

2009年度応用マクロ経済学講義ノート 不 完備市場モデル

阿部修人

平成 21 年 12 月 11 日

概要

1 不完備市場のモデル化

最適成長モデルや標準的 RBC モデルは、厚生経済学の第一基本定理が成立する経済モデルを用い、経済成長や景気循環の様々な側面の分析においてある程度の成功を収めている。しかしながら、このような「効率的」な経済モデル、すなわち市場が完備かつ競争的な経済モデルでは分析が不能な経済現象は多く存在しており、近年ではマクロ経済的諸側面においても、「完備」でなく、「競争的」でもなく、また取引費用等、市場行動そのものに費用がかかるようなモデリングが必要であるという認識が広まっている。不完備市場の理論分析は古くは Radner や Cass 等により行われているが、近年の研究では不完備性の Empirical Implications を重視している点に特徴がある。¹

Aiyagari(1994) は、1993 年に Mineapolis 連銀により開催されたコンファレンスの要約において、不完備市場を考える必要として下記の七点を挙げている。

(1) 完備市場の下では、個人の所得リスクは保険によりカバーされるため、全ての家計消費は、唯一カバーされないマクロの動向と完全に同じ動きをすればであるが、実際にはそうになっていない。

(2) 家計消費の Volatility は大きいですが、一方マクロの消費はスムーズになっている。

¹ 独占的競争モデルや価格の硬直性、寡占市場の分析はマクロ経済学においては長い歴史があり、Hicks の Value and Capital の中で、独占市場が価格の硬直性を作り出し、市場均衡とは異なる一時均衡を作り出す可能性があるとして述べている。無論、1980 年代から延々と続いている New Keynesian 達のモデルもこの延長上にある。経済モデルが非効率な均衡を導き出すモデルは財政学ではその中心にあるし（でなければ政府は所得再分配しかすることがない）、常に応用経済学で用いられてきた。動的計画法に基づくカリブレーションでは、状態変数を少数に制限する必要があり、必ずしも、今までのケインジアンや財政学者達の貢献を十分に生かしてきていない。これは、裏を返せば、数値解析技術の発展により分析可能になる経済現象は極めて多いということでもある。

(3) 家計所得の階層間移動は頻繁に起きている。たとえば、アメリカでは10年間で3/5の家計が所得分位の位置を変えている。

(4) Mehra - Prescott のパズル

(5) 景気循環の Propagation Mechanism として、標準的RBCでは技術ショックの自己相関が大方説明してしまい、モデルの必要性が感じられない。

(6) 企業投資をマイクロで見ると極めて Lumpy なので、産業別にみるとスムーズになっている。

(7) 消費をスムージングしないと仮定した場合の welfare costs は、完備市場の下では極めて小さく (1 Quarter で10セントくらい)、near rational な家計を考えると、消費=所得と仮定する極端な消費関数を否定できなくなってしまう。

これらは、90年代前半に特に集中して研究されたものであり、現在までにかかなりの進展が得られている。ミクロのパネルデータへのアクセスが容易になり、多くの実証研究が家計消費に関して行われてきているが、その中には完備市場モデルと整合的ではない結果が得られている。Aiyagari の後、Kocherlakota や Attanasio, Perri, Kruger 等の一連の研究は、不完備市場の考察によるデータと理論の乖離を埋めようとするものであり、現在、もっとも進展速度の速い分野でもある。

Aiyagari モデルと呼ばれるものは、今振り返ると、極めて単純であるが、しかしその後の研究のベンチマークとも言えるものになっている。理論構造そのものは Bewley による古典的な研究に準拠しているが、比較的小さなプログラムコードで実行可能であり、かつ様々な応用可能性を兼ね備えている。特徴としては (1) 一般均衡であること (2) Ex ante に同質な家計間に Ex post に異質性が発生すること (3) 生産経済であり資本蓄積が存在すること (4) 流動性制約により、借りに制約が存在すること (5) 個人の借りに制約を緩めることで inside money を導入可能であること、など等、望ましい性質が多く含まれている。また、モデルがシンプルであることから、離散近似で分析可能であることも良い点である。²

1.1 流動性制約と一般均衡 (Aiyagari モデル)

家計: 無数に存在するが、選好同一。初期状態において資産・労働 status も同一。無限視野。借りに不可能。金利 r , 賃金 w は所与。

労働: 家計は一単位の労働を供給するが、働ける場合は $(s=)1$ 、働けない場合は $(s=)0$ の供給を行う。失業するかしないかは家計にとり完全に外生であり、家計間での相関はない。すなわち、マクロ経済的な失業確率の変動はな

²Aiyagari の論文では Piece wise linear approximation で解いている。

い。今期失業するかしないかは、前期の就業状態に依存するマルコフチェーン (2×2 の推移確率行列 P) に従う。

生産: 経済全体の資本および労働投入の集計量を投入とするコブ・ダグラス型技術により消費財が生産される。

政府: 何もしない。または失業給付を lump sum tax により finance する、あるいは貨幣 (outside money) を供給する。

家計の Bellman Equation

$$v(k, s) = \max_{c, k'} [u(c) + \beta E[v(k', s'|s)]]$$

s.t.

$$c + k' = rk + ws + (1 - \delta)k, 0 \leq k'$$

すなわち、家計は今期の就業状態 (s) と資産水準 (k) を下に、今期の消費と来期の資産を決める。このとき、今期の就業状態は来期の就業状態に関して情報を有しているため、その情報を用いて将来予測を行っている。この DP における資産水準の policy function を $k' = g(k, s)$ とする。

家計にとっての状態変数は s と k であるが、我々の関心は個別家計の動向ではなく、経済全体の変数の状態に関心がある。すなわち、 s と k 平均値を我々は知りたい。そこで、 s と k の推移を描写するマルコフチェーンを作り、その定常分布を求める。そのためには、policy function および推移確率 P から s, k のマルコフチェーンを作成する必要がある。

$$\text{prob}(k_{t+1} = k', s_{t+1} = s') = \lambda(k', s')$$

とする。この確率分布の中で、定常分布となるものを見つけるのがとりあえずの目標である。

まず、 k と s をグリッドに分割し、グリッドに入るか、入らないかという情報を用いて推移確率行列を計算する。

$$I(k', s, k) = 1 \quad \text{if } k' = g(k, s) \quad \text{otherwise zero.}$$

とする Indicator Function を考える。これは、 s, k の下で、次の期に k' になるか否かを判定する。 k' と s' のペアに到達する確率を考えると、

(k, s) から (k', s') に移行する確率は

$$\text{prob}(k', s'|k, s) = \text{prob}(s_{t+1} = s'|s_t = s) I(k', s, k)$$

で与えられる。これは推移確率行列の成分を与える。定常分布は

$$\lambda(k', s') = \sum_k \sum_s \lambda(k, s) \text{prob}(s_{t+1} = s'|s_t = s) I(k', s, k)$$

で、 λ に関する不動点を求めれば、それが定常分布となる。均衡では、個人の資本ストックや労働供給が集計されてマクロ資本および労働投入量になるから、

$$K = \int k\lambda(k, s) dkds$$

$$N = \int s\lambda(k, s) dkds$$

したがって、要素価格は

$$r = r(K, N) = \alpha \left(\frac{K}{N} \right)^{\alpha-1}$$

$$w = w(K, N) = (1 - \alpha) \left(\frac{K}{N} \right)^{\alpha}$$

で与えられる。

1.2 Aiyagari モデルの単純版

このモデルを解くコードは有名であり、また、数値計算の精度の限界が明らかになるものでもある。得られる分布の大まかな形状は正しいが、グリッドを増やしても細かな奇妙な振動は消滅せず、また、収束基準を厳しくすると、ループに陥り、Bellman Equation が収束しなくなる。

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%                               SMODEL 2
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% set parameter values
%
sigma = 1.50; % risk aversion
beta = 0.98; % subjective discount factor
prob = [.8 .2; .5 .5]; % prob(i,j) = probability (s(t+1)=sj | s(t) = si)
delta = 0.97; % 1 - depreciation
A = 1.00; % production technology
alpha = 0.25; % capital's share of income
theta = 0.05; % non-rental income if unemployed is theta*wage
Kstart = 10.0; % initial value for aggregate capital stock

```

```

g = 0.20; % iteration  $\theta$  relaxation parameter
%
% form capital grid
%
maxkap = 20; % maximum value of capital grid
inckap = 0.05; % size of capital grid increments
nkap = round(maxkap/inckap+1); % number of grid points
%
% calculate aggregate labor supply
%
[v1,d1]=eig(prob');
[dmax,imax]=max(diag(d1));
probst1=v1(:,imax);
ss=sum(probst1);
probst1=probst1/ss;
pempl=probst1(1,1);
%
N = 1.0*pempl + theta*(1-pempl);
%
% loop to find fixed point for agregate capital stock
%
liter = 1;
maxiter = 50;%iteration に時間がかかるので、
toler = 0.001;% 収束条件に関しては注意すること。あまり厳しくすると収
束しなくなる
metric = 10;
K = Kstart;
disp('ITERATING ON K');
disp('');
disp(' liter metric meanK Kold');
%
% 均衡総資本を求めるループ
%
while (metric > toler) & (liter <= maxiter);
%
% calculate rental rate of capital and wage
%
wage = (1-alpha) * A * K^(alpha) * N^(-alpha);
rent = (alpha) * A * K^(alpha-1) * N^(1-alpha);

```

```

%
% tabulate the utility function such that for zero or negative
% consumption utility remains a large negative number so that
% such values will never be chosen as utility maximizing
%
util1=-10000*ones(nkap,nkap); % utility when employed
util2=-10000*ones(nkap,nkap); % utility when unemployed
for i=1:nkap;
    kap=(i-1)*inckap;
    for j=1:nkap;
        kapp = (j-1)*inckap;
        cons1 = wage + (rent + delta)*kap - kapp;
        if cons1 > 0;
            util1(j,i)=(cons1)^(1-sigma)/(1-sigma);
        end;
        cons2 = theta*wage + (rent + delta)*kap - kapp;
        if cons2 > 0;
            util2(j,i)=(cons2)^(1-sigma)/(1-sigma);
        end;
    end;
end;
%
% initialize some variables
%
v = zeros(nkap,2);
decis = zeros(nkap,2);
test = 10;
[rs,cs] = size(util1);
%
% iterate on Bellman's equation and get the decision
% rules and the value function at the optimum
%
while test ~ = 0;
    for i=1:cs;
        r1(:,i)=util1(:,i)+beta*(prob(1,1)*v(:,1)+ prob(1,2)*v(:,2));
        r2(:,i)=util2(:,i)+beta*(prob(2,1)*v(:,1)+ prob(2,2)*v(:,2));
    end;
    [tv1,tdecis1]=max(r1);
    [tv2,tdecis2]=max(r2);
end;

```

```

    tdecis=[tdecis1' tdecis2'];
    tv=[tv1' tv2'];
    test=max(any(tdecis-decis));
    v=tv;
    decis=tdecis;
end;
decis=(decis-1)*inckap;
%
% form transition matrix
% trans is the transition matrix from state at t (row)
% to the state at t+1 (column)
% The eigenvector associated with the unit eigenvalue
% of trans' is the stationary distribution.
%
g2=sparse(cs,cs);
g1=sparse(cs,cs);
for i=1:cs
    g1(i,tdecis1(i))=1;
    g2(i,tdecis2(i))=1;
end
trans=[ prob(1,1)*g1 prob(1,2)*g1; prob(2,1)*g2 prob(2,2)*g2];
trans=trans';
probst = (1/(2*nkap))*ones(2*nkap,1);
test=1;
while test > 10^(-8);
    probst1 = trans*probst;
    test = max(abs(probst1-probst));
    probst = probst1;
end;
%
% vectorize the decision rule to be conformable with probst
% calculate new aggregate capital stock meanK
%
kk=decis(:);
meanK=probst'*kk;
%
% calculate measure over (k,s) pairs
% lambda has same dimensions as decis
%
```

```

lambda=zeros(cs,2);
lambda(:)=probst;
%
% calculate stationary distribution of k
%
[v1,d1]=eig(prob');
[dmax,imax]=max(diag(d1));
probst1=v1(:,imax);
ss=sum(probst1);
probst1=probst1/ss;
probk=sum(lambda'); % stationary distribution of 'capital'
probk=probk';
%
% form metric and update K
%
Kold = K;
Knew = g*meanK + (1-g)*Kold;
metric = abs((Kold-meanK)/Kold);
K = Knew;
disp([ liter metric meanK Kold ]);
liter = liter+1;
end;
%
% print out results
%
disp('PARAMETER VALUES');
disp("");
disp(' sigma beta delta A alpha theta');
disp([ sigma beta delta A alpha theta]);
disp("");
disp('EQUILIBRIUM RESULTS ');
disp("");
disp(' K N wage rent');
disp([ Kold N wage rent ]);
%
% simulate life histories of the agent
%
disp('SIMULATING LIFE HISTORY');
k = Kold; % initial level of capital

```

```

n = 100; % number of periods to simulate
s0 = 1; % initial state
hist = zeros(n-1,2);
cons = zeros(n-1,1);
invest = zeros(n-1,1);
grid = [ (0:inckap:maxkap)'];
[chain,state] = markov(prob,n,s0);
for i = 1:n-1;
    hist(i,:) = [ k chain(i) ];
    I1 = round(k/inckap) ;
    I2 = round(k/inckap) + 1;
    if I1 == 0;
        I1=1;
        disp('N.B. I1 = 0');
    end;
    if I2 > nkap;
        I2 = nkap;
        disp('N.B. I2 > nkap');
    end;
    weight = (grid(I2,1) - k)/inckap;
    kprime = weight*(decis(I1,chain(i))) + (1-weight)*(decis(I2,chain(i)));
    if chain(i) == 1;
        cons(i) = wage + (rent + delta)*k - kprime;
    elseif chain(i) == 2;
        cons(i) = wage*theta + (rent + delta)*k - kprime;
    else;
        disp('something is wrong with chain');
    end;
    chain
end;
k = kprime;
invest(i) = kprime;
end;
%
%
subplot(2,2,1),plot((1:n-1)',invest,(1:n-1)',cons);
title('MODEL 2: INVESTMENT AND CONSUMPTION');
print histmod2
disp('Covariance matrix');
disp([cov(cons,invest)]);

```

```

%
% calculate income distribution
%
income = [ (rent*grid + wage) (rent*grid + wage*theta) ] ;
[ pinc, index ] = sort(income(:));
plambda = lambda(:);
%
subplot(2,2,2), plot(pinc,plambda(index));
title('MODEL 2: INCOME DISTRIBUTION');
xlabel('INCOME LEVEL');
ylabel('% OF AGENTS');
print distmod2
%
% calculate capital distribution
%
subplot(2,2,3),plot(grid,probk);
title('MODEL 2: CAPITAL DISTRIBUTION');
xlabel('CAPITAL GRID');
ylabel('% OF AGENTS');
print capdmod2

%%%%%%%%%%%%%%
% マルコフチェーンで履歴をシミュレートする
%

function [chain,state]=markov(T,n,s0,V);
%function [chain,state]=markov(T,n,s0,V);
% chain generates a simulation from a Markov chain of dimension
% the size of T
%
% T is transition matrix
% n is number of periods to simulate
% s0 is initial state
% V is the quantity corresponding to each state
% state is a matrix recording the number of the realized state at time t
%
%
[r c]=size(T);
if nargin == 1;
V=[1:r];

```

```

s0=1;
n=100;
end;
if nargin == 2;
V=[1:r];
s0=1;
end;
if nargin == 3;
V=[1:r];
end;
%
if r ~ = c;
disp('error using markov function');
disp('transition matrix must be square');
return;
end;
%
for k=1:r;
if sum(T(k,:)) ~ = 1;
disp('error using markov function')
disp(['row ',num2str(k),' does not sum to one']);
disp(' it sums to :');
disp([ sum(T(k,:))  ]);
disp(['normalizing row ',num2str(k),"]);
T(k,:)=T(k,+)/sum(T(k,:));
end;
end;
[v1 v2]=size(V);
if v1 ~ = 1 |v2 ~ =r
disp('error using markov function');
disp(['state value vector V must be 1 x ',num2str(r),"]);
if v2 == 1 &v2 == r;
disp('transposing state valuation vector');
V=V';
else;
return;
end;
end
end
if s0 < 1 |s0 > r;

```

```

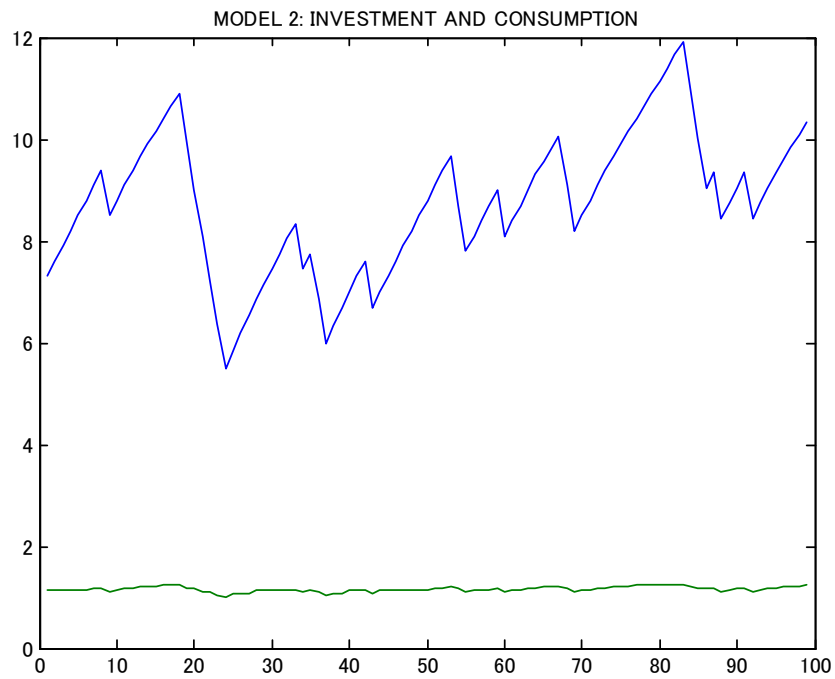
disp(['initial state ',num2str(s0),' is out of range']);
disp(['initial state defaulting to 1']);
s0=1;
end;
%
%T
%rand('uniform');
X=rand(n-1,1);
s=zeros(r,1);
s(s0)=1;
cum=T*triu(ones(size(T)));
%
for k=1:length(X);
state(:,k)=s;
ppi=[0 s'*cum];
s=((X(k)<=ppi(2:r+1)).*(X(k)>ppi(1:r)))';
end;
chain=V*state;

%%%%%%%%%%%%%%
% 出力結果

%
>> MODEL 2
ITERATING ON K
liter metric meanK Kold
1.0000 0.5501 4.4994 10.0000
2.0000 0.4545 4.8553 8.8999
3.0000 0.3396 5.3429 8.0910
4.0000 0.2175 5.9010 7.5413
5.0000 0.1014 6.4818 7.2133
6.0000 0.0336 6.8297 7.0670
7.0000 0.0108 6.9434 7.0195
8.0000 0.0081 6.9475 7.0043
9.0000 0.0039 7.0199 6.9929
10.0000 0.0010 7.0055 6.9983
11.0000 0.0013 7.0090 6.9998
12.0000 0.0030 7.0223 7.0016
13.0000 0.0023 6.9895 7.0058
14.0000 0.0027 7.0215 7.0025

```

15.0000 0.0084 6.9475 7.0063
16.0000 0.0036 7.0199 6.9946
17.0000 0.0013 7.0090 6.9996
18.0000 0.0030 7.0223 7.0015
19.0000 0.0023 6.9895 7.0057
20.0000 0.0010 7.0091 7.0024
PARAMETER VALUES
sigma beta delta A alpha theta
1.5000 0.9800 0.9700 1.0000 0.2500 0.0500
EQUILIBRIUM RESULTS
K N wage rent
7.0024 0.7286 1.3206 0.0458
SIMULATING LIFE HISTORY
Covariance matrix
0.0078 0.1165
0.1165 1.9059



グリッドの分割幅が 0.5 の場合、3GB の Dell のラップトップでは、約 12 秒で計算は終了する。分割幅 0.01 とし、5 倍にすると、三分弱で終了する。

