

HW1-ECON-482

Kungang Zhang

October 27, 2016

1 Prob 1:

- (1) According to the statement, we use flat prior, so that maximizing the posterior is equivalent to maximizing likelihood (“conditional likelihood” given the initial data). There are two ways to do prediction:

- Moving window:

$$\begin{array}{ccccccc} [& 1, & 2, \dots, & T, \dots, & T+h &] \\ & & [2, \dots, & T, \dots, & T+h, & T+h+1 &] \\ & & & \dots & \dots & \dots & \\ & & & [T, \dots, & T+h, & T+h+1, \dots, & 2T+h-1 &] \\ & & & & \dots & \dots & \dots \end{array}$$

- Extending window:

$$\begin{array}{ccccccc} [& 1, & 2, \dots, & T, \dots, & T+h &] \\ [& 1, & 2, \dots, & T, \dots, & T+h, & T+h+1 &] \\ & & & \dots & \dots & \dots & \\ [& 1, & 2, \dots, & T, \dots, & T+h, & T+h+1, \dots, & 2T+h-1 &] \\ & & & & \dots & \dots & \dots \end{array}$$

Here I tried both cases and get the MSFE:

According to Table 1, looks like the extending window method has smaller MSFE.

where $K = Mp + 1$, $M = \dim(y)$, $p = \max \text{ time lag}$, and the given observations are y_{-p+1}, \dots, y_0 and the values to forecast are y_1, \dots, y_T . Part of these expressions come from [2], Section 5.2. The last two equation mean the parameters $\hat{\sigma}_i^2$ are MLE based on $AR(1)$ model, given $y_{i,0}$. Here, we don't need to know Σ because we only need a point estimation of $\hat{\beta}$. Using the same procedure as in (1), we get the MSFE as: The

MSFE	log-real GDP	log-real GDP	log-GDP deflator	log-GDP deflator
Average growth rate	1 quarter	4 quarters	1 quarter	4 quarters
Extending window	$9.387e - 4$	$1.493e - 4$	$5.444e - 4$	$1.856e - 4$

Table 2: Mean squared forecast error for moving window and extending window.

MSFE for Minnesota priors for 1 quarter forecast is smaller than that for flat prior, but the MSEF for Minnesota priors for 4 quarter forecast is larger.

(3) The marginal likelihood is (as shown in [1] and its supplemental file):

$$\begin{aligned}
P(y|\lambda) = & \left(\frac{1}{\pi}\right)^{\frac{MT}{2}} \frac{\Gamma_M(\frac{T+d}{2})}{\Gamma_M(\frac{d}{2})} |\Omega|^{-\frac{M}{2}} |\Psi|^{\frac{d}{2}} |X^T X + \Omega^{-1}|^{-\frac{M}{2}} \\
& * |\Psi + \hat{\epsilon}^T \hat{\epsilon} + (\hat{B} - \hat{b})^T \Omega^{-1} (\hat{B} - \hat{b})|^{-\frac{T+d}{2}} \\
& \hat{\epsilon} = Y - X\hat{B}
\end{aligned} \tag{9}$$

Maximizing the posterior with flat prior is equivalent to maximizing the marginal likelihood. By using notation $E_\Psi E_\Psi^T = \Psi^{-1}$, $D_\Omega D_\Omega^T = \Omega$, and $\lambda^2 \Phi = \Omega$ or $\lambda D_\Phi = D_\Omega$, the 9 can be simplified as:

$$\begin{aligned}
P(y|\lambda) = & \left(\frac{1}{\pi}\right)^{\frac{MT}{2}} \frac{\Gamma_M(\frac{T+d}{2})}{\Gamma_M(\frac{d}{2})} |\Psi|^{-\frac{T}{2}} |D_\Omega^T X^T X D_\Omega + I_K|^{-\frac{M}{2}} \\
& * \left| I_M + E_\Psi^T \left[\hat{\epsilon}^T \hat{\epsilon} + (\hat{B} - \hat{b})^T \Omega^{-1} (\hat{B} - \hat{b}) \right] E_\Psi \right|^{-\frac{T+d}{2}} \\
& \propto |\lambda^2 D_\Phi^T X^T X D_\Phi + I_K|^{-\frac{M}{2}} \\
& * \left| \lambda^2 I_M + E_\Psi^T \left[\lambda^2 \hat{\epsilon}^T \hat{\epsilon} + (\hat{B} - \hat{b})^T \Phi^{-1} (\hat{B} - \hat{b}) \right] E_\Psi \right|^{-\frac{T+d}{2}} \lambda^{M(T+d)} \\
& \hat{\epsilon}^T \hat{\epsilon} = (Y - X\hat{B})^T (Y - X\hat{B}) \\
& \hat{B} = (\lambda^2 X^T X + \Phi^{-1})^{-1} (\lambda^2 X^T Y + \Phi^{-1} \hat{b})
\end{aligned} \tag{10}$$

We can use gradient method to optimize 10. First we take log and then take derivative w.r.t. λ . However, the code to implement this would be much more complex. Here, I just use grid-search first. The MSFE depends on the interval we specify to do grid search.

MSFE	log-real GDP	log-real GDP	log-GDP deflator	log-GDP deflator
Average growth rate	1 quarter	4 quarters	1 quarter	4 quarters
[0.0001, 1]	$9.524e - 4$	$1.748e - 4$	$5.020e - 4$	$2.014e - 4$
[0.05, 1]	$9.063e - 4$	$1.604e - 4$	$4.995e - 4$	$1.934e - 4$

Table 3: Mean squared forecast error for moving window and extending window.

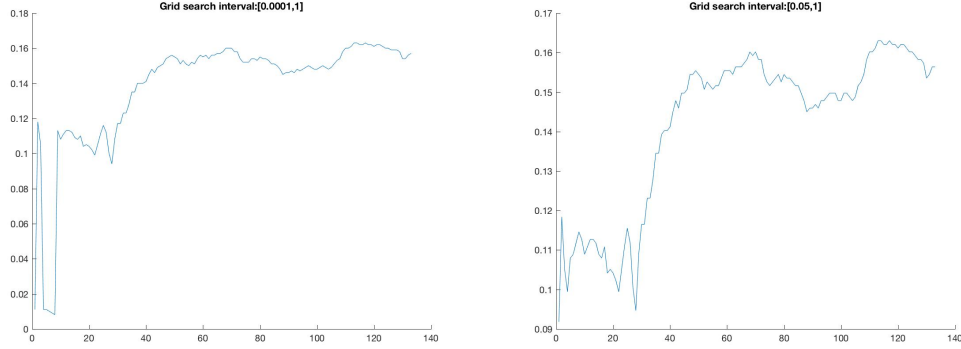


Figure 1: The optimal λ with extending windows and two different search intervals. I am not sure why when λ is near 0 the path has a big jump. From MSFE, we see they are comparable with those obtained by setting $\lambda = 0.2$.

2 Prob 2:

- (1) First use the whole data set to get MLE for parameters by maximizing likelihood (conditional likelihood conditioning on the first p observations). Then, use the obtained coefficients to do projection from $p + 1$ to T , based on the given initial observations and later projected values.

Name	Observed data	Flat prior	Minnesota prior	SOC (1)	SOC (5)
GDP	3.35E+00	3.26E+00	3.23E+00	3.27E+00	3.33E+00
pri-infla	5.65E-04	1.38E-04	1.38E-04	6.88E-05	3.13E-05
federal	1.07E-03	2.71E-04	2.16E-04	1.55E-04	6.32E-05
wage-infla	1.02E-03	1.44E-04	1.24E-04	8.18E-05	3.31E-05
labor	5.96E-02	6.23E-02	5.53E-02	7.04E-02	6.76E-02
cons	2.27E-02	2.20E-02	2.13E-02	1.93E-02	1.74E-02

As shown in above form, the variance for log-real GDP, log-labor share, and log-consumption ratio are mostly explained by the deterministic components, but for quarterly price inflation, federal funds rate, and quarterly wage inflation, most of variance cannot be explained by the deterministic components. I think it makes sense because the former three variables are more stable, while the later three variables can be easily affected by the time and current economical situation.

- (2) Repeat the same exercise for Minnesota priors.

The difference is very small. They are comparable.

- (3) Repeat the same exercise for SOC priors.

The plot is following:

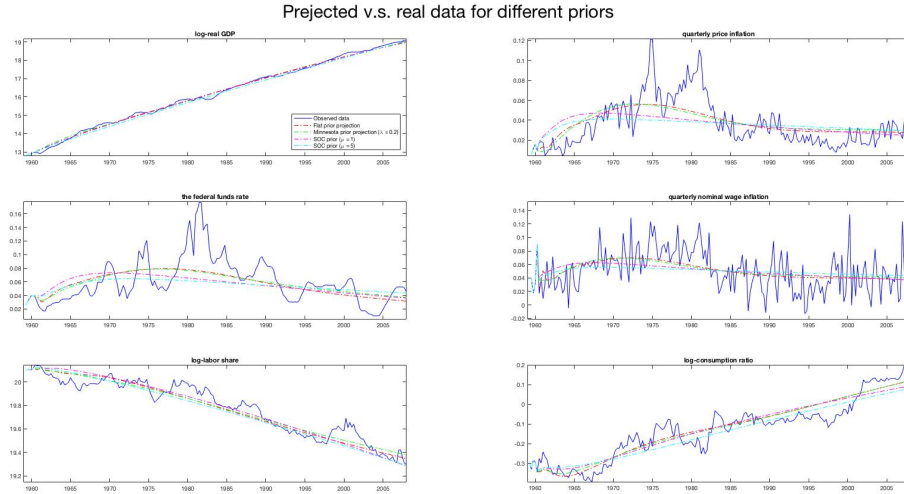


Figure 2: The projections based on the whole data set with flat prior, Minnesota prior, SOC with $\mu = 1$, and SOC with $\mu = 5$.

Different priors give very similar results.

```

1 clear();
2 % Load data into the working space
3 load('dataVARmedium','-mat');
4 p = 5;
5 T = 64-5;
6 N = size(y,1);
7 M = 7;
8 K = M*p+1;
9 h = 4;
10
11 %% (1) OLS solution for prediction
12 %% Moving window prediction
13 tic
14 z_comp = zeros(N-(T+p+h)+1,8); % the first four columns are predicted z (
    average growth rate of GDP) and the second four columns is the real z
15 for i = 1:(N-(T+p+h)+1)
16     % design matrix
17     X = zeros(T,K); % a T*K matrix
18     for j = 1:T
19         X(j,:) = [1, reshape(y((i+j+p-2:-1:i+j-1),:),' ,1,K-1)]; % have to
    transpose because the reshape function operate in column
20     end
21     % OLS solution for coefficients
22     B_hat = (X'*X)\X'*y((i+p:i+p+T-1),:); % a K*M matrix
23     % for h = 1 & 4
24     pred_y = zeros(h,M); % a h*M matrix
25     y_lag = [1, reshape(y(i+p+T-1:-1:i+T,:),' ,1,K-1)]; % lag of y for the
    prediction
26     for k = 1:h
27         % forecast
28         pred_y(k,:) = y_lag*B_hat;
29         if k<h
30             % reform the lag for y of prediction
31             y_lag = [1, reshape(pred_y(k:-1:1,:),' ,1,k*M), reshape(y(i+p+T-1:-1:
    i+T+k,:),' ,1,K-1-k*M)];
32         end
33     end
34     z_comp(i,1:2) = pred_y(1,1:2)-y(i+p+T-1,1:2); %predicted average growth
    rates for one quarter of (i) log-real GDP and (ii) log-GDP delector
35     z_comp(i,3:4) = (pred_y(h,1:2)-y(i+p+T-1,1:2))/h; %predicted average
    growth rates for h quarters of (i) log-real GDP and (ii) log-GDP delector
36     z_comp(i,5:6) = y(i+p+T,1:2)-y(i+p+T-1,1:2); %average growth rates for one
    quarter of (i) log-real GDP and (ii) log-GDP delector
37     z_comp(i,7:8) = (y(i+p+T+h-1,1:2)-y(i+p+T-1,1:2))/h; %average growth rates
    for h quarters of (i) log-real GDP and (ii) log-GDP delector
38 end
39 % Calculate the average squared forecast error
40 MSFE1 = sum((z_comp(:,1:4)-z_comp(:,5:8)).^2,1)/size(z_comp,1)
41 csvwrite('mse-ex1-1-moving.csv',MSFE1)

```

```

42 toc
43 %% Expanding window prediction
44 tic
45 z_comp = zeros(N-(T+p+h)+1,8); % the first four columns are predicted z (
    average growth rate of GDP) and the second four columns is the real z
46 for i = 1:(N-(T+p+h)+1)
47     if i ==1
48         % design matrix
49         X = zeros(T,K); % a T*K matrix
50         for j = 1:T
51             X(j,:) = [1, reshape(y((j+p-1:-1:j),:)',1,K-1)]; % have to
    transpose because the reshape function operate in column
52         end
53     else
54         new_row = [1, reshape(y((i+T+p-2:-1:i+T-1),:)',1,K-1)];
55         X = [X; new_row]; % a (T+i-1)*K matrix
56     end
57     % OLS solution for coefficients
58     B_hat = (X'*X)\X'*y((p+1:p+T+i-1),:); % a K*M matrix
59     % for h = 1 & 4
60     pred_y = zeros(h,M); % a h*M matrix
61     y_lag = [1, reshape(y(T+i+p-1:-1:T+i,:),1,K-1)]; % lag of y for the
    prediction
62     for k = 1:h
63         % forecast
64         pred_y(k,:) = y_lag*B_hat;
65         if k < h
66             % reform the lag for y of prediction
67             y_lag = [1, reshape(pred_y(k:-1:1,:),1,k*M), reshape(y(T+i+p-1:-1:
    T+i+k,:),1,K-1-k*M)];
68         end
69     end
70     z_comp(i,1:2) = pred_y(1,1:2)-y(i+p+T-1,1:2); % predicted average growth
    rates for one quarter of (i) log-real GDP and (ii) log-GDP delector
71     z_comp(i,3:4) = (pred_y(h,1:2)-y(i+p+T-1,1:2))/h; % predicted average
    growth rates for h quarters of (i) log-real GDP and (ii) log-GDP delector
72     z_comp(i,5:6) = y(i+p+T,1:2)-y(i+p+T-1,1:2); % average growth rates for one
    quarter of (i) log-real GDP and (ii) log-GDP delector
73     z_comp(i,7:8) = (y(i+p+T+h-1,1:2)-y(i+p+T-1,1:2))/h; % average growth rates
    for h quarters of (i) log-real GDP and (ii) log-GDP delector
74 end
75 % Calculate the average squared forecast error
76 MSFE2 = sum((z_comp(:,1:4)-z_comp(:,5:8)).^2,1)/size(z_comp,1)
77 csvwrite('mse_ex1-1-extending.csv',MSFE2)
78 toc
79 %% (2) Minnesota prior with a given lambda
80 %% Expanding window prediction
81 tic
82 z_comp = zeros(N-(T+p+h)+1,8); % the first four columns are predicted z (
    average growth rate of GDP) and the second four columns is the real z

```

```

83 % Given lambda
84 lam = 0.2;
85 % Estimate b
86 b_hat = [zeros(M,1), eye(M,M*p)];
87 for i = 1:(N-(T+p+h)+1)
88     if i == 1
89         % design matrix
90         X = zeros(T,K); % a T*K matrix
91         for j = 1:T
92             X(j,:) = [1, reshape(y((j+p-1:-1:j),:), 1, K-1)]; % have to
transpose because the reshape function operate in column
93         end
94     else
95         new_row = [1, reshape(y((i+T+p-2:-1:i+T-1),:), 1, K-1)];
96         X = [X; new_row]; % a (T+i-1)*K matrix
97     end
98 % Estimate elements of Omega
99 % The first observation is given (regarded as index 0) and the rest of
observation up to the
100 % end of current window (index 1 to T) are for prediction as stated in
the problem
101 % statement
102 % Follow Hamilton 1994 to get the MLE for variance
103 sum1 = sum(y(1:p+T+i-2,:), 1); %sum_1^T y_-(t-1)
104 sum2 = sum(y(2:p+T+i-1,:), 1); %sum_1^T y_-(t)
105 sum3 = sum(y(1:p+T+i-2,:).^2, 1); %sum_1^T y_-(t-1)^2
106 sum4 = sum(y(2:p+T+i-1,:).^2, 1); %sum_1^T y_-(t)^2
107 sum5 = sum(y(2:p+T+i-1,:) .* y(1:p+T+i-2,:), 1); %sum_1^T y_-(t) y_-(t-1)
108 sigma2_hat = zeros(1,M);
109 for m = 1:M
110     c_phi_hat = [p+T+i-2, sum1(m); sum1(m), sum3(m)] \ [sum2(m); sum5(m)]; %
c_phi_hat = [c_hat, phi_hat]
111     sigma2_hat(m) = 1.0 / (p+T+i-2) * (sum4(m) + c_phi_hat(2)^2 * sum3(m) + 2*
c_phi_hat(1) * c_phi_hat(2) * sum1(m) ...
112         - 2 * c_phi_hat(1) * sum2(m) - 2 * c_phi_hat(2) * sum5(m)) +
c_phi_hat(1)^2;
113     end
114     omega_d = [10^6];
115     for j = 1:p
116         omega_d = [omega_d, (sigma2_hat * j^2).^(-1)];
117     end
118 % Times the lambda^2
119 omega_d = lam^2 * omega_d;
120 % MAP solution for coefficients
121 B_hat = (X' * X + diag(omega_d.^( -1))) \ (X' * y((p+1:p+T+i-1),:) + diag(omega_d.^( -1)) * b_hat'); % a K*M matrix
122 % for h = 1 & 4
123 pred_y = zeros(h,M); % a h*M matrix
124 y_lag = [1, reshape(y(T+i+p-1:-1:T+i,:), 1, K-1)]; % lag of y for the
prediction

```



```

125     for k = 1:h
126         % forecast
127         pred_y(k,:) = y_lag*B_hat;
128         if k<h
129             % reform the lag for y of prediction
130             y_lag = [1,reshape(pred_y(k:-1:1,:),1,k*M),reshape(y(T+i+p-1:-1:
T+i+k,:),1,K-1-k*M)];
131         end
132     end
133     z_comp(i,1:2) = pred_y(1,1:2)-y(i+p+T-1,1:2);%predicted average growth
rates for one quarter of (i) log-real GDP and (ii) log-GDP delector
134     z_comp(i,3:4) = (pred_y(h,1:2)-y(i+p+T-1,1:2))/h;%predicted average
growth rates for h quarters of (i) log-real GDP and (ii) log-GDP delector
135     z_comp(i,5:6) = y(i+p+T,1:2)-y(i+p+T-1,1:2);%average growth rates for one
quarter of (i) log-real GDP and (ii) log-GDP delector
136     z_comp(i,7:8) = (y(i+p+T+h-1,1:2)-y(i+p+T-1,1:2))/h;%average growth rates
for h quarters of (i) log-real GDP and (ii) log-GDP delector
137 end
138 % Calculate the average squared forecast error
139 MSFE3 = sum((z_comp(:,1:4)-z_comp(:,5:8)).^2,1)/size(z_comp,1)
140 csvwrite('mse_ex1-1-lambda.given.csv',MSFE3)
141 toc
142 %% (3)Minnesota prior without a given lambda (optimize marginal likelihood to
get lambda each time)
143 %% Expanding window prediction
144 tic
145 %Cannot set lambda = 0, because when take log of lambda, 0 is not
146 %meaningful
147 lam_bd = [0.0001,1];
148 N_try = 1000;
149 d = M + 2;
150 z_comp = zeros(N-(T+p+h)+1,8); % the first four columns are predicted z (
average growth rate of GDP) and the second four columns is the real z
151 % Record optimal value of lambda
152 rec_lam_opt = zeros(N-(T+p+h)+1,1);
153 % Estimate b
154 b_hat = [zeros(M,1),eye(M,M*p)];
155 for i = 1:(N-(T+p+h)+1)
156     if i ==1
157         % design matrix
158         X = zeros(T,K);% a T*K matrix
159         for j = 1:T
160             X(j,:) = [1,reshape(y((j+p-1:-1:j),:),1,K-1)];% have to
transpose because the reshape function operate in column
161         end
162     else
163         new_row = [1,reshape(y((i+T+p-2:-1:i+T-1),:),1,K-1)];
164         X = [X;new_row]; % a (T+i-1)*K matrix
165     end
166     % Estimate elements of Omega

```

```

167 % The first observation is given and the rest of observation up to the
168 % end of current window are for prediction as stated in the problem
169 % statement
170 % Follow Hamilton 1994 to get the MLE for variance
171 sum1 = sum(y(1:p+T+i-2,:),1);%sum_1^T y_-(t-1)
172 sum2 = sum(y(2:p+T+i-1,:),1);%sum_1^T y_-(t)
173 sum3 = sum(y(1:p+T+i-2,:).^2,1);%sum_1^T y_-(t-1)^2
174 sum4 = sum(y(2:p+T+i-1,:).^2,1);%sum_1^T y_-(t)^2
175 sum5 = sum(y(2:p+T+i-1,:).*y(1:p+T+i-2,:),1);%sum_1^T y_-(t)y_-(t-1)
176 sigma2_hat = zeros(1,M);
177 for m = 1:M
178     c_phi_hat = [p+T+i-2,sum1(m);sum1(m),sum3(m)]\ [sum2(m);sum5(m)];%
179     c_phi_hat = [c_hat,phi_hat]
180     temp_sum = 0;
181     sigma2_hat(m) = 1.0/(p+T+i-2)*(sum4(m)+c_phi_hat(2)^2*sum3(m)+2*
182     c_phi_hat(1)*c_phi_hat(2)*sum1(m)...
183     -2*c_phi_hat(1)*sum2(m)-2*c_phi_hat(2)*sum5(m))+
184     c_phi_hat(1)^2;
185 end
186 omega_d = [10^6];
187 for j = 1:p
188     omega_d = [omega_d,(sigma2_hat*j^2).^(-1)];
189 end
190 % Optimize in terms of lambda
191 % Use grid-search method
192 %options_optim = optimoptions(@fminunc,'Display','iter-detailed','
193 Algorithm','quasi-newton','SpecifyObjectiveGradient',true,'MaxIterations
194 ',1000);
195 rec_lam_opt(i) = lam_mle_grid(@loglikelih,lam_bd,N_try,X,y((p+1:p+T+i-1)
196 :,),sigma2_hat,omega_d,b_hat',M,T+i-1,d);
197 % Times the lambda^2
198 omega_d = rec_lam_opt(i)^2 * omega_d;
199 % MAP solution for coefficients
200 B_hat = (X'*X+diag(omega_d.^( -1)))\ (X'*y((p+1:p+T+i-1),:)+diag(omega_d
201 .^( -1))*b_hat');% a K*M matrix
202 % for h = 1 & 4
203 pred_y = zeros(h,M);% a h*M matrix
204 y_lag = [1,reshape(y(T+i+p-1:-1:T+i,:),1,K-1)];% lag of y for the
205 prediction
206 for k = 1:h
207     % forecast
208     pred_y(k,:) = y_lag*B_hat;
209     if k<h
210         % reform the lag for y of prediction
211         y_lag = [1,reshape(pred_y(k:-1:1,:),1,k*M),reshape(y(T+i+p-1:-1:
212 T+i+k,:),1,K-1-k*M)];
213     end
214 end
215 z_comp(i,1:2) = pred_y(1,1:2)-y(i+p+T-1,1:2);%predicted average growth
216 rates for one quarter of (i) log-real GDP and (ii) log-GDP delecor

```

```

207     z_comp(i,3:4) = (pred_y(h,1:2)-y(i+p+T-1,1:2))/h;%predicted average
growth rates for h quarters of (i) log-real GDP and (ii) log-GDP delector
208     z_comp(i,5:6) = y(i+p+T,1:2)-y(i+p+T-1,1:2);%average growth rates for one
quarter of (i) log-real GDP and (ii) log-GDP delector
209     z_comp(i,7:8) = (y(i+p+T+h-1,1:2)-y(i+p+T-1,1:2))/h;%average growth rates
for h quarters of (i) log-real GDP and (ii) log-GDP delector
210 end
211 % Calculate the average squared forecast error
212 MSFE4 = sum((z_comp(:,1:4)-z_comp(:,5:8)).^2,1)/size(z_comp,1)
213 csvwrite('mse_ex1-1-lambda_opt.csv',MSFE4)
214 csvwrite('lambda_opt.csv',rec_lam_opt)
215 figure();
216 hold on;
217 plot(rec_lam_opt);
218 int_str = [' ',num2str(lam_bd(1))',' ',num2str(lam_bd(2))',' '];
219 title(['Grid search interval:',int_str]);
220 cwd = '/Users/kungangzhang/Documents/OneDrive/Northwestern/Study/Courses/ECON
-482/HW1/';
221 saveas(gca,[cwd,['lambda_opt_1']], 'fig');
222 saveas(gca,[cwd,['lambda_opt_1']], 'jpg');
223 toc

```

Listing 1: Matlab code for problem 1.

```

1 clear();
2 % Load data into the working space
3 load('dataVARmedium','-mat');
4
5 lgGDP = y(:,1);%log-real GDP
6 qpriinf = diff(y(:,2));%quarterly price inflation
7 fedfr = y(:,3);%federal funds rate
8 qnomwainf = diff(y(:,7)+y(:,2));%quarterly nominal wage inflation
9 lglabsh = y(:,7)+y(:,6)-y(:,1);%log-labor share
10 lgcomra = y(:,4)-y(:,1);%log-consumption ratio
11
12 p = 5;
13 N = size(y,1);
14 T = N-p;%Initial time horizon for forecasting
15 M = 7;
16 K = M*p+1;
17 h = 0;%The maximum horizon of out-of-sample forecast
18
19 %% (1) OLS solution for prediction
20 tic
21 for i = 1:(N-(T+p+h)+1)
22     if i ==1
23         % design matrix
24         X = zeros(T,K);% a T*K matrix
25         for j = 1:T
26             X(j,:) = [1,reshape(y((j+p-1:-1:j)),',1,K-1)];% have to
transpose because the reshape function operate in column

```

```

27     end
28 else
29     new_row = [1, reshape(y((i+T+p-2:-1:i+T-1),:), 1, K-1)];
30     X = [X; new_row]; % a (T+i-1)*K matrix
31 end
32 % OLS solution for coefficients
33 B_hat = (X'*X)\X'*y((p+1:p+T+i-1),:); % a K*M matrix
34 % Do projection based on MLE of coefficients
35 yproj1 = y(1:p,:);
36 for i = p+1:N
37     yproj1(i,:) = [1, reshape(yproj1(i-1:-1:i-p,:), 1, K-1)]*B_hat;
38 end
39 lgGDP1 = yproj1(:,1);
40 qpriinf1 = diff(yproj1(:,2));
41 fedfr1 = yproj1(:,3);
42 qnomwainf1 = diff(yproj1(:,7)+yproj1(:,2));
43 lglabsh1 = yproj1(:,7)+yproj1(:,6)-yproj1(:,1);
44 lgcomra1 = yproj1(:,4)-yproj1(:,1);
45 end
46 xt = ((1959+0.25): 0.25: (2008+1))';
47 figure();
48 hold on;
49 box on;
50 %title('Projected v.s. real data');
51
52 annotation('textbox', [0 0.9 1 0.1], ...
53     'String', 'Projected v.s. real data for different priors', ...
54     'EdgeColor', 'none', ...
55     'HorizontalAlignment', 'center')
56
57 legendinfo{1} = ['Observed data'];
58 handle(1) = plot_all([xt, lgGDP1], [xt(2:end), qpriinf1], [xt, fedfr1], [xt(2:end),
59     qnomwainf1], [xt, lglabsh1], [xt, lgcomra1], '-b', 'Observed data');
59 xt1 = xt(1:end);
60 legendinfo{2} = ['Flat prior projection'];
61 handle(2) = plot_all([xt1, lgGDP1], [xt1(2:end), qpriinf1], [xt1, fedfr1], [xt1(2:
62     end), qnomwainf1], [xt1, lglabsh1], [xt1, lgcomra1], '-r', 'Flat prior
63     projection');
62 toc
63
64 %% (2) Minnesota prior with a given lambda
65 tic
66 % Given lambda
67 lam = 0.2;
68 % Estimate b
69 b_hat = [zeros(M,1), eye(M,M*p)];
70 for i = 1:(N-(T+p+h)+1)
71     if i == 1
72         % design matrix
73         X = zeros(T,K); % a T*K matrix

```

```

74         for j = 1:T
75             X(j,:) = [1, reshape(y((j+p-1:-1:j),:)',1,K-1)]; % have to
transpose because the reshape function operate in column
76         end
77     else
78         new_row = [1, reshape(y((i+T+p-2:-1:i+T-1),:)',1,K-1)];
79         X = [X; new_row]; % a (T+i-1)*K matrix
80     end
81     % Estimate elements of Omega
82     % The first observation is given (regarded as index 0) and the rest of
83     % observation up to the
84     % end of current window (index 1 to T) are for prediction as stated in
85     % the problem
86     % statement
87     % Follow Hamilton 1994 to get the MLE for variance
88     sum1 = sum(y(1:p+T+i-2,:),1); %sum_1^T y_-(t-1)
89     sum2 = sum(y(2:p+T+i-1,:),1); %sum_1^T y_-(t)
90     sum3 = sum(y(1:p+T+i-2,:).^2,1); %sum_1^T y_-(t-1)^2
91     sum4 = sum(y(2:p+T+i-1,:).^2,1); %sum_1^T y_-(t)^2
92     sum5 = sum(y(2:p+T+i-1,:).*y(1:p+T+i-2,:),1); %sum_1^T y_-(t)y_-(t-1)
93     sigma2_hat = zeros(1,M);
94     for m = 1:M
95         c_phi_hat = [p+T+i-2,sum1(m);sum1(m),sum3(m)] \ [sum2(m);sum5(m)]; %
96         c_phi_hat = [c_hat, phi_hat]
97         sigma2_hat(m) = 1.0/(p+T+i-2)*(sum4(m)+c_phi_hat(2)^2*sum3(m)+2*
98         c_phi_hat(1)*c_phi_hat(2)*sum1(m) ...
99         -2*c_phi_hat(1)*sum2(m)-2*c_phi_hat(2)*sum5(m))+
100         c_phi_hat(1)^2;
101     end
102     omega_d = [10^6];
103     for j = 1:p
104         omega_d = [omega_d, (sigma2_hat*j^2).^(-1)];
105     end
106     % Times the lambda^2
107     omega_d = lam^2 * omega_d;
108     % MAP solution for coefficients
109     B_hat = (X'*X+diag(omega_d.^( -1))) \ (X'*y((p+1:p+T+i-1),:)+diag(omega_d
110     .^( -1))*b_hat'); % a K*M matrix
111     % Do projection based on MLE of coefficients
112     yproj2 = y(1:p,:);
113     for i = p+1:N
114         yproj2(i,:) = [1, reshape(yproj2(i-1:-1:i-p,:)',1,K-1)]*B_hat;
115     end
116     lgGDP2 = yproj2(:,1);
117     qpriinf2 = diff(yproj2(:,2));
118     fedfr2 = yproj2(:,3);
119     qnomwainf2 = diff(yproj2(:,7)+yproj2(:,2));
120     lglabsh2 = yproj2(:,7)+yproj2(:,6)-yproj2(:,1);
121     lgcomra2 = yproj2(:,4)-yproj2(:,1);

```

```

117 xt2 = xt(1:end);
118 legendinfo{3} = ['Minnesota prior projection (\lambda = 0.2)'];
119 handle(3) = plot_all([xt2,lgGDP2],[xt2(2:end),qpriinf2],[xt2,fedfr2],[xt2(2:
    end),qnomwainf2],[xt2,lglabsh2],[xt2,lgcomra2],'-g','Minnesota prior
    projection (\lambda = 0.2)');
120
121 toc
122 %% (3)Minnesota prior without a given lambda (optimize marginal likelihood to
    get lambda each time)
123 tic
124 % Given lambda
125 lam = 0.2;
126 mu = 1;
127 % Estimate b
128 b_hat = [zeros(M,1),eye(M,M*p)];
129 for i = 1:(N-(T+p+h)+1)
130     if i == 1
131         % design matrix
132         X = zeros(T,K); % a T*K matrix
133         for j = 1:T
134             X(j,:) = [1,reshape(y((j+p-1:-1:j),:)',1,K-1)]; % have to
                transpose because the reshape function operate in column
135         end
136     else
137         new_row = [1,reshape(y((i+T+p-2:-1:i+T-1),:)',1,K-1)];
138         X = [X;new_row]; % a (T+i-1)*K matrix
139     end
140     % Estimate elements of Omega
141     % The first observation is given (regarded as index 0) and the rest of
    observation up to the
142     % end of current window (index 1 to T) are for prediction as stated in
    the problem
143     % statement
144     % Follow Hamilton 1994 to get the MLE for variance
145     sum1 = sum(y(1:p+T+i-2,:),1); %sum_1^T y_-(t-1)
146     sum2 = sum(y(2:p+T+i-1,:),1); %sum_1^T y_-(t)
147     sum3 = sum(y(1:p+T+i-2,:).^2,1); %sum_1^T y_-(t-1)^2
148     sum4 = sum(y(2:p+T+i-1,:).^2,1); %sum_1^T y_-(t)^2
149     sum5 = sum(y(2:p+T+i-1,:).*y(1:p+T+i-2,:),1); %sum_1^T y_-(t)y_-(t-1)
150     sigma2_hat = zeros(1,M);
151     for m = 1:M
152         c_phi_hat = [p+T+i-2,sum1(m);sum1(m),sum3(m)] \ [sum2(m);sum5(m)]; %
        c_phi_hat = [c_hat,phi_hat]
153         sigma2_hat(m) = 1.0/(p+T+i-2)*(sum4(m)+c_phi_hat(2)^2*sum3(m)+2*
        c_phi_hat(1)*c_phi_hat(2)*sum1(m) ...
154             -2*c_phi_hat(1)*sum2(m)-2*c_phi_hat(2)*sum5(m))+
        c_phi_hat(1)^2;
155     end
156     omega_d = [10^6];
157     for j = 1:p

```

```

158     omega_d = [omega_d, (sigma2_hat*j^2).^(-1)];
159 end
160 % Times the lambda^2
161 omega_d = lam^2 * omega_d;
162
163 %stack the artificial observations at the bottom to get soc prior
164 Xplus = [zeros(M,1), repmat(diag(mu*mean(y(1:p,:),1)), [1 p])];
165 X= [X;Xplus]; % stack x at the bottom
166 yplus = diag(mu*mean(y(1:p,:),1));
167 ynew = [y((p+1:p+T+i-1),:); yplus];
168
169 % MAP solution for coefficients
170 B_hat = (X'*X+diag(omega_d.^(-1))) \ (X'*ynew+diag(omega_d.^(-1))*b_hat '); %
    a K*M matrix
171 % Do projection based on MLE of coefficients
172 yproj3 = y(1:p,:);
173 for j = p+1:N
174     yproj3(j,:) = [1, reshape(yproj3(j-1:-1:j-p,:), '1, K-1)]*B_hat;
175 end
176 lgGDP3 = yproj3(:,1);
177 qpriinf3 = diff(yproj3(:,2));
178 fedfr3 = yproj3(:,3);
179 qnomwainf3 = diff(yproj3(:,7)+yproj3(:,2));
180 lglabsh3 = yproj3(:,7)+yproj3(:,6)-yproj3(:,1);
181 lgcomra3 = yproj3(:,4)-yproj3(:,1);
182 end
183 xt3 = xt(1:end);
184 legendinfo{4} = ['SOC prior (\mu = 1)'];
185 handle(4) = plot_all([xt3, lgGDP3], [xt3(2:end), qpriinf3], [xt3, fedfr3], [xt3(2:
    end), qnomwainf3], [xt3, lglabsh3], [xt3, lgcomra3], '-m', 'SOC prior (\mu = 1)
    ');
186
187 toc
188
189 tic
190 % Given lambda
191 lam = 0.2;
192 mu = 5;
193 % Estimate b
194 b_hat = [zeros(M,1), eye(M,M*p)];
195 for i = 1:(N-(T+p+h)+1)
196     if i ==1
197         % design matrix
198         X = zeros(T,K); % a T*K matrix
199         for j = 1:T
200             X(j,:) = [1, reshape(y((j+p-1:-1:j),:), '1, K-1)]; % have to
                transpose because the reshape function operate in column
201         end
202     else
203         new_row = [1, reshape(y((i+T+p-2:-1:i+T-1),:), '1, K-1)];

```

```

204     X = [X;new_row]; % a (T+i-1)*K matrix
205     end
206     % Estimate elements of Omega
207     % The first observation is given (regarded as index 0) and the rest of
208     observation up to the
209     % end of current window (index 1 to T) are for prediction as stated in
210     the problem
211     % statement
212     % Follow Hamilton 1994 to get the MLE for variance
213     sum1 = sum(y(1:p+T+i-2,:),1);%sum_1^T y_-(t-1)
214     sum2 = sum(y(2:p+T+i-1,:),1);%sum_1^T y_-(t)
215     sum3 = sum(y(1:p+T+i-2,).^2,1);%sum_1^T y_-(t-1)^2
216     sum4 = sum(y(2:p+T+i-1,).^2,1);%sum_1^T y_-(t)^2
217     sum5 = sum(y(2:p+T+i-1,:).*y(1:p+T+i-2,:),1);%sum_1^T y_-(t)y_-(t-1)
218     sigma2_hat = zeros(1,M);
219     for m = 1:M
220         c_phi_hat = [p+T+i-2,sum1(m);sum1(m),sum3(m)]\ [sum2(m);sum5(m)];%
221         c_phi_hat = [c_hat,phi_hat]
222         sigma2_hat(m) = 1.0/(p+T+i-2)*(sum4(m)+c_phi_hat(2)^2*sum3(m)+2*
223         c_phi_hat(1)*c_phi_hat(2)*sum1(m)...
224         -2*c_phi_hat(1)*sum2(m)-2*c_phi_hat(2)*sum5(m))+
225         c_phi_hat(1)^2;
226     end
227     omega_d = [10^6];
228     for j = 1:p
229         omega_d = [omega_d,(sigma2_hat*j^2).^(-1)];
230     end
231     % Times the lambda^2
232     omega_d = lam^2 * omega_d;
233
234     %stack the artificial observations at the bottom to get soc prior
235     Xplus = [zeros(M,1), repmat(diag(mu*mean(y(1:p,:),1)),[1 p])];
236     X= [X;Xplus]; % stack x at the bottom
237     yplus = diag(mu*mean(y(1:p,:),1));
238     ynew = [y((p+1:p+T+i-1),:);yplus];
239
240     % MAP solution for coefficients
241     B_hat = (X'*X+diag(omega_d.^(-1)))\'(X'*ynew+diag(omega_d.^(-1))*b_hat');%
242     a K*M matrix
243     % Do projection based on MLE of coefficients
244     yproj4 = y(1:p,:);
245     for j = p+1:N
246         yproj4(j,:) = [1,reshape(yproj4(j-1:-1:j-p,:),1,K-1)]*B_hat;
247     end
248     lgGDP4 = yproj4(:,1);
249     qpriinf4 = diff(yproj4(:,2));
250     fedfr4 = yproj4(:,3);
251     qnomwainf4 = diff(yproj4(:,7)+yproj4(:,2));
252     lglabsh4 = yproj4(:,7)+yproj4(:,6)-yproj4(:,1);
253     lgcomra4 = yproj4(:,4)-yproj4(:,1);

```



```

248 end
249 xt4 = xt(1:end);
250 legendinfo{5} = ['SOC prior (\mu = 5)'];
251 handle(5) = plot_all([xt4,lgGDP4],[xt4(2:end),qpriinf4],[xt4,fedfr4],[xt4(2:
    end),qnomwainf4],[xt4,lglabsh4],[xt4,lgcomra4],'-c','SOC prior (\mu = 5)
    ');
252
253 toc
254
255 var1=[var(lgGDP),var(lgGDP1),var(lgGDP2),var(lgGDP3),var(lgGDP4)];
256 var2=[var(qpriinf),var(qpriinf1),var(qpriinf2),var(qpriinf3),var(qpriinf4)];
257 var3=[var(fedfr),var(fedfr1),var(fedfr2),var(fedfr3),var(fedfr4)];
258 var4=[var(qnomwainf),var(qnomwainf1),var(qnomwainf2),var(qnomwainf3),var(
    qnomwainf4)];
259 var5=[var(lglabsh),var(lglabsh1),var(lglabsh2),var(lglabsh3),var(lglabsh4)];
260 var6=[var(lgcomra),var(lgcomra1),var(lgcomra2),var(lgcomra3),var(lgcomra4)];
261 var_all = [var1;var2;var3;var4;var5;var6];
262 legend(handle,legendinfo,'Location','southeast');

```

Listing 2: Matlab code for problem 2.

```

1 function val = logmlikelih(lam,X,Y,d_psi,d_phi,b_hat,M,T,d)
2 % Calculate the log marginal likelihood
3 % Input:
4 % lam: lambda value which we are optimizing in terms of
5 % X: matrix X in time series (lag);
6 % Y: matrix Y which are to be forecasted;
7 % d_psi: diagnoal elements of Psi matrix (which is diagonal);
8 % d_ome: diagonal elements of Omega matrix (which is diagonal);
9 % M: the dimension of y
10 % T: time horizon to be forecasted
11 % d: some constant specified in prior
12 % Output:
13 % val: the log marginal likelihood up to adding a constant
14 tempm1 = X*diag(d_phi.^0.5);
15 eig1 = eig(tempm1'*tempm1);
16 B_hat = (lam^2*(X'*X)+diag(d_phi.^(-1)))/(lam^2*X'*Y + diag(d_phi.^(-1))*
    b_hat);
17 eps_hat = Y - X*B_hat;
18 tempm2 = B_hat-b_hat;
19 eig2 = eig(diag(d_psi.^(-0.5))*(lam^2*(eps_hat'*eps_hat)+tempm2'*diag(d_phi
    .^(-1))*tempm2)*diag(d_psi.^(-0.5)));
20 val = (-M/2)*sum(log(1+lam^2*eig1))+(-(T+d)/2)*sum(log(lam^2+eig2))+M*(T+d)*
    log(lam);
21 end

```

Listing 3: Matlab code for evaluate the log marginal likelihood.

```

1 function lam_opt = lam_mle_grid(objFunc,lam_bd,N_try,X,Y,d_psi,d_phi,b_hat,M,
    T,d)

```

```

2 % Use grid search to find the optimal solution for lam with several starting
   points
3 % Input:
4 % objFunc: objective function which return kernel of marginal likelihood
5 % (or log of it)
6 %=====
7     % Calculate the marginal likelihood
8     % Input:
9     % lam: lambda value which we are optimizing in terms of
10    % X: matrix X in time series (lag);
11    % Y: matrix Y which are to be forecasted;
12    % d_psi: diagnoal elements of Psi matrix (which is diagonal);
13    % d_ome: diagonal elements of Omega matrix (which is diagonal);
14    % M: the dimension of y
15    % T: time horizon to be forecasted
16    % d: some constant specified in prior
17    % Output:
18    % val: the marginal likelihood up to a constant factor
19 %=====
20 % lam_bd: lambda value upper and lower bound for initial value exploring
21 %=====
22     % [lam_lower_bound,lam_upper_bound]
23 %=====
24 % N_try: how many initial tries to do inside the lam_bd
25 % X: matrix X in time series (lag);
26 % Y: matrix Y which are to be forecasted;
27 % d_psi: diagnoal elements of Psi matrix (which is diagonal);
28 % d_ome: diagonal elements of Omega matrix (which is diagonal);
29 % M: the dimension of y
30 % T: time horizon to be forecasted
31 % d: some constant specified in prior
32 % options_optim: options when calling the fminunc function
33 % Output:
34 % opt_lam: optimal lambda that maximizing the marginal likelihood
35
36 %% code
37 %options_optim = optimoptions(@fminunc,'Display','iter-detailed','Algorithm
   ','trust-region','SpecifyObjectiveGradient',true);
38
39 %fun = @(lam)objFunc(lam,X,Y,d_psi,d_phi,b_hat,M,T,d);
40
41
42 lam_try = lam_bd(1):(lam_bd(2)-lam_bd(1))/N_try:lam_bd(2);
43 rec_lam = zeros(length(lam_try),2);
44 for i = 1:length(lam_try)
45     %rec_lam(i,:)=fminunc(fun,lam_init(i),options_optim);
46     rec_lam(i,:) = [lam_try(i),objFunc(lam_try(i),X,Y,d_psi,d_phi,b_hat,M,T,d
   )];
47 end
48 [~,opt_ind] = max(rec_lam(:,2));

```

```

49 lam_opt = rec_lam(opt_ind,1);
50 end

```

Listing 4: Matlab code for find the optimal λ using grid search.

```

1 function h=plot_all(dat1,dat2,dat3,dat4,dat5,dat6,pattern,text)
2     subplot(3,2,1);
3     hold on;
4     box on;
5     h=plot(dat1(:,1),dat1(:,2),pattern);
6     axis([1959,2008,-inf,inf])
7     % legend('Observed Data','DC Prediction')
8     %legend(h,text)
9     title('log-real GDP')
10
11     subplot(3,2,2);
12     hold on;
13     box on;
14     plot(dat2(:,1),dat2(:,2),pattern);
15     axis([1959,2008,-inf,inf])
16     % legend('Observed Data','DC Prediction')
17     title('quarterly price inflation')
18
19     subplot(3,2,3);
20     hold on;
21     box on;
22     plot(dat3(:,1),dat3(:,2),pattern);%,xt,rt2.predgdp,'-m',xt,rt3.predgdp
23     ,'-c',xt,rt4.predgdp,':r')
24     axis([1959,2008,-inf,inf])
25     % legend('Observed Data','DC Prediction')
26     title('the federal funds rate')
27
28     subplot(3,2,4);
29     hold on;
30     box on;
31     plot(dat4(:,1),dat4(:,2),pattern);%,xt,rt2.predgdp,'-m',xt,rt3.predgdp
32     ,'-c',xt,rt4.predgdp,':r')
33     axis([1959,2008,-inf,inf])
34     % legend('Observed Data','DC Prediction')
35     title('quarterly nominal wage inflation')
36
37     subplot(3,2,5);
38     hold on;
39     box on;
40     plot(dat5(:,1),dat5(:,2),pattern);%,xt,rt2.predgdp,'-m',xt,rt3.predgdp
41     ,'-c',xt,rt4.predgdp,':r')
42     axis([1959,2008,-inf,inf])
43     % legend('Observed Data','DC Prediction')
44     title('log-labor share')
45
46     subplot(3,2,6);

```

```

44 hold on;
45 box on;
46 plot(dat6(:,1),dat6(:,2),pattern);%,xt,rt2.predgdp,'-m',xt,rt3.predgdp
,'-c',xt,rt4.predgdp,':r')
47 axis([1959,2008,-inf,inf])
48 % legend('Observed Data','DC Prediction')
49 title('log-consumption ratio')
50 end

```

Listing 5: Matlab code for plot data.

References

- [1] D. Giannone, M. Lenza, and G. E. Primiceri, Priors for the long run, (2016).
- [2] J. D. Hamilton, *Time series analysis*, volume 2, Princeton university press Princeton, 1994.