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:

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

- Extending window:

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

Here I tried both cases and get the MSFE:

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

MSFE	log-real GDP	log-real GDP	log-GDP deflator	log-GDP deflator
Average growth rate	1 quarter	4 quarters	1 quarter	4 quarters
Moving window	$2.500e-3$	$3.872e-4$	$1.374e-3$	$4.272e-4$
Extending window	$1.919e-3$	$2.195e-4$	$1.246e-3$	$2.289e-4$

Table 1: Mean squared forecast error for moving window and extending window.

(2) According to [1], with Minnesota prior as specified, the posterior would looks like:

$$\beta|\Sigma, Y \sim \mathcal{N}(\hat{\beta}, \Sigma \otimes \Omega) \tag{1}$$

$$\hat{B} = (X^T X + \Omega^{-1})^{-1}(X^T Y + \Omega^{-1}\hat{b}) \tag{2}$$

$$\hat{\beta} = vec[\hat{B}] \tag{3}$$

$$b = vec[\hat{b}] \tag{4}$$

$$\Omega = \lambda^2 \begin{bmatrix} 10^6 & & & & & & & \\ & \frac{1}{\hat{\sigma}_1^2} & & & & 0 & & \\ & & \ddots & & & & & \\ & & & \frac{1}{\hat{\sigma}_M^2} & & & & 0 \\ & & & & \frac{2^{-2}}{\hat{\sigma}_1^2} & & 0 & \\ & & & & & \ddots & & \\ & & & & & & \frac{2^{-2}}{\hat{\sigma}_M^2} & \\ & & & & & & & \ddots \end{bmatrix}_{K\times K} \tag{5}$$

$$b = ([0, 1, 0, \cdots, 0|0, 0, 1, 0, \cdots, 0|\cdots]^T)_{K\times 1} \tag{6}$$

$$\begin{aligned} \hat{\sigma}_i^2 &= \frac{1}{T}\sum_{t=1}^{T}\left[(y_{i,t} - \hat{c}_i - \hat{\phi}_i y_{i,t-1})^2\right] \\ &= \frac{1}{T}\sum_{t=1}^{T}(y_{i,t}^2 + \hat{c}_i^2 + \hat{\phi}_i^2 y_{i,t-1}^2 + 2\hat{c}_i\hat{\phi}_i y_{i,t-1} - 2\hat{c}_i y_{i,t} - 2\hat{\phi}_i y_{i,t-1} y_{i,t}) \end{aligned} \tag{7}$$

$$\begin{bmatrix} \hat{c}_i \\ \hat{\phi}_i \end{bmatrix} = \begin{bmatrix} T & \sum_{t=1}^T y_{i,t-1} \\ \sum_{t=1}^T y_{i,t-1} & \sum_{t=1}^T y_{i,t-1}^2 \end{bmatrix}^{-1} \begin{bmatrix} \sum_{t=1}^T y_{i,t} \\ \sum_{t=1}^T y_{i,t-1} y_{i,t} \end{bmatrix} \tag{8}$$

where $K = Mp + 1$, $M = dim(y)$, $p = max\ time\ lag$, and the given observations are $y_{-p+1}, \cdots, y_0$ and the values to forecast are $y_1, \cdots, y_T$. 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):

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

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:

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

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.

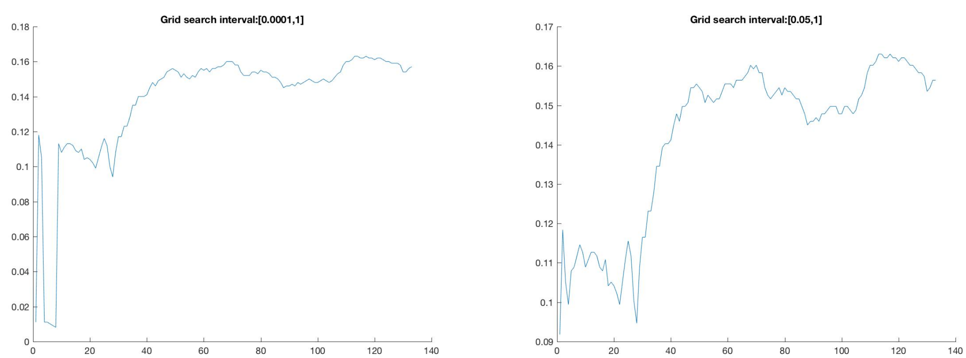

Figure 1: The optimal λ with extending windows and two different search intervals. I am not sure why when λ is near 0 the path has a big jump. From MSFE, we see they are comparable with those obtained by setting $\lambda = 0.2$.

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:

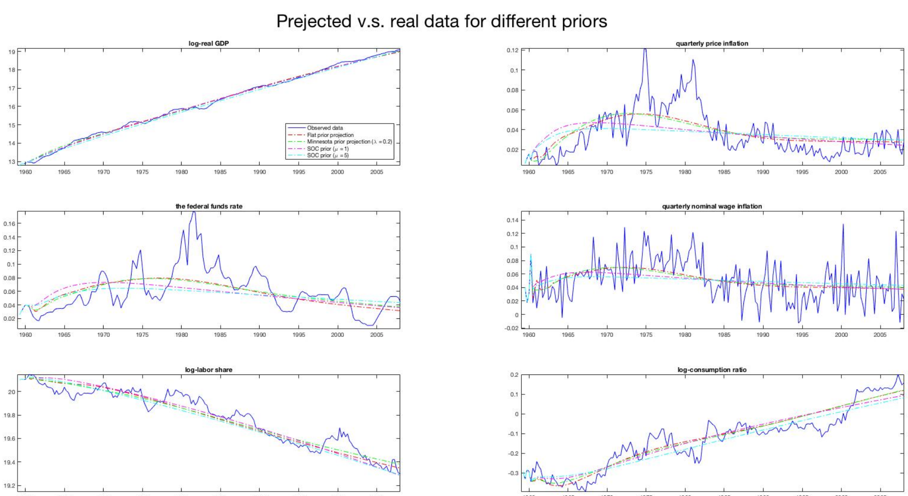

Figure 2: The projections based on the whole data set with flat prior, Minnesota prior, SOC with $\mu = 1$, and SOC with $\mu = 5$.

Different priors give very similar results.

```
clear();
% Load data into the working space
load('dataVARmedium','-mat');
p = 5;
T = 64-5;
N = size(y,1);
M = 7;
K = M*p+1;
h = 4;

%% (1) OLS solution for prediction
%% Moving window prediction
tic
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
for i = 1:(N-(T+p+h)+1)
    % design matrix
    X = zeros(T,K);% a T*K matrix
    for j = 1:T
        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
    end
    % OLS solution for coefficients
    B_hat = (X'*X)\X'*y((i+p:i+p+T-1),:);% a K*M matrix
    % for h = 1 & 4
    pred_y = zeros(h,M);% a h*M matrix
    y_lag = [1,reshape(y(i+p+T-1:-1:i+T,:)',1,K-1)];% lag of y for the prediction
    for k = 1:h
        % forecast
        pred_y(k,:) = y_lag*B_hat;
        if k<h
            % reform the lag for y of prediction
            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)];
        end
    end
    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
    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
    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
    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
end
% Calculate the average squared forecast error
MSFE1 = sum((z_comp(:,1:4)-z_comp(:,5:8)).^2,1)/size(z_comp,1)
csvwrite('mse_ex1-1-moving.csv',MSFE1)
```

```
toc
%% Expanding window prediction
tic
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
for i = 1:(N-(T+p+h)+1)
    if i ==1
        % design matrix
        X = zeros(T,K);% a T*K matrix
        for j = 1:T
            X(j,:) = [1,reshape(y((j+p-1:-1:j),:)',1,K-1)];% have to transpose because the reshape function operate in column
        end
    else
        new_row = [1,reshape(y((i+T+p-2:-1:i+T-1),:)',1,K-1)];
        X = [X;new_row]; % a (T+i-1)*K matrix
    end
    % OLS solution for coefficients
    B_hat = (X'*X)\X'*y((p+1:p+T+i-1),:);% a K*M matrix
    % for h = 1 & 4
    pred_y = zeros(h,M);% a h*M matrix
    y_lag = [1,reshape(y(T+i+p-1:-1:T+i,:)',1,K-1)];% lag of y for the prediction
    for k = 1:h
        % forecast
        pred_y(k,:) = y_lag*B_hat;
        if k<h
            % reform the lag for y of prediction
            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)];
        end
    end
    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
    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
    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
    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
end
% Calculate the average squared forecast error
MSFE2 = sum((z_comp(:,1:4)-z_comp(:,5:8)).^2,1)/size(z_comp,1)
csvwrite('mse_ex1-1-extending.csv',MSFE2)
toc
%% (2)Minnesota prior with a given lambda
%% Expanding window prediction
tic
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
```

```matlab
% Given lambda
lam = 0.2;
% Estimate b
b_hat = [zeros(M,1),eye(M,M*p)];
for i = 1:(N-(T+p+h)+1)
    if i ==1
        % design matrix
        X = zeros(T,K);% a T*K matrix
        for j = 1:T
            X(j,:) = [1,reshape(y((j+p-1:-1:j),:)',1,K-1)];% have to transpose because the reshape function operate in column
        end
    else
        new_row = [1,reshape(y((i+T+p-2:-1:i+T-1),:)',1,K-1)];
        X = [X;new_row]; % a (T+i-1)*K matrix
    end
    % Estimate elements of Omega
    % The first observation is given (regarded as index 0) and the rest of observation up to the
    % end of current window (index 1 to T) are for prediction as stated in the problem
    % statement
    % Follow Hamilton 1994 to get the MLE for variance
    sum1 = sum(y(1:p+T+i-2,:),1);%sum_1^T y_(t-1)
    sum2 = sum(y(2:p+T+i-1,:),1);%sum_1^T y_(t)
    sum3 = sum(y(1:p+T+i-2,:).^2,1);%sum_1^T y_(t-1)^2
    sum4 = sum(y(2:p+T+i-1,:).^2,1);%sum_1^T y_(t)^2
    sum5 = sum(y(2:p+T+i-1,:).*y(1:p+T+i-2,:),1);%sum_1^T y_(t)y_(t-1)
    sigma2_hat = zeros(1,M);
    for m = 1:M
        c_phi_hat = [p+T+i-2,sum1(m);sum1(m),sum3(m)]\[sum2(m);sum5(m)];% c_phi_hat = [c_hat,phi_hat]
        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)...
                        -2*c_phi_hat(1)*sum2(m)-2*c_phi_hat(2)*sum5(m))+c_phi_hat(1)^2;
    end
    omega_d = [10^6];
    for j = 1:p
        omega_d = [omega_d,(sigma2_hat*j^2).^(-1)];
    end
    % Times the lambda^2
    omega_d = lam^2 * omega_d;
    % MAP solution for coefficients
    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
    % for h = 1 & 4
    pred_y = zeros(h,M);% a h*M matrix
    y_lag = [1,reshape(y(T+i+p-1:-1:T+i,:)',1,K-1)];% lag of y for the prediction
```

```
    for k = 1:h
        % forecast
        pred_y(k,:) = y_lag*B_hat;
        if k<h
            % reform the lag for y of prediction
            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)];
        end
    end
    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
    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
    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
    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
end
% Calculate the average squared forecast error
MSFE3 = sum((z_comp(:,1:4)-z_comp(:,5:8)).^2,1)/size(z_comp,1)
csvwrite('mse_ex1-1-lambda_given.csv',MSFE3)
toc
%% (3)Minnesota prior without a given lambda (optimize marginal likelihood to get lambda each time)
%% Expanding window prediction
tic
%Cannot set lambda = 0, because when take log of lambda, 0 is not
%meaningful
lam_bd = [0.0001,1];
N_try = 1000;
d = M + 2;
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
% Record optimal value of lambda
rec_lam_opt = zeros(N-(T+p+h)+1,1);
% Estimate b
b_hat = [zeros(M,1),eye(M,M*p)];
for i = 1:(N-(T+p+h)+1)
    if i ==1
        % design matrix
        X = zeros(T,K);% a T*K matrix
        for j = 1:T
            X(j,:) = [1,reshape(y((j+p-1:-1:j),:)',1,K-1)];% have to transpose because the reshape function operate in column
        end
    else
        new_row = [1,reshape(y((i+T+p-2:-1:i+T-1),:)',1,K-1)];
        X = [X;new_row]; % a (T+i-1)*K matrix
    end
    % Estimate elements of Omega
```

```
    % The first observation is given and the rest of observation up to the
    % end of current window are for prediction as stated in the problem
    % statement
    % Follow Hamilton 1994 to get the MLE for variance
    sum1 = sum(y(1:p+T+i-2,:),1);%sum_1^T y_(t-1)
    sum2 = sum(y(2:p+T+i-1,:),1);%sum_1^T y_(t)
    sum3 = sum(y(1:p+T+i-2,:).^2,1);%sum_1^T y_(t-1)^2
    sum4 = sum(y(2:p+T+i-1,:).^2,1);%sum_1^T y_(t)^2
    sum5 = sum(y(2:p+T+i-1,:).*y(1:p+T+i-2,:),1);%sum_1^T y_(t)y_(t-1)
    sigma2_hat = zeros(1,M);
    for m = 1:M
        c_phi_hat = [p+T+i-2,sum1(m);sum1(m),sum3(m)]\[sum2(m);sum5(m)];%c_phi_hat = [c_hat,phi_hat]
        temp_sum = 0;
        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)...
                            -2*c_phi_hat(1)*sum2(m)-2*c_phi_hat(2)*sum5(m))+c_phi_hat(1)^2;
    end
    omega_d = [10^6];
    for j = 1:p
        omega_d = [omega_d,(sigma2_hat*j^2).^(-1)];
    end
    % Optimize in terms of lambda
    % Use grid-search method
    %options_optim = optimoptions(@fminunc,'Display','iter-detailed','Algorithm','quasi-newton','SpecifyObjectiveGradient',true,'MaxIterations',1000);
    rec_lam_opt(i) = lam_mle_grid(@logmlikelih,lam_bd,N_try,X,y((p+1:p+T+i-1),:),sigma2_hat,omega_d,b_hat',M,T+i-1,d);
    % Times the lambda^2
    omega_d = rec_lam_opt(i)^2 * omega_d;
    % MAP solution for coefficients
    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
    % for h = 1 & 4
    pred_y = zeros(h,M);% a h*M matrix
    y_lag = [1,reshape(y(T+i+p-1:-1:T+i,:)',1,K-1)];% lag of y for the prediction
    for k = 1:h
        % forecast
        pred_y(k,:) = y_lag*B_hat;
        if k<h
            % reform the lag for y of prediction
            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)];
        end
    end
    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
```

```
    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
    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
    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
end
% Calculate the average squared forecast error
MSFE4 = sum((z_comp(:,1:4)-z_comp(:,5:8)).^2,1)/size(z_comp,1)
csvwrite('mse_ex1-1-lambda_opt.csv',MSFE4)
csvwrite('lambda_opt.csv',rec_lam_opt)
figure();
hold on;
plot(rec_lam_opt);
int_str = ['[',num2str(lam_bd(1)),',',num2str(lam_bd(2)),']'];
title(['Grid search interval:',int_str]);
cwd = '/Users/kungangzhang/Documents/OneDrive/Northwestern/Study/Courses/ECON-482/HW1/';
saveas(gca,[cwd,['lambda_opt_1']],'fig');
saveas(gca,[cwd,['lambda_opt_1']],'jpg');
toc
```

Listing 1: Matlab code for problem 1.

```
clear();
% Load data into the working space
load('dataVARmedium','-mat');

lgGDP = y(:,1);%log-real GDP
qpriinf = diff(y(:,2));%quarterly price inflation
fedfr = y(:,3);%federal funds rate
qnomwainf = diff(y(:,7)+y(:,2));%quarterly nominal wage inflation
lglabsh = y(:,7)+y(:,6)-y(:,1);%log-labor share
lgcomra = y(:,4)-y(:,1);%log-consumption ratio

p = 5;
N = size(y,1);
T = N-p;%Initial time horizon for forecasting
M = 7;
K = M*p+1;
h = 0;%The maximum horizon of out-of-sample forecast

%% (1) OLS solution for prediction
tic
for i = 1:(N-(T+p+h)+1)
    if i ==1
        % design matrix
        X = zeros(T,K);% a T*K matrix
        for j = 1:T
            X(j,:) = [1,reshape(y((j+p-1:-1:j),:)',1,K-1)];% have to transpose because the reshape function operate in column
```

```
        end
    else
        new_row = [1,reshape(y((i+T+p-2:-1:i+T-1),:)',1,K-1)];
        X = [X;new_row]; % a (T+i-1)*K matrix
    end
    % OLS solution for coefficients
    B_hat = (X'*X)\X'*y((p+1:p+T+i-1),:);% a K*M matrix
    % Do projection based on MLE of coefficients
    yproj1 = y(1:p,:);
    for i = p+1:N
        yproj1(i,:) = [1,reshape(yproj1(i-1:-1:i-p,:)',1,K-1)]*B_hat;
    end
    lgGDP1 = yproj1(:,1);
    qpriinf1 = diff(yproj1(:,2));
    fedfr1 = yproj1(:,3);
    qnomwainf1 = diff(yproj1(:,7)+yproj1(:,2));
    lglabsh1 = yproj1(:,7)+yproj1(:,6)-yproj1(:,1);
    lgcomra1 = yproj1(:,4)-yproj1(:,1);
end
xt = ((1959+0.25): 0.25: (2008+1))';
figure();
hold on;
box on;
%title('Prejected v.s. real data');

annotation('textbox', [0 0.9 1 0.1], ...
    'String', 'Prejected v.s. real data for different priors', ...
    'EdgeColor', 'none', ...
    'HorizontalAlignment', 'center')

legendinfo{1} = ['Observed data'];
handle(1) = plot_all([xt,lgGDP],[xt(2:end),qpriinf],[xt,fedfr],[xt(2:end),
    qnomwainf],[xt,lglabsh],[xt,lgcomra],'-b','Observed data');
xt1 = xt(1:end);
legendinfo{2} = ['Flat prior projection'];
handle(2) = plot_all([xt1,lgGDP1],[xt1(2:end),qpriinf1],[xt1,fedfr1],[xt1(2:
    end),qnomwainf1],[xt1,lglabsh1],[xt1,lgcomra1],'-.r','Flat prior
    projection');
toc

%% (2)Minnesota prior with a given lambda
tic
% Given lambda
lam = 0.2;
% Estimate b
b_hat = [zeros(M,1),eye(M,M*p)];
for i = 1:(N-(T+p+h)+1)
    if i ==1
        % design matrix
        X = zeros(T,K);% a T*K matrix
```

```
        for j = 1:T
            X(j,:) = [1,reshape(y((j+p-1:-1:j),:)',1,K-1)];% have to transpose because the reshape function operate in column
        end
    else
        new_row = [1,reshape(y((i+T+p-2:-1:i+T-1),:)',1,K-1)];
        X = [X;new_row]; % a (T+i-1)*K matrix
    end
    % Estimate elements of Omega
    % The first observation is given (regarded as index 0) and the rest of observation up to the
    % end of current window (index 1 to T) are for prediction as stated in the problem
    % statement
    % Follow Hamilton 1994 to get the MLE for variance
    sum1 = sum(y(1:p+T+i-2,:),1);%sum_1^T y_(t-1)
    sum2 = sum(y(2:p+T+i-1,:),1);%sum_1^T y_(t)
    sum3 = sum(y(1:p+T+i-2,:).^2,1);%sum_1^T y_(t-1)^2
    sum4 = sum(y(2:p+T+i-1,:).^2,1);%sum_1^T y_(t)^2
    sum5 = sum(y(2:p+T+i-1,:).*y(1:p+T+i-2,:),1);%sum_1^T y_(t)y_(t-1)
    sigma2_hat = zeros(1,M);
    for m = 1:M
        c_phi_hat = [p+T+i-2,sum1(m);sum1(m),sum3(m)]\[sum2(m);sum5(m)];% c_phi_hat = [c_hat,phi_hat]
        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)...
                        -2*c_phi_hat(1)*sum2(m)-2*c_phi_hat(2)*sum5(m))+c_phi_hat(1)^2;
    end
    omega_d = [10^6];
    for j = 1:p
        omega_d = [omega_d,(sigma2_hat*j^2).^(-1)];
    end
    % Times the lambda^2
    omega_d = lam^2 * omega_d;
    % MAP solution for coefficients
    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
    % Do projection based on MLE of coefficients
    yproj2 = y(1:p,:);
    for i = p+1:N
        yproj2(i,:) = [1,reshape(yproj2(i-1:-1:i-p,:)',1,K-1)]*B_hat;
    end
    lgGDP2 = yproj2(:,1);
    qpriinf2 = diff(yproj2(:,2));
    fedfr2 = yproj2(:,3);
    qnomwainf2 = diff(yproj2(:,7)+yproj2(:,2));
    lglabsh2 = yproj2(:,7)+yproj2(:,6)-yproj2(:,1);
    lgcomra2 = yproj2(:,4)-yproj2(:,1);
end
```

```
xt2 = xt(1:end);
legendinfo{3} = ['Minnesota prior projection (\lambda = 0.2)'];
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)');

toc
%% (3)Minnesota prior without a given lambda (optimize marginal likelihood to get lambda each time)
tic
% Given lambda
lam = 0.2;
mu = 1;
% Estimate b
b_hat = [zeros(M,1),eye(M,M*p)];
for i = 1:(N-(T+p+h)+1)
    if i ==1
        % design matrix
        X = zeros(T,K);% a T*K matrix
        for j = 1:T
            X(j,:) = [1,reshape(y((j+p-1:-1:j),:)',1,K-1)];% have to transpose because the reshape function operate in column
        end
    else
        new_row = [1,reshape(y((i+T+p-2:-1:i+T-1),:)',1,K-1)];
        X = [X;new_row]; % a (T+i-1)*K matrix
    end
    % Estimate elements of Omega
    % The first observation is given (regarded as index 0) and the rest of observation up to the
    % end of current window (index 1 to T) are for prediction as stated in the problem
    % statement
    % Follow Hamilton 1994 to get the MLE for variance
    sum1 = sum(y(1:p+T+i-2,:),1);%sum_1^T y_(t-1)
    sum2 = sum(y(2:p+T+i-1,:),1);%sum_1^T y_(t)
    sum3 = sum(y(1:p+T+i-2,:).^2,1);%sum_1^T y_(t-1)^2
    sum4 = sum(y(2:p+T+i-1,:).^2,1);%sum_1^T y_(t)^2
    sum5 = sum(y(2:p+T+i-1,:).*y(1:p+T+i-2,:),1);%sum_1^T y_(t)y_(t-1)
    sigma2_hat = zeros(1,M);
    for m = 1:M
        c_phi_hat = [p+T+i-2,sum1(m);sum1(m),sum3(m)]\[sum2(m);sum5(m)];% c_phi_hat = [c_hat,phi_hat]
        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)...
                            -2*c_phi_hat(1)*sum2(m)-2*c_phi_hat(2)*sum5(m))+c_phi_hat(1)^2;
    end
    omega_d = [10^6];
    for j = 1:p
```

```
            omega_d = [omega_d,(sigma2_hat*j^2).^(-1)];
        end
        % Times the lambda^2
        omega_d = lam^2 * omega_d;

        %stack the artificial observations at the bottom to get soc prior
        Xplus = [zeros(M,1),repmat(diag(mu*mean(y(1:p,:),1)),[1 p])];
        X= [X;Xplus]; % stack x at the bottom
        yplus = diag(mu*mean(y(1:p,:),1));
        ynew = [y((p+1:p+T+i-1),:);yplus];

        % MAP solution for coefficients
        B_hat = (X'*X+diag(omega_d.^(-1)))\(X'*ynew+diag(omega_d.^(-1))*b_hat');% a K*M matrix
        % Do projection based on MLE of coefficients
        yproj3 = y(1:p,:);
        for j = p+1:N
            yproj3(j,:) = [1,reshape(yproj3(j-1:-1:j-p,:)',1,K-1)]*B_hat;
        end
        lgGDP3 = yproj3(:,1);
        qpriinf3 = diff(yproj3(:,2));
        fedfr3 = yproj3(:,3);
        qnomwainf3 = diff(yproj3(:,7)+yproj3(:,2));
        lglabsh3 = yproj3(:,7)+yproj3(:,6)-yproj3(:,1);
        lgcomra3 = yproj3(:,4)-yproj3(:,1);
end
xt3 = xt(1:end);
legendinfo{4} = ['SOC prior (\mu = 1)'];
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)');

toc

tic
% Given lambda
lam = 0.2;
mu = 5;
% Estimate b
b_hat = [zeros(M,1),eye(M,M*p)];
for i = 1:(N-(T+p+h)+1)
    if i ==1
        % design matrix
        X = zeros(T,K);% a T*K matrix
        for j = 1:T
            X(j,:) = [1,reshape(y((j+p-1:-1:j),:)',1,K-1)];% have to transpose because the reshape function operate in column
        end
    else
        new_row = [1,reshape(y((i+T+p-2:-1:i+T-1),:)',1,K-1)];
```

```
        X = [X;new_row]; % a (T+i-1)*K matrix
    end
    % Estimate elements of Omega
    % The first observation is given (regarded as index 0) and the rest of observation up to the
    % end of current window (index 1 to T) are for prediction as stated in the problem
    % statement
    % Follow Hamilton 1994 to get the MLE for variance
    sum1 = sum(y(1:p+T+i-2,:),1);%sum_1^T y_(t-1)
    sum2 = sum(y(2:p+T+i-1,:),1);%sum_1^T y_(t)
    sum3 = sum(y(1:p+T+i-2,:).^2,1);%sum_1^T y_(t-1)^2
    sum4 = sum(y(2:p+T+i-1,:).^2,1);%sum_1^T y_(t)^2
    sum5 = sum(y(2:p+T+i-1,:).*y(1:p+T+i-2,:),1);%sum_1^T y_(t)y_(t-1)
    sigma2_hat = zeros(1,M);
    for m = 1:M
        c_phi_hat = [p+T+i-2,sum1(m);sum1(m),sum3(m)]\[sum2(m);sum5(m)];% c_phi_hat = [c_hat,phi_hat]
        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)...
                            -2*c_phi_hat(1)*sum2(m)-2*c_phi_hat(2)*sum5(m))+c_phi_hat(1)^2;
    end
    omega_d = [10^6];
    for j = 1:p
        omega_d = [omega_d,(sigma2_hat*j^2).^(-1)];
    end
    % Times the lambda^2
    omega_d = lam^2 * omega_d;

    %stack the artificial observations at the bottom to get soc prior
    Xplus = [zeros(M,1),repmat(diag(mu*mean(y(1:p,:),1)),[1 p])];
    X= [X;Xplus]; % stack x at the bottom
    yplus = diag(mu*mean(y(1:p,:),1));
    ynew = [y((p+1:p+T+i-1),:);yplus];

    % MAP solution for coefficients
    B_hat = (X'*X+diag(omega_d.^(-1)))\(X'*ynew+diag(omega_d.^(-1))*b_hat');% a K*M matrix
    % Do projection based on MLE of coefficients
    yproj4 = y(1:p,:);
    for j = p+1:N
        yproj4(j,:) = [1,reshape(yproj4(j-1:-1:j-p,:)',1,K-1)]*B_hat;
    end
    lgGDP4 = yproj4(:,1);
    qpriinf4 = diff(yproj4(:,2));
    fedfr4 = yproj4(:,3);
    qnomwainf4 = diff(yproj4(:,7)+yproj4(:,2));
    lglabsh4 = yproj4(:,7)+yproj4(:,6)-yproj4(:,1);
    lgcomra4 = yproj4(:,4)-yproj4(:,1);
```

```matlab
end
xt4 = xt(1:end);
legendinfo{5} = ['SOC prior (\mu = 5)'];
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)');

toc

var1=[var(lgGDP),var(lgGDP1),var(lgGDP2),var(lgGDP3),var(lgGDP4)];
var2=[var(qpriinf),var(qpriinf1),var(qpriinf2),var(qpriinf3),var(qpriinf4)];
var3=[var(fedfr),var(fedfr1),var(fedfr2),var(fedfr3),var(fedfr4)];
var4=[var(qnomwainf),var(qnomwainf1),var(qnomwainf2),var(qnomwainf3),var(qnomwainf4)];
var5=[var(lglabsh),var(lglabsh1),var(lglabsh2),var(lglabsh3),var(lglabsh4)];
var6=[var(lgcomra),var(lgcomra1),var(lgcomra2),var(lgcomra3),var(lgcomra4)];
var_all = [var1;var2;var3;var4;var5;var6];
legend(handle,legendinfo,'Location','southeast');
```

Listing 2: Matlab code for problem 2.

```matlab
function val = logmlikelih(lam,X,Y,d_psi,d_phi,b_hat,M,T,d)
% Calculate the log marginal likelihood
% Input:
% lam: lambda value which we are optimizing in terms of
% X: matrix X in time series (lag);
% Y: matrix Y which are to be forecasted;
% d_psi: diagnoal elements of Psi matrix (which is diagonal);
% d_ome: diagonal elements of Omega matrix (which is diagonal);
% M: the dimension of y
% T: time horizon to be forecasted
% d: some constant specified in prior
% Output:
% val: the log marginal likelihood up to adding a constant
tempm1 = X*diag(d_phi.^0.5);
eig1 = eig(tempm1'*tempm1);
B_hat = (lam^2*(X'*X)+diag(d_phi.^(-1)))\(lam^2*X'*Y + diag(d_phi.^(-1))*b_hat);
eps_hat = Y - X*B_hat;
tempm2 = B_hat-b_hat;
eig2 = eig(diag(d_psi.^(-0.5))*(lam^2*(eps_hat'*eps_hat)+tempm2'*diag(d_phi.^(-1))*tempm2)*diag(d_psi.^(-0.5)));
val = (-M/2)*sum(log(1+lam^2*eig1))+(-(T+d)/2)*sum(log(lam^2+eig2))+M*(T+d)*log(lam);
end
```

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

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

```matlab
% Use grid search to find the optimal solution for lam with several starting points
% Input:
% objFunc: objective function which return kernel of marginal likelihood
% (or log of it)
%================================
    % Calculate the marginal likelihood
    % Input:
    % lam: lambda value which we are optimizing in terms of
    % X: matrix X in time series (lag);
    % Y: matrix Y which are to be forecasted;
    % d_psi: diagnoal elements of Psi matrix (which is diagonal);
    % d_ome: diagonal elements of Omega matrix (which is diagonal);
    % M: the dimension of y
    % T: time horizon to be forecasted
    % d: some constant specified in prior
    % Output:
    % val: the marginal likelihood up to a constant factor
%================================
% lam_bd: lambda value upper and lower bound for initial value exploring
%================================
    % [lam_lower_bound,lam_upper_bound]
%================================
% N_try: how many initial tries to do inside the lam_bd
% X: matrix X in time series (lag);
% Y: matrix Y which are to be forecasted;
% d_psi: diagnoal elements of Psi matrix (which is diagonal);
% d_ome: diagonal elements of Omega matrix (which is diagonal);
% M: the dimension of y
% T: time horizon to be forecasted
% d: some constant specified in prior
% options_optim: options when calling the fminunc function
% Output:
% opt_lam: optimal lambda that maximizing the marginal likelihood

%% code
%options_optim = optimoptions(@fminunc,'Display','iter-detailed','Algorithm','trust-region','SpecifyObjectiveGradient',true);

%fun = @(lam)objFunc(lam,X,Y,d_psi,d_phi,b_hat,M,T,d);


lam_try = lam_bd(1):(lam_bd(2)-lam_bd(1))/N_try:lam_bd(2);
rec_lam = zeros(length(lam_try),2);
for i = 1:length(lam_try)
    %rec_lam(i,:)=fminunc(fun,lam_init(i),options_optim);
    rec_lam(i,:) = [lam_try(i),objFunc(lam_try(i),X,Y,d_psi,d_phi,b_hat,M,T,d)];
end
[~,opt_ind] = max(rec_lam(:,2));
```

```
lam_opt = rec_lam(opt_ind,1);
end
```

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

```
function h=plot_all(dat1,dat2,dat3,dat4,dat5,dat6,pattern,text)
    subplot(3,2,1);
    hold on;
    box on;
    h=plot(dat1(:,1),dat1(:,2),pattern);
    axis([1959,2008,-inf,inf])
    % legend('Observed Data','DC Prediction')
    %legend(h,text)
    title('log-real GDP')

    subplot(3,2,2);
    hold on;
    box on;
    plot(dat2(:,1),dat2(:,2),pattern);
    axis([1959,2008,-inf,inf])
    % legend('Observed Data','DC Prediction')
    title('quarterly price inflation')

    subplot(3,2,3);
    hold on;
    box on;
    plot(dat3(:,1),dat3(:,2),pattern);%,xt,rt2.predgdp,'-.m',xt,rt3.predgdp,'--c',xt,rt4.predgdp,':r')
    axis([1959,2008,-inf,inf])
    % legend('Observed Data','DC Prediction')
    title('the federal funds rate')

    subplot(3,2,4);
    hold on;
    box on;
    plot(dat4(:,1),dat4(:,2),pattern);%,xt,rt2.predgdp,'-.m',xt,rt3.predgdp,'--c',xt,rt4.predgdp,':r')
    axis([1959,2008,-inf,inf])
    % legend('Observed Data','DC Prediction')
    title('quarterly nominal wage inflation')

    subplot(3,2,5);
    hold on;
    box on;
    plot(dat5(:,1),dat5(:,2),pattern);%,xt,rt2.predgdp,'-.m',xt,rt3.predgdp,'--c',xt,rt4.predgdp,':r')
    axis([1959,2008,-inf,inf])
    % legend('Observed Data','DC Prediction')
    title('log-labor share')

    subplot(3,2,6);
```

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