Contents

1 Prior knowledge processing

This material refers to the book [1] – 6.3{98} and 8.3{248}.

On-line references are:

Coding and use of prior knowledge (Mixtools Guide)

Article "Use of Prior Information in Structure Estimation"

Processing examples follow.

1.1 Prior Knowledge in Structure Estimation

Aim:

To show influence of prior knowledge model in factor structure estimation.

Description:

Observed data are generated by a dynamic simulated mixture. A sufficiently large richest model order *ord* is selected. It specifies the space of possible model structures.

A guess of the system gain serves as the processed prior knowledge. It is specified as a range of values *gain*. The range implies both the expected value of the gain as well as the weight of this piece of knowledge.

The initial model is built and its structure is estimated with and without prior knowledge. The display is done via the internal form of structures – as a list of channels and delays (channel 0 stands for offset), e.g. the structure

```
str = [1 1  2 2
       1 2  0 1]
```

means that the regression vector at a time t is composed of the data value on the channel 1 with delays 1 and 2 (i.e. $d_{1;t-1}$, $d_{1;t-2}$) together with the data value on the channel 2 with delays 0 and 1 ($d_{2;t}$, $d_{2;t-1}$).

Specification:

System: realization of *ndat* two-dimensional data generated by a normal dynamic mixture with one component. The realization is determined by *seed*. View code for details.

Decision: the estimate of the structure of the first factor.

Experience: past data up to *ndat*, prior knowledge about the system gain is either exploited or not.

Ignorance: model structure and parameters.

Loss function: negative v-log-likelihood.

Recommended experiments:

The structure estimation task depends on signal-to-noise ratio, space of possible regressors and the prior knowledge. The experiments should show the contribution of prior knowledge to the structure estimation task. It is worth inspecting the influence of

- extent of learning data, say *ndat* = 100, 500, 200
- process-noise covariances, *covy*= 0.001, 0.01 and *covu* = 1, 0.1
- richest model order *ord* = 5, 4, 3
- gain range, gain = [0.09 1.01] or [0.9 1.1].

run example read Guide article see results view code

Contents

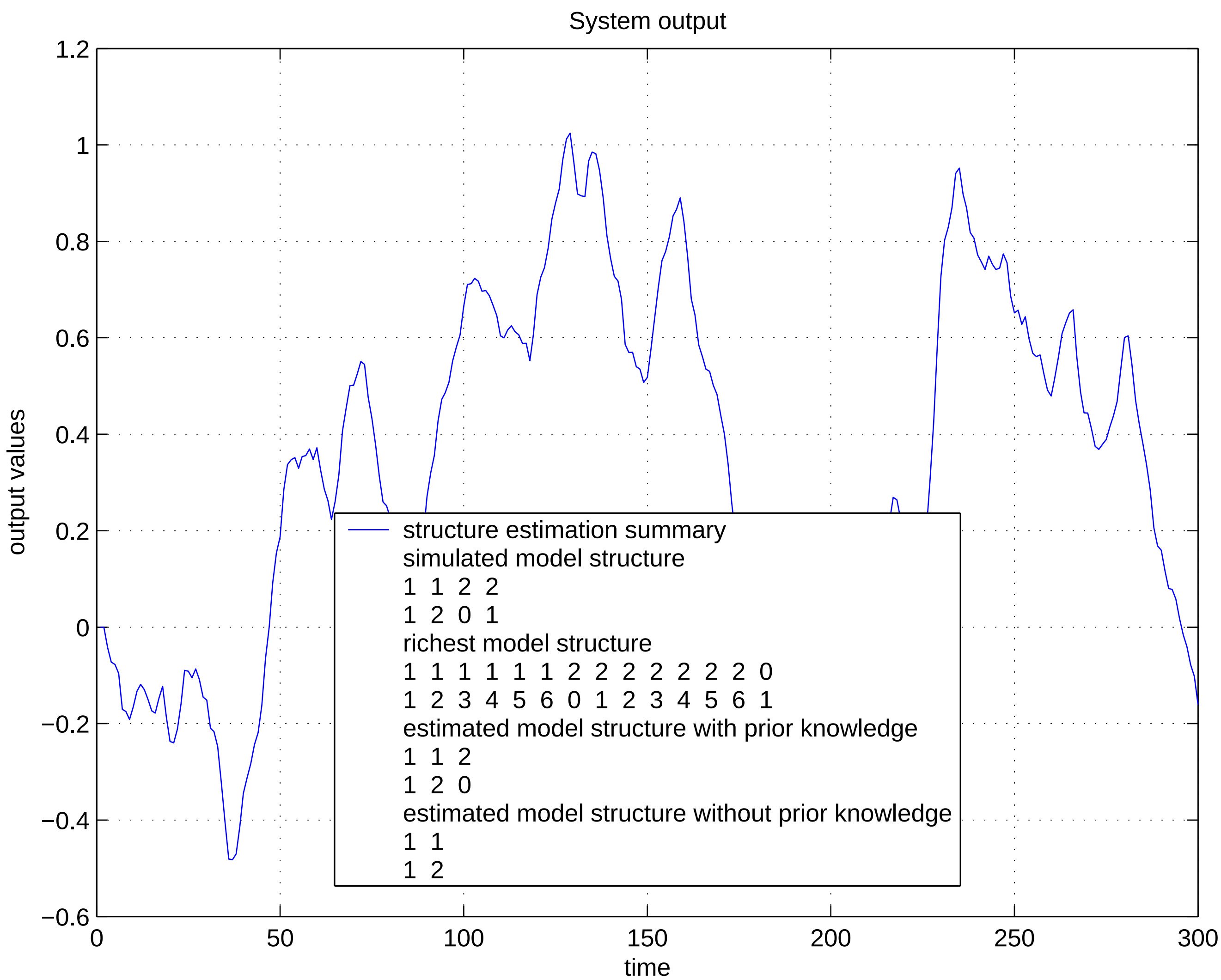
System output
output values
time
1.2
1
0.8
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
0
50
100
150
200
250
300
structure estimation summary
simulated model structure
1 1 2 2
1 2 0 1
richest model structure
1 1 1 1 1 1 2 2 2 2 2 2 2 0
1 2 3 4 5 6 0 1 2 3 4 5 6 1
estimated model structure with prior knowledge
1 1 2
1 2 0
estimated model structure without prior knowledge
1 1
1 2

Figure 1:

1.2 Prior knowledge in estimation

Aim:

study influence of prior knowledge model in estimation.

Description:

Prior knowledge about the system gain is known (1). from it, initial model for estimation is build.

The trajectory of $b0$ and $b1$ regression coefficients are displayed. The show that the start with prior knowledge is better than the default one.

Specification:

System: realization of *ndat* two-dimensional data generated by a normal dynamic mixture of one component. the noise covariance is *covy*. The input covariance is is *covu*. The realization is determined by *seed*. View code for details.

Decision: is it better to start estimation from prior model than from defaults ?

Experience: Past data up to *ndat* and dynamic factor structure.

Ignorance: regression coefficients.

Recommended experiments:

study influence of
- data sample size (e.g. *ndat* = 100, 200)
- process-noise variances (*covy*= 1, 2 and *covu* = 0.1, 1, 10
- try estimation with different values of *diaCth* (= 0.1, 1, 10)

run example article read Guide see results see coding

Contents

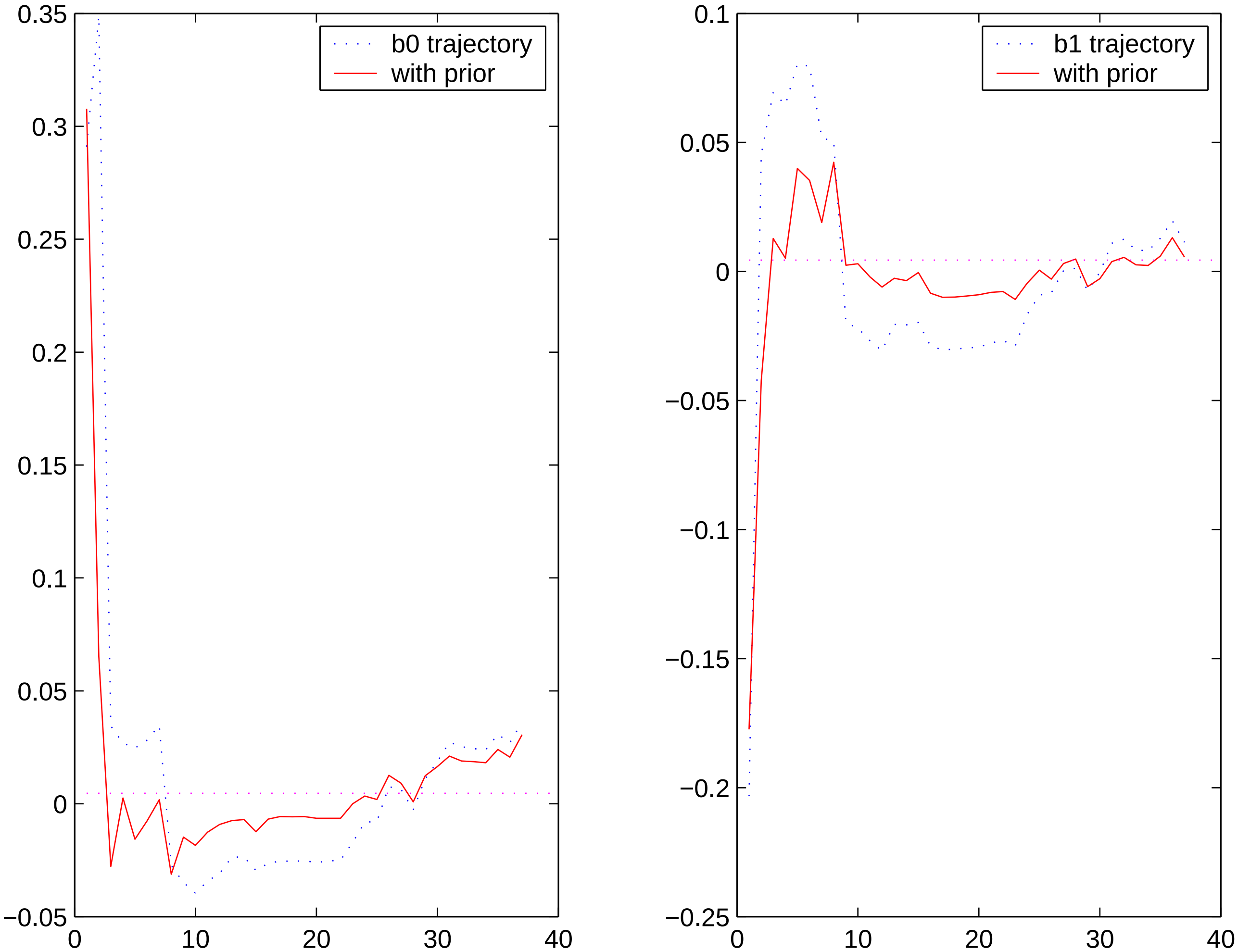

Figure 2: Results with defaults

1.3 Different Prior Knowledge in Structure Estimation

Aim:

study of influence of prior knowledge of different types in estimation.

Description:

Prior knowledge type is selected from menu. The menu consists off From it, initial model for estimation is build. View code for details.

The trajectory of $b0, b1, b2$ regression coefficients are displayed (input coefficients). The same is displayed for starting from defaults.

Specification:

System: realization of *ndat* two-dimensional data generated by a normal dynamic mixture of one component. the noise covariance is *covy*. The input covariance is is *covu*. The realization is determined by *seed*. View code for details.

Decision: is it better to start estimation from prior model than from defaults ?

Experience: Past data up to *ndat* and dynamic factor structure.

Ignorance: regression coefficients.

Recommended experiments

Observe influence of
- data sample size (e.g. *ndat* = 100, 200)
- process-noise variances (*covy*= 1, 2 and *covu* = 0.1, 1, 10
- try estimation with different values of *diaCth* (= 0.1, 1, 10)

run example | article | read Guide | see results | see coding
Contents

```
function [Eth,str,ts,uref,yref,ssig,sigma,gain,tcons,msig,wcutc,freqc]= system1
msig=0;
chanu = 2;

%references
uref=0;  yref=0;

% system
A=[1.0000   -2.7145    2.4562   -0.7408];
B=[0.0001547   0.0005740    0.0001331 0];
ts    = 0.1;
ssig  = sum(B)/2;
sigma = sum(B)/2;


%timing
   lendata = 100;        % length of design data
   lenpri  = 1000;       % length of prior info
   lenst   = 100;        % length of steady state
   randn('seed', 123);   % fix random generator

% gain
gain =(B*ones(max(size(B)),1))/(A*ones(max(size(A)),1));

% cut zero Bs
lA=length(A); lB=length(B);
while  B(lB)==0,
   lB=lB-1;
end

% parameters
if chanu==1, Eth  = [B(1:lB), -A(:,2:length(A)) ];
else
   Eth  = [-A(:,2:length(A)) B(1:lB)];
end

% structure
if chanu==1
   str1 = [1+ones(1,lB), ones(1,lA-1)];
   str2 = [0:(lB-1), 1:(lA-1)];
else
   str1 = [ones(1,lA-1),1+ones(1,lB)];
   str2 = [1:(lA-1),0:(lB-1)];
end
str  = [str1;str2];
str_org=str;
% time constants
   ii= find(str(1,:)==1);
   a  = Eth(ii);
   A  = [1, -a];
   rr  = roots(A); r=[];
   len=length(rr); j=1;
   while 1
       if isreal(rr(j)), r=[r,rr(j)];      j=j+1;
       else,             r=[r,abs(rr(j))]; j=j+2;
       end
       if j>len, break; end
   end
   jj = find(abs(r)>1e-10);
```

```
   r  = r(jj);
   r  = -log(r);
   r  = abs(r);
   t1 = 1/r(1) * ts;
   if length(r)>1,
      t2 = 1/r(2) * ts;
      if abs(t1-t2)>1e-4
         if t1>t2, tt=[t2,t1]; else, tt=[t1,t2]; end
      end
   else, tt = t1;
   end

   if length(tt)>1
     tcons=[tt(1)*0.99, tt(1)*1.01, tt(2)*0.99, tt(2)*1.01];
     else tcons=[tt*0.99 tt*1.1];
   end

%convertion discrete tf to continuous one
sysd=tf(B, A,ts );
sysc=d2c(sysd);
[num,den]=tfdata(sysc,'v');

[m, ph,w]=bode(sysc);
kk=find(m<=0.01*m(1)); wcutc=w(kk(1));
ll=find(m<=0.5623*m(1)); freqc=w(ll(1));
```

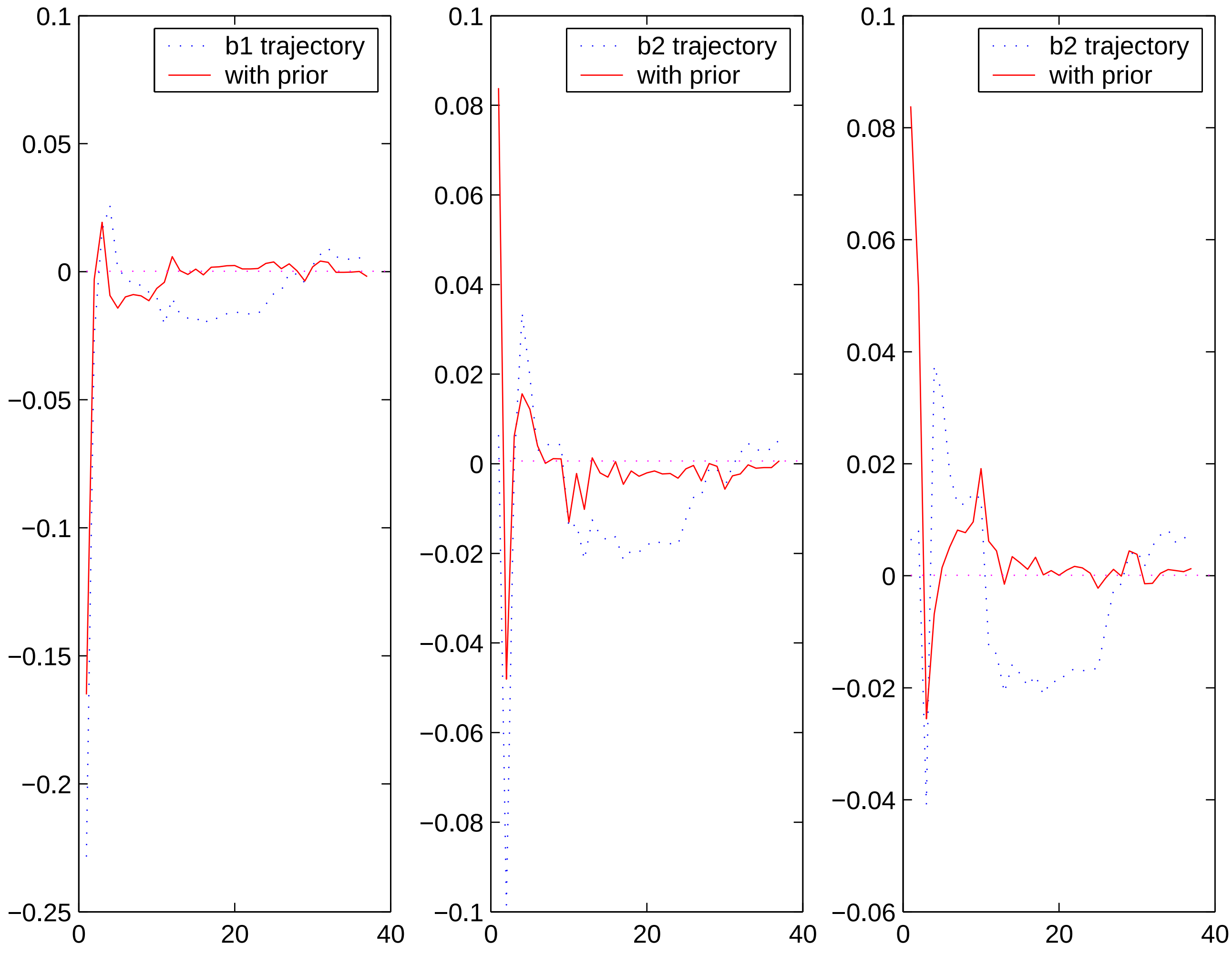

Figure 3: Results with defaults

References

[1] M. Kárný, J. Böhm, T.V. Guy, L. Jirsa, I. Nagy, P. Nedoma, and L. Tesař, *Optimized Bayesian Dynamic Advising: Theory and Algorithms*, Springer, London, 2004, to appear.