $T = 2$ correspond donc au problème du vendeur sur une journée.

```
// -*- Mode:scilab -*-
// Vendeur de journaux à T-pas de temps

exec('vendeur-T-un.sci');

alpha=1.0;

function y=ct(u,x)
  // coût instantané pour t autre que la date de départ
  y=cf*(u > 0)+c*u+alpha*cs*max(x,0)+alpha*cm*max(-x,0)
endfunction

function y=c0(u)
  // coût instantané pour t=1
  y=cf*(u > 0)+c*u
endfunction

function y=K(x)
```

```
  // coût final
  y=cs*max(x,0)+cm*max(-x,0)
endfunction

// on transforme la dynamique pour rester dans un domaine
// borné [-100,100]

xmin=-100;xmax=100;

function y=f(x,u,w)
  y=min(max(x+u-w,xmin),xmax)
endfunction

// VT
// On commencera avec T=2 qui doit donner le problème à un pas de temps


T=2;// l'Horizon

etats=xmin:xmax;// les etats possibles dans un domaine borné
allu=0:xmax;// les commandes de journaux possible

// La fonction v_T(x) = K(x)
VT=alpha^T*K(etats);

// On va calculer V_t pour T-1,...,1
// et stocker les résultats dans une liste
// On stocke aussi la commande optimale

L=list(VT);// liste qui permet de stocker (V_T,V_{T-1},.....,V_1)
U=list(0);// liste qui permet de stocker (-- ,U_{T-1},.....,U_1)

// boucle rétrograde en temps
for t=T-1:-1:1 do
  printf("t=%d\n",t);
  Vt=zeros(1,size(etats,'*'));
  Ut=%nan*ones(1,size(etats,'*'));
  Vtp1=L($);
  // On cherche a calculer ici Vt et Ut
  // Vtp1 est la fonction de Bellman au temps t+1
  for i=1:size(etats,'*') do
    x=etats(i);
```

```
    mcost=%inf;
    // boucle sur les commandes possibles
    for k=1:size(allu,'*') do
      u=allu(k);
      // on calcule dans cost le cout associé a la commande u
      // initialisation:
      cost=0;
      // faire une boucle sur les aléas pour calculer
      // E[ ct(u,x) + V_{t+1}(f(x,u,W))]
      for xxx=zzz do
        //AFAIRE ...
        // xtp1: utiliser la dynamique de la chaine
        //AFAIRE xtp1 = ...
        ix=find(xtp1==etats);// donne l'indice de xtp1 dans le
        // vecteurs des etats
        // actualiser le cout
        //AFAIRE cost = cost + ...
      end
      // Il faut maintenant comparer
      // cost au meilleur trouvé jusque la (on minimise)
      // mcost et mettre à jour mcost
      // et kumin (l'indice de la commande qui a donné le meilleur coût)
      //AFAIRE if ... ...  end
    end
    // On stocke pour l'état x d'indice i le coût optimal
    Vt(i)=mcost;
    // et la commande optimale
    Ut(i)=allu(kumin);
  end
  // On range Vt et Ut dans la liste
  L($+1)=Vt;
  U($+1)=Ut;
end

// dessin de la fonction de Bellman au temps t=1;
xset('window',1);
xbasc();
plot2d2(etats,L($))

// dessin de la commande optimale au temps t=1;
xset('window',1);
xbasc();
```

```
plot2d2(etats,U($))
```

Question 2

- *Compléter le code pour résoudre l'équation de Bellman.*
- *Faire tourner le code pour $T = 2$ et retrouver les résultats du problème à un pas de temps.*
- *Faire tourner le code pour $T = 8$ et vérifier que l'on obtient aussi une stratégie $[s, S]$.*
- *Simuler des trajectoires de la chaîne de Markov controlée avec la commande optimale et tracer des trajectoires*
- *Caluler à partir de simulations pour un point inital fixé x un coût moyen et le comparer à la valeur de Bellman calculée.*