A panel data toolbox for MATLAB

James P. LeSage
Department of Economics
University of Toledo

June, 2021

Preface

This manual describes a set of MATLAB functions that produce *static* panel data model estimates. Toolboxes are the name given by the MathWorks Inc. to related sets of MATLAB functions aimed at solving a particular class of problems. Toolboxes of functions useful in signal processing, optimization, statistics, finance and a host of other areas are available from the MathWorks Inc. as add-ons to the standard MATLAB software distribution. I use the term *Bayesian Panel Data Toolbox* to refer to the collection of functions described in this manual.

The intended audience is faculty and students interested in applied econometric modeling based on the increasingly popular panel data sets. The functions described here work only for *static*, *balanced panel* data models, where the number of observational units remain the same for all time periods.

The functions allow both maximum likelihood and Bayesian MCMC estimation of a set of spatial regression models, including ordinary least-squares (OLS), spatial autoregressive (SAR), spatial Durbin (SDM), spatial error (SEM), spatial Durbin error (SDEM), spatial lag of X (SLX), and convex combination of spatial weights models recently proposed by Debarsy and LeSage (2021).

In addition to functions for estimation, there are also functions for model comparison based on Bayesian log-marginal likelihoods, that can be used to calculate model probabilities.

The collection of functions and demonstration programs are described in different chapters of this manual. A consistent design was implemented that provides documentation, example programs, and functions to produce printed as well as graphical presentation of estimation and MCMC diagnostic results for most of the estimation functions.

The goal is to allow users to easily compare maximum likelihood and Bayesian MCMC estimation results, and to compare alternative panel data regression model specifications. Given a dependent variable y, a set of explanatory variables X, and one or more spatial weight matrices, W, users should be able to produce estimates from a wide range of alternative panel data spatial (and non-spatial) model specifications. Log-marginal likelihoods should allow comparison of these alternative specifications for their consistency with the sample data set.

These functions have been incorporated into an existing broader toolbox of MATLAB functions for econometrics and spatial econometric estimation. This should put an entire set of econometric routines at the disposal of users. The newly developed panel data estimation functions rely on many functions contained in the broader toolbox of MATLAB functions for econometrics and spatial econometric estimation. So, users need the download the entire toolbox as a compressed zip-file which contains many subfolders of functions for different purposes.

Finally, there are obviously omissions, bugs and perhaps programming errors in the *Econometrics Toolbox*. This would likely be the case with any such endeavor. I would be grateful if users would notify me when they encounter problems.

The latest version of the *Econometrics Toolbox* functions in the form of a compressed zip-file can be found on the Internet at: https://www.spatial-econometrics.com/ under the Download link.

After downloading the zip-file, unzip it which will produce a main folder plus sub-folders. Use the MATLAB 'set path' function to add the main plus sub-folders to your MATLAB path. To check if this was successful, type 'help sar_panel_FE_g' in the MATLAB command window. You should see the documentation for this function appear in the command window.

Acknowledgements

First I would like to acknowledge my students from the fall semester econometrics course in the new Quantitative Finance and Economics program at Texas State University who suffered through a buggy and error filled version of these panel data toolbox functions. I owe a great deal to their patience and persistence in completing projects under such conditions.

I would like to thank J. Paul Elhorst for supplying maximum likelihood estimation functions which have been incorporated in this toolbox, and for his book on *Spatial Panel Data Models* (2013), which greatly enhanced my knowledge on this topic.

As always, R. Kelly Pace provided many insightful discussions and numerous MATLAB functions that are incorporated in these toolbox functions.

Nicolas Debarsy was a collaborator on methods for estimating the convex combinations of spatial weight matrices models that represent an innovative aspect of this toolbox.

My colleague Yao-Yu Chih from Texas State University has also been a great collaborator and person that will listen to my crazy and frequently incorrect ideas.

Contents

List of Figures

List of Tables

Chapter 1

An overview

This chapter provides an enumeration of the functions available in the Toolbox for estimation of various static spatial econometric panel data models. A detailed discussion of the function for estimating the ordinary least-squares (OLS) and spatial autoregressive (SAR) models which has played a prominent role in spatial econometrics is provided. All estimation functions in the toolbox rely on similar input options set to the function using a MATLAB *structure* variable, with *fields* in this variable indicating specific options desired by the user.

The functions also follow a consistent naming convention to be motivated later.

1.1 Model estimation functions

The list of estimation functions for a range of static spatial panel data model is shown below.

```
------- model estimation functions -----------

ols_panel_FE_g    : MCMC regression model estimates for static panels

sar_panel_FE      : ML SAR model estimates for spatial panels

sar_panel_FE_g    : MCMC SAR model estimates for static spatial panels

sdem_panel_FE     : ML SDEM model estimates for  static spatial panels

sdem_panel_FE_g   : MCMC SDEM estimates for static spatial panels

sdm_panel_FE      : ML SDM model estimates for spatial panels

sdm_panel_FE_g    : MCMC SDM model estimates for static spatial panels

sem_panel_FE      : ML SEM model estimates for spatial panels

sem_panel_FE_g    : MCMC SEM model estimates for static spatial panels

slx_panel_FE_g    : MCMC SLX model estimates for static spatial panels

sar_conv_panel_g  : MCMC estimates for SAR convex combination of W model

sem_conv_panel_g  : MCMC estimates for SEM convex combination of W model
```

```
sdm_conv_panel_g  : MCMC estimates for SDM convex combination of W model

sdem_conv_panel_g : MCMC estimates for SDEM convex combination of W model

sar_conv_panel_bma_g  : Bayesian panel SAR model averaged estimates for all M combinations

sdm_conv_panel_bma_g  : Bayesian panel SDM model averaged estimates for all M combinations

sdem_conv_panel_bma_g  : Bayesian panel SDEM model averaged estimates for all M combinations
```

1.1.1 Installing the Panel_g Toolbox functions

A zip file was downloaded from www.spatial-econometrics.com, that is named *toolbox_panelg.zip* file will unzip to produce the following folders:

```
toolbox_panelg (the top-level folder)
subfolders:
demo_data (a folder containing data used for the demos)
documentation (a folder with the Acrobat PDF manual file: panelg_manual.pdf)
demo_programs (a top-level folder)
subfolders:
chapter1 (demos discussed in chapter 1 of the manual)
chapter2 (demos discussed in chapter 2 of the manual)
chapter3 (demos discussed in chapter 3 of the manual)
chapter4 (demos discussed in chapter 4 of the manual)
chapter5 (demos discussed in chapter 5 of the manual)
chapter6 (demos discussed in chapter 6 of the manual)
chapter7 (demos discussed in chapter 7 of the manual)
panel_g (the MATLAB code for the functions)
subfolder:
support_funcs (a folder with some helper functions)
```

Click on the MATLAB *Set Path* tool and select the menu item *add with subfolders* and use the file navigation menu to select the top-level folder *toolbox_panelg*. Then select *save* when exiting the *Set Path* tool. This will add all of the functions, demo programs, demo data to your MATLAB path.

To see if this was successful, type: *help ols_panel_FE_g* in the MATLAB command window. If you have successfully added the toolbox functions to your MATLAB path, you should see documentation for the *ols_panel_FE_g()* function that begins as follows:

```
 PURPOSE: computes MCMC regression model estimates for static panels
          (N regions*T time periods) with spatial fixed effects (sfe)
          and/or time period fixed effects (tfe)
          y = X*b + sfe(optional) + tfe(optional) + e,
          e = N(0,sige*V),
          V = diag(v_1,v_2,...v_N*T), r/vi = ID chi(r)/r, r = 5 (default)
          b   = N(c,C), default c = 0, C = eye(2*k)*1e+12
          sige = gamma(nu,d0), default nu=0, d0=0
 Supply data sorted first by time and then by spatial units, so first region 1,
 region 2, et cetera, in the first year, then region 1, region 2, et
 cetera in the second year, and so on
 ols_panel_FE_g transforms y and x to deviation of the spatial and/or time means
```

```
--------------------------------------------------
 USAGE: results = ols_panel_FE_g(y,x,T,ndraw,nomit,prior)
```

It is important to note that these functions use many of the functions from my Spatial Econometric Toolbox for MATLAB. You need to download the zip file for that toolbox from the www.spatial-econometrics.com website, unzip it and use the same procedure to add the top-level folder (with subfolders) to your MATLAB path. A test of whether this was successful is the type *help ols_g* in the MATLAB command window, which should produce documentation for the *ols_g()* estimation function.

This manual begins with a description of functions for estimation of ordinary least-squares panel data regressions.

1.2 *ols_panel_FE_g()*

This function produces Markov Chain Monte Carlo (MCMC) estimates for a non-spatial regression model shown in (1.1), where y is an $N \times T$ vector of the dependent variable, organized so that all N regions for time period $t = 1$ are first, all N regions for time period $t = 2$, next, and so on. up to time $t = T$. The $NT \times k$ matrix X is organized in the same way.

$$
\begin{aligned}
y &= X\beta + \iota_T \otimes \mu + \nu \otimes \iota_N + \varepsilon && (1.1) \\
\varepsilon &\sim N(0_{NT}, \sigma^2 V) \\
V &= \begin{pmatrix} v_1 & 0 & \dots & 0 \\ 0 & v_2 & \dots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & \dots & 0 & v_{NT} \end{pmatrix} \\
\pi(\beta) &\sim N(c, C) && (1.2) \\
\pi(r/v_i) &\sim iid\ \chi^2(r), i = 1, \dots, N*T && (1.3) \\
\pi(\sigma^2) &\sim IG(a, b) && (1.4)
\end{aligned}
$$

The $\iota_T \otimes \mu$ represent an $N-$vector of region-specific fixed effects, one for each region, and ι_T is a $T-$vector of ones, with $\otimes$ being a Kronecker product that repeats the vector μ for each time period. Similarly, the $\nu \otimes \iota_N$ reflects a Kronecker product of the $T-$vector of time-specific effects, one for each time period. These parameter vectors are not actually estimated, but can be recovered and printed out if you are interested in seeing these.

Bayesian prior distributions can (optionally) be assigned to the parameters β, σ^2 if desired, the default estimation is to not incorporate prior information for these parameters. You can assign a prior value for the parameter r, with the default being $r = 5$, which produces estimates for the heteroscedastic variance scalars $v_i, i = 1, \dots, NT$ that are allowed to vary far from their prior mean values of one (see LeSage and Pace, 2009, p. 146). There is also an option to fix all $v_i = 1$, producing homoscedastic estimates.

The introduction of MCMC estimation for this type of model with variance scalars was introduced by Geweke (1993). The Chi-squared prior distribution assigned for the v_i terms takes the form of a set of NT independent, identically distributed, $\chi^2(r)/r$ distributions, where r represents the single parameter of the χ^2 distribution. This allows estimation of NT individual variance scaling parameters v_i by adding only a single parameter r, to our model. Use of a flexible family of distributions that is controlled by a single parameter such as r to specify a prior distribution is a common Bayesian approach. The parameter r that controls this family of prior distributions has been labeled a *hyperparameter*, allowing changes in this single parameter to restrict the resulting v_i estimates towards the prior mean value of one. Using a value of $r = 30$ will produce estimates for all v_i scalars near one. In this case you should set the option to not use these scalars.

The role of large posterior estimates for the variance scalars v_i is to accommodate outliers or observations containing large variances. These observations will be down-weighted as in the case of generalized least-squares where large variances result in less weight assigned to an observation. I note that this type of distribution has frequently been used to deal with sample data containing outliers Lange, Little, and Taylor (1989).

Some applied examples of using this model follows.

1.2.1 Using the *ols_panel_FE_g()* function

The following code generates a panel data $y-$vector, using known parameter values, then calls the function *ols_panel_fe_g()* to produce estimates and the *prt_panel()* function to print out results.

```
% file: ols_panel_gd.m
clear all;
rng(10203040);

n = 200;
t = 10;

k = 2;
x = randn(n*t,k); % random normal x-variables
beta = ones(k,1); % true beta = 1
sige = 5;         % true noise variance = 5
evec = randn(n*t,1)*sqrt(sige); % random normal disturbances
% fixed effects for regions and time periods
tts = (1:n)*(1/n);
SFE = kron(ones(t,1),tts');
ttt = (1:t)*(1/t);
TFE = kron(ttt',ones(n,1));
% true DGP (data generating process)
y = (x*beta + SFE + TFE + evec);

ndraw = 2500;
nomit = 500;
prior.novi_flag = 1; % homoscedastic model v_{i} = 1
prior.model = 3;     % model with fixed effects for regions and time period
result1 = ols_panel_FE_g(y,x,t,ndraw,nomit,prior);
vnames = strvcat('y','x1','x2');
prt_panel(result1,vnames);

prior2.rval=5;    % heterosedastic model
prior2.model = 1; % model with fixed effects for regions only
% add outliers to y
youtlier = reshape(y,n,t);
youtlier(:,5) = youtlier(:,5) + 10;
```

```
youtlier(:,6) = youtlier(:,6) + 10;
yvec = vec(youtlier);
result2 = ols_panel_FE_g(yvec,x,t,ndraw,nomit,prior2);
prt_panel(result2,vnames);
vmean = result2.vmean;
tt=1:n*t;
plot(tt,vmean);
ylabel('v_{it} estimates');
xlabel('n \times t observations');

prior3.novi_flag = 1;        % homoscedastic model v_{it} = 1
prior3.beta = ones(2,1)*0.5; % prior means for beta
prior3.bcov = eye(2)*0.001;  % prior variances for beta
prior3.nu = 0.1;             % IG(a,b) a-value uninformative
prior3.d0 = 0.1;             % IG(a,b) b-value uninformative
result3 = ols_panel_FE_g(y,x,t,ndraw,nomit,prior3);
prt_panel(result3,vnames);
```

The function call must include the number of MCMC draws to use as well as a MATLAB *structure* variable, named prior here. This allows inputting many options using *fields* of the structure variable, e.g., *prior._novi_flag=1*, which eliminates the variance scalars, setting all $v_i = 1$ during estimation. The field *.model=0,1,2,3* allows choice of a model (=0) with no fixed effects, (=1) region-specific effects, (=2) time-specific effects, and (=3) both region- and time-specific effects.

The first call to the estimation function produced homoscedastic estimates based on setting *prior.novi_flag = 1;*, including both region and time effects because *prior.model=3*.

The second call to the estimation function uses a $y-$vector that was altered to include outliers for all N observations during time periods $t = 5$ and $t = 6$. Robust/heteroscedastic estimates were produced using the *prior2.rval=5;* input option, and only region-specific effects were used, based on *prior2.model=1*.

The last call illustrates input for a prior mean and variance for the β parameters, where we set the prior means for both β_1, β_2 to a value of 0.5, and the prior variances for both of these to 0.001, which implies prior standard deviations of 0.0316 ($\sqrt{0.001} = 0.0316$), which should greatly bias the estimates for these parameters towards their prior mean values. An uninformative prior is placed on σ^2 because IG(a,b), where $a \to 0, b \to 0$ reflects no prior information for this parameter.

The printed output from executing the file *ols_panel_gd.m* is shown below, where we see the first set of homoscedastic estimates for β, σ^2 close to the true values used to generate the $y-$vector.

The second set of estimates introduce outliers by adjusting the actual $y-$vector to have be the true $y-$value plus a value of 10 for time periods 5 and 6. This model is estimated with only region-specific fixed effects. Figure 1.1 shows a plot of the variance scalar v_i estimates produced, which clearly show evidence of the outliers for time periods 5 and 6. Estimates show for the parameters β_1, β_2 are still close to their true values of 1 because the estimation uses $1/v_i$ to down-weight outliers, or observations with large (estimated) variances. This would result in these observations being down-weighted to about 1/3 of the weight given to other observations when calculating estimates for β_1, β_2. Specifically, $\hat{\beta} = (X'V^{-1}X)^{-1}X'V^{-1}y$, where the diagonal $NT \times NT$ matrix V reflect the points plotted in the figure.

```
printed output from: ols_panel_gd.m

Homoscedastic model
```

```
MCMC OLS model with both region and time period fixed effects
Dependent Variable =              y
R-squared           =    0.3800
corr-squared        =    0.2904
sigma^2             =    4.3532
Nobs,Nvar,#FE       =   2000,     2,   210
log-likelihood      =       -4306.9634
prior rvalue        =      0
total time in secs  =    0.1760
ndraws,nomit        =   2500,   500
time for MCMC draws =    0.1480
***************************************************************
Variable      Coefficient  Asymptot t-stat    z-probability
x1               0.965422       19.191054        0.000000
x2               1.021364       20.686742        0.000000

Heteroscedastic model
MCMC OLS model with region fixed effects
Dependent Variable =              y
R-squared           =    0.1091
corr-squared        =    0.0766
sigma^2             =   14.3289
Nobs,Nvar,#FE       =   2000,     2,   200
log-likelihood      =       -5936.6032
prior rvalue        =      5
total time in secs  =    1.2350
ndraws,nomit        =   2500,   500
time for MCMC draws =    1.2250
***************************************************************
Variable      Coefficient  Asymptot t-stat    z-probability
x1               0.907880        8.647284        0.000000
x2               1.022776        9.983947        0.000000

Homoscedastic model
MCMC OLS model with region fixed effects
Dependent Variable =              y
R-squared           =    0.3390
corr-squared        =    0.2891
sigma^2             =    4.6388
Nobs,Nvar,#FE       =   2000,     2,   200
log-likelihood      =       -4371.1105
prior rvalue        =      0
total time in secs  =    0.1500
ndraws,nomit        =   2500,   500
time for MCMC draws =    0.1350
***************************************************************
Variable       Prior Mean    Std Deviation
x1               0.500000        0.031623
x2               0.500000        0.031623

***************************************************************
Variable      Coefficient  Asymptot t-stat    z-probability
x1               0.628465       23.776869        0.000000
x2               0.649580       23.180537        0.000000
```

The final set of estimates are those based on a prior mean and diagonal variance-covariance matrix assigned to the parameters β_1, β_2, where both parameters are assigned prior means of 0.5, and small prior standard deviations of 0.0315. This should "bias" the posterior mean estimates towards the prior mean values, which we see is the case in the printed estimation results. Of course, this indicates that you should use good prior information if you impose the prior values tightly by using small prior variances.

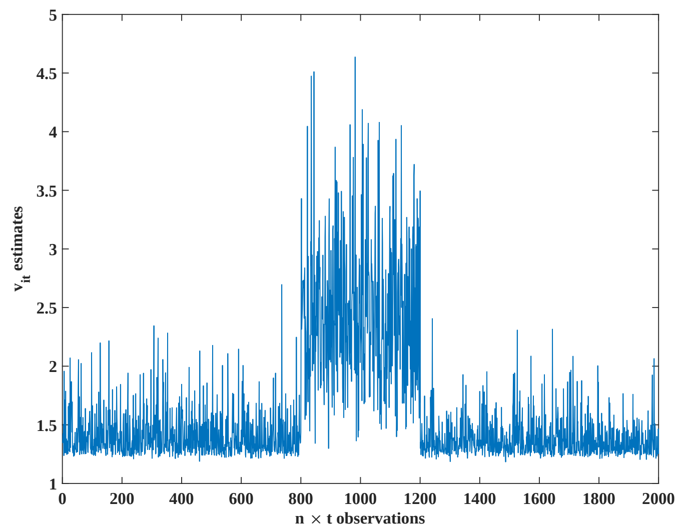

Figure 1.1: Variance scalar estimates v_{it}

A second demonstration file illustrates that we can recover the fixed effects estimates and print or plot these values. A map of these can also be informative in the case of spatial data samples. Printout of the fixed effects estimates is controlled by the *prior.fe=1* setting.

```
% file ols_panel_gd2.m demo file2
clear all;
n = 200;
t = 10;

k = 2;
x = randn(n*t,k);
beta = ones(k,1);
sige = 0.1;
evec = randn(n*t,1)*sqrt(sige);

tts = (1:n)*(1/n);
SFE = kron(ones(t,1),tts');
ttt = (1:t)*(1/t);
TFE = kron(ttt',ones(n,1));

y = (x*beta + SFE + TFE + evec);

ndraw = 2500;
nomit = 500;
```

```
prior.novi_flag = 1; % homoscedastic model v_{it} = 1
prior.model = 3;     % model with fixed effects for regions and time period
prior.fe = 1;
result1 = ols_panel_FE_g(y,x,t,ndraw,nomit,prior);
vnames = strvcat('y','x1','x2');
prt_panel(result1,vnames);

tt = 1:n;
subplot(2,1,1),
plot(tt,0.5*result1.con+result1.sfe,'o',tt,tts,'+');
xlabel('N \times T observations');
ylabel('region-specific FE');
legend('estimated','true');
subplot(2,1,2),
tt=1:t;
plot(tt,0.5*result1.con+result1.tfe,'o',tt,ttt,'+');
xlabel('N \times T observations');
ylabel('time-specific FE');
legend('estimated','true');
```

The printed results are shown below for a set of only 10 of the $N = 200$ regions and all $T = 10$ time periods. There is also an overall intercept term associated with these effects, so the sum of all result1.sfe and result1.tfe are zero.

```
Mean intercept, region and time period fixed effects
Variable       Coefficient  Asymptot t-stat    z-probability
intercept         1.060598         9.573673         0.000000
sfe 1            -0.489430        -0.087507         0.930268
sfe 2            -0.498198        -0.048217         0.961543
sfe 3            -0.475671        -0.106177         0.915442
sfe 4            -0.491195        -0.635715         0.524962
sfe 5            -0.449434        -0.060930         0.951415
sfe 6            -0.498836        -0.054654         0.956414
sfe 7            -0.476688        -0.170468         0.864642
sfe 8            -0.436714        -0.065189         0.948024
sfe 9            -0.431982        -0.180108         0.857068
sfe 10           -0.569110        -0.145262         0.884504
tfe 1            -0.433066        -0.658605         0.510150
tfe 2            -0.342128        -0.426200         0.669962
tfe 3            -0.242013        -0.125944         0.899776
tfe 4            -0.158719        -0.197793         0.843207
tfe 5            -0.041293        -0.044229         0.964722
tfe 6             0.054029         0.208322         0.834978
tfe 7             0.157500         0.088794         0.929245
tfe 8             0.222402         0.139806         0.888813
tfe 9             0.330944         0.722128         0.470216
tfe 10            0.452344         0.314495         0.753145
```

A plot of the true spatial fixed effects (SFE) and true time fixed effects (TFE) is shown in Figure 1.2, where I have added one-half of the constant to the region-specific effects estimates and one-half of the constant to the time-specific effects. The estimated effects are close to the true effects used when generating the data in this illustration. The printed $t-$statistics and $t-$probabilities are from code in functions provided by J.P. Elhorst, and these seem incorrect to me. As a check of this I produced my own MCMC estimates of the standard deviation based on retained MCMC draws and found very significant effects estimates. This is in contrast to the reported $t-$statistics in printed results shown.

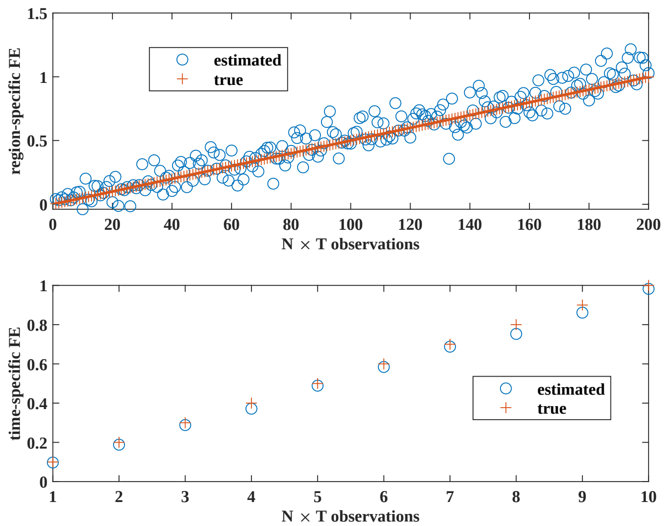

Figure 1.2: Region- and time-specific effects estimates

There is an option to add region- and time-labels to the printout. Given a string vector of region names and another for time period labels, use of: *prt_panel(results,vnames,se_names,te_names)* will add these names to the printout. There is also an option to print results to an output file, and both of these are illustrated in the code snippet below.

```
filename = 'output.txt';
snames = [];
for i=1:n
snames = strvcat(snames,['region' num2str(i)]);
end
tnames = [];
for i=1:t
tnames = strvcat(tnames,['time' num2str(i)]);
end
prt_panel(results,vnames,snames,tnames,filename);

Mean intercept, region and time period fixed effects
Variable       Coefficient  Asymptot t-stat    z-probability
intercept         1.060598         9.573673         0.000000
region1          -0.489430        -0.087507         0.930268
region2          -0.498198        -0.048217         0.961543
region3          -0.475671        -0.106177         0.915442
region4          -0.491195        -0.635715         0.524962
region5          -0.449434        -0.060930         0.951415
region6          -0.498836        -0.054654         0.956414
region7          -0.476688        -0.170468         0.864642
region8          -0.436714        -0.065189         0.948024
```

```
region9         -0.431982      -0.180108      0.857068
region10        -0.569110      -0.145262      0.884504
time1           -0.433066      -0.658605      0.510150
time2           -0.342128      -0.426200      0.669962
time3           -0.242013      -0.125944      0.899776
time4           -0.158719      -0.197793      0.843207
time5           -0.041293      -0.044229      0.964722
time6            0.054029       0.208322      0.834978
time7            0.157500       0.088794      0.929245
time8            0.222402       0.139806      0.888813
time9            0.330944       0.722128      0.470216
time10           0.452344       0.314495      0.753145
```

1.3 sar_panel_FE, sar_panel_FE_g()

The function *sar_panel_FE()* produces maximum likelihood estimates, while *sar_panel_FE_g()* uses MCMC estimation and allows Bayesian prior distributions to be assigned to the parameters, as well as the variance scalars v_{it} to be estimated. This is done as illustrated for the *ols_panel_FE_g()* function.

The maximum likelihood (ML) estimates are based on code from J.P. Elhorst, and these estimates should be replicated by MCMC estimation when no prior information is used. A point to note is that for poorly scaled sample data, ML estimates for the standard deviations coming from an analytical or numerical Hessian calculation may be inaccurate, whereas the MCMC estimates do not take this approach to producing estimates of dispersion for the parameters. If there are differences between the ML and MCMC estimates you should rely on the MCMC estimates as they will be more accurate.

In the case of the ML estimation function, options allow for: *info.model=0,1,2,3* to allow for various types of fixed effects choices, as in the case of the *ols_panel_FE_g()* function. There is also the option to printout fixed effects (*info.fe = 1*), estimates, and to use labels for the regions and time periods (if desired), as illustrated for the case of the *ols_panel_FE_g()* function.

The *sar_panel_FE_g()* function produces Markov Chain Monte Carlo (MCMC) estimates for a spatial autoregressive (SAR) model shown in (1.5), where y is an $N \times T$ vector of the dependent variable, organized so that all N regions for time period $t = 1$ are first, all N regions for time period $t = 2$, next, and so on. up to time $t = T$. The $NT \times k$ matrix X is organized in the same way. Again, we have incorporated fixed effects into the model, as in the case of the *ols_panel_FE_g()* function.

$$
\begin{aligned}
y &= \rho W y + X\beta + \iota_T \otimes \mu + \nu \otimes \iota_N + \varepsilon && (1.5)\\
\varepsilon &\sim N(0_{NT}, \sigma^2 V) \\
V &= \begin{pmatrix} v_1 & 0 & \dots & 0 \\ 0 & v_2 & \dots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & \dots & 0 & v_{NT} \end{pmatrix} \\
\pi(\beta) &\sim N(c, C) && (1.6)\\
\pi(r/v_i &\sim iid\ \chi^2(r), i = 1, \dots, N * T && (1.7)
\end{aligned}
$$

$$\pi(\sigma^2) \sim IG(a,b) \quad (1.8)$$
$$\pi(\rho) \sim U(1/\lambda_{min}, 1/\lambda_{max}) \quad (1.9)$$

One important difference from the ML estimation function is that the user can feed down a small $N \times N$ spatial weight matrix W to the function, or optionally, a larger $NT \times NT$ matrix W that allows for different weight matrices for each time period. The function will determine which type of W matrix has been input and do the appropriate thing. For example, if you use an $N \times N$ spatial weight matrix W, the function creates $I_T \times W$ for you, whereas if you use a larger $NT \times NT$ matrix W, the function uses this for estimation. An illustration of how to utilize this aspect of the function is provided later.

Some applied examples of using this model follows.

1.4 Using the *sar_panel_FE()*, *sar_panel_FE_g()* functions

The demo file below produces ML and MCMC estimates based on no prior distributions assigned to the parameters, to demonstrates that these two sets of estimates are nearly the same. The demo file also feeds down a small $N \times N$ weight matrix named W as well as a larger $NT \times NT$ matrix named $Wbig$ to the *sar_panel_FE_g()*. (This feature is not available for the ML estimation function.)

```
file: sar_panel_gd.m
clear all;
rng(10203040);
n = 200;
t = 10;

rho = 0.6;
k = 2;
x = randn(n*t,k);
beta = ones(k,1);
sige = 1;
evec = randn(n*t,1)*sqrt(sige);

latt = rand(n,1);
long = rand(n,1);

W = make_neighborsw(latt,long,5);

Wbig = kron(eye(t),W);

% add fixed effects to the DGP
tts = (1:n)*(1/n);
SFE = kron(ones(t,1),tts');
ttt = (1:t)*(1/t);
TFE = kron(ttt',ones(n,1));

y = (speye(n*t) - rho*Wbig)\(x*beta + SFE + TFE + evec);

prior.model = 3;
result1 = sar_panel_FE(y,x,W,t,prior);
vnames = strvcat('y','x1','x2');
```

```
prt_panel(result1,vnames);

ndraw = 2500;
nomit = 500;
prior2.novi_flag = 1;
prior2.model = 3;
result2 = sar_panel_FE_g(y,x,W,t,ndraw,nomit,prior2);
prt_panel(result2,vnames);

result3 = sar_panel_FE_g(y,x,Wbig,t,ndraw,nomit,prior2);
prt_panel(result3,vnames);
```

In the demonstration, the larger matrix $Wbig$ is no different from the smaller matrix W, since it was created using the MATLAB Kronecker product function: *Wbig = kron(eye(t),W);*, so we should see nearly identical estimates. MCMC estimates will vary *slightly* from different estimation runs, because they reflect simulation-based estimates. If there are large differences in estimates from two different estimation runs, this is an indication of problems with convergence of the MCMC sampler.

In the case of the demonstration file, all three sets of estimates are nearly the same. Note that the function automatically prints out direct, indirect and total effects estimates. The direct effects estimates are the scalar summary measures of own-partial derivatives, $\partial E(y_i)/\partial X_i^r$, associated with the rth explanatory variable, proposed by LeSage and Pace (2009). Reporting of SAR model estimates based on these scalar summaries has seen widespread use in the spatial econometrics literature. As LeSage and Pace (2009) point out the printed estimates for the coefficients β on the explanatory variables are not meaningful if one is interested in how changes in the explanatory variables impact the $y-$variable outcomes (which is typically the thing of interest for these models).

The indirect effects are scalar summary measures of the *cumulative* cross-partial derivatives, $\partial E(y_i)/\partial X_j^r$, where these are cumulated across all observations $j \neq i$, then averaged. These estimates are often referred to as a measure of cumulative spatial spillovers, i.e., the impact of changes in other-region explanatory variables on own-region outcomes, or what is the same thing, the impact of changes in own-region explanatory variables on other-region outcomes.

```
% output from: sar_panel_gd.m file
Homoscedastic model
MaxLike SAR model with both region and time period fixed effects
Dependent Variable =              y
R-squared           =    0.8247
corr-squared        =    0.6762
sigma^2             =    0.8694
Nobs,Nvar,#FE       =   2000,     3,    12
log-likelihood      =       -2794.2143
# of iterations     =     20
min and max rho     =   -1.0000,   1.0000
total time in secs  =    0.1820
time for lndet      =    0.0160
time for MCMC draws =    0.0310
Pace and Barry, 1999 MC lndet approximation used
order for MC appr   =     50
iter  for MC appr   =     30
***************************************************************
Variable      Coefficient  Asymptot t-stat    z-probability
x1              0.984479       44.010559         0.000000
```

```
x2              1.009949       46.059889         0.000000
rho             0.598993       40.351442         0.000000

Direct       Coefficient          t-stat           t-prob         lower 05        upper 95
x1              1.078746       43.002838         0.000000         1.125375        1.030078
x2              1.106949       45.300136         0.000000         1.153570        1.056558

Indirect     Coefficient          t-stat           t-prob         lower 05        upper 95
x1              1.380097       15.569507         0.000000         1.566430        1.213571
x2              1.416136       15.836093         0.000000         1.605164        1.250592

Total        Coefficient          t-stat           t-prob         lower 05        upper 95
x1              2.458843       23.559215         0.000000         2.678459        2.270484
x2              2.523085       24.223048         0.000000         2.732605        2.321244

Homoscedastic model
MCMC SAR model with both region and time period fixed effects
Dependent Variable =               y
R-squared           =    0.8243
corr-squared        =    0.6760
sigma^2             =    0.8742
Nobs,Nvar,#FE       =   2000,     2,   210
ndraw,nomit         =   2500,   500
rvalue              =      0
min and max rho     =   -1.0000,   1.0000
total time in secs  =    1.2670
time for lndet      =    0.0210
time for MCMC draws =    1.0850
Pace and Barry, 1999 MC lndet approximation used
order for MC appr  =     50
iter  for MC appr  =     30
***************************************************************
Variable      Coefficient  Asymptot t-stat    z-probability
x1              0.985475       43.222595         0.000000
x2              1.010456       45.690178         0.000000
rho             0.593735       40.230558         0.000000

Direct       Coefficient          t-stat           t-prob         lower 05        upper 95
x1              1.078109       41.743353         0.000000         1.128047        1.027713
x2              1.105439       43.897214         0.000000         1.156031        1.056973

Indirect     Coefficient          t-stat           t-prob         lower 05        upper 95
x1              1.350795       15.430958         0.000000         1.527901        1.191169
x2              1.385044       15.498397         0.000000         1.569000        1.224448

Total        Coefficient          t-stat           t-prob         lower 05        upper 95
x1              2.428904       23.175083         0.000000         2.640351        2.234755
x2              2.490483       23.467536         0.000000         2.705577        2.292809

Homoscedastic model
MCMC SAR model with both region and time period fixed effects
Dependent Variable =               y
R-squared           =    0.8244
corr-squared        =    0.6760
sigma^2             =    0.8726
```

```
Nobs,Nvar,#FE       =   2000,     2,   210
ndraw,nomit         =   2500,   500
rvalue              =      0
min and max rho     =   -1.0000,   1.0000
total time in secs  =    1.0350
time for lndet      =    0.0200
time for MCMC draws =    0.8640
Pace and Barry, 1999 MC lndet approximation used
order for MC appr  =     50
iter  for MC appr  =     30
***************************************************************
Variable      Coefficient  Asymptot t-stat    z-probability
x1               0.984530       43.722267         0.000000
x2               1.010846       46.704967         0.000000
rho              0.595268       40.135865         0.000000

Direct        Coefficient           t-stat           t-prob        lower 05        upper 95
x1               1.077774       41.992070         0.000000        1.127829        1.027485
x2               1.106584       44.473910         0.000000        1.156157        1.058523

Indirect      Coefficient           t-stat           t-prob        lower 05        upper 95
x1               1.358074       15.314359         0.000000        1.536980        1.189010
x2               1.394397       15.369981         0.000000        1.577469        1.221030

Total         Coefficient           t-stat           t-prob        lower 05        upper 95
x1               2.435848       22.972530         0.000000        2.647108        2.228955
x2               2.500981       23.250743         0.000000        2.713614        2.295248
```

All of the functions in the Toolbox have documentation that can be displayed in the MATLAB command window, using the command *help sar_panel_FE_g*. The printed documentation is shown below, where both usage information regarding the input options from the structure variable *prior* are described as well as documentation for the fields returned in the *results* structure variable. The estimation function returns MCMC draws for the parameters β, ρ, σ^2 as well as MCMC draws for the *direct,indirect and total effects* estimates. In addition to returning MCMC draws, the function also returns an $nvar \times 5$ matrix containing posterior means for the *effects* estimates along with calculated $t-$statistics, $t-$probabilities, and lower 0.05 and upper 0.95 credible intervals. These statistics for the scalar summary effects estimates are calculated using the retained (ndraw-nomit) MCMC draws. For example, the mean of the draws for the direct effects (for each explanatory variable) and standard deviation of the draws is used to construct a $t-$statistic, which is used to find the associated $t-$probability. The lower 0.05 and upper 0.95 intervals are also based on the MCMC draws. Given a set of 10,000 MCMC draws, the lower 0.05 interval would be determined by the 500th value from the lowest set of sorted values, and the upper 0.95 interval by the 9,500th value from this sorted set.

```
>> help sar_panel_FE_g
  PURPOSE: MCMC SAR model estimates for static spatial panels
           (N regions*T time periods) with spatial fixed effects (sfe)
           and/or time period fixed effects (tfe)
           y = rho*W*y + X*b + sfe(optional) + tfe(optional) + e,
           e = N(0,sige*V),
           V = diag(v_1,v_2,...v_N*T), r/vi = ID chi(r)/r, r = 5 (default)
           b   = N(c,C), default c = 0, C = eye(k)*1e+12
```

```
        sige = gamma(nu,d0), default nu=0, d0=0
        no prior for rho
Supply data sorted first by time and then by spatial units, so first region 1,
region 2, et cetera, in the first year, then region 1, region 2, et
cetera in the second year, and so on
sar_panel_FE_g transforms y and x to deviation of the spatial and/or time means
---------------------------------------------------
 USAGE: results = sar_panel_FE_g(y,x,W,T,ndraw,nomit,prior)
 where:  y = N*T x 1 dependent variable vector
        x = N*T x k independent variables matrix
        W = spatial weights matrix (standardized)
        N. B. W-matrix can be N*T x N*T or N x N
        T = number of points in time
     prior = a structure variable with input options:
     prior.novi_flag = 1, for e = N(0,sige*I), homoscedastic model
                     = 0, for e = N(0,sige*V), heteroscedastic model
                          sets V = diag(v_1,v_2,...v_N*T), rval/vi = ID chi(rval)/rval, rval = 5 (default)
     prior.rval = rval, r prior hyperparameter, default=5
     prior.model = 0 pooled model without fixed effects (default, x may contain an intercept)
                 = 1 spatial fixed effects (x may not contain an intercept)
                 = 2 time period fixed effects (x may not contain an intercept)
                 = 3 spatial and time period fixed effects (x may not contain an intercept)
     prior.fe    = report fixed effects and their t-values in prt_panel
                   (default=0=not reported; prior.fe=1=report)
             prior.beta, prior means for beta,   b (default (k x 1) vector = 0)
             priov.bcov, prior beta covariance, T above (default eye(k)*1e+12)
             prior.rval, rval prior hyperparameter, default=4
             prior.nu,   informative Gamma(nu,d0) prior on sige
             prior.d0    informative Gamma(nu,d0) prior on sige
                         default for above: nu=0,d0=0 (diffuse prior)
     prior.rmin  = (optional) minimum value of rho to use in search
     prior.rmax  = (optional) maximum value of rho to use in search
     prior.lflag = 0 for full lndet computation (default = 1, fastest)
                 = 1 for MC lndet approximation (fast for very large problems)
                 = 2 for Spline lndet approximation (medium speed)
     prior.order = order to use with info.lflag = 1 option (default = 50)
     prior.iter  = iterations to use with info.lflag = 1 option (default = 30)
     prior.lndet = a matrix returned in results.lndet containing log-determinant information to save time
---------------------------------------------------
 RETURNS: a structure
       results.meth  = 'sarsfe_g' if prior.model=1
                     = 'sartfe_g' if prior.model=2
                     = 'sarstfe_g' if prior.model=3
       results.beta  = bhat
       results.rho   = rho
       results.bdraw = (ndraw-nomit)xk matrix of MCMC draws for beta
       results.pdraw = (ndraw-nomit)x1 vector of MCMC draws for rho
       results.vmean = N*T x 1 vector of v_{it} means
       results.sdraw = (ndraw-nomit)x1 vector of MCMC draws for sige
       results.bmean = b prior means (prior.beta from input)
       results.bstd  = b prior std deviation, sqrt(diag(prior.bcov))
       results.nu    = prior nu-value for sige prior (default = 0)
       results.d0    = prior d0-value for sige prior (default = 0)
       results.iprior = 1 for informative prior on beta,
                      = 0 for default no prior on beta
```

```
        results.direct   = nvar x 5 matrix with direct effect, t-stat, t-prob, lower05, upper95
        results.indirect = nvar x 5 matrix with indirect effect, t-stat, t-prob, lower05, upper95
        results.total    = nvar x 5 matrix with total effect, t-stat, t-prob, lower05, upper95
        results.direct_draws   = ndraw x nvar matrix of direct effect draws
        results.indirect_draws = ndraw x nvar matrix of indirect effect draws
        results.total_draws    = ndraw x nvar matrix of total effect draws
        results.cov   = asymptotic variance-covariance matrix of the parameters b(eta) and rho
        results.tstat = asymp t-stat (last entry is rho=spatial autoregressive coefficient)
        results.yhat  = [inv(y-p*W)]*[x*b+fixed effects] (according to prediction formula)
        results.resid = y-p*W*y-x*b
        results.sige  = (y-p*W*y-x*b)'*(y-p*W*y-x*b)/n
        results.rsqr  = rsquared
        results.corr2 = goodness-of-fit between actual and fitted values
        results.sfe   = spatial fixed effects (if prior.model=1 or 3)
        results.tfe   = time period fixed effects (if prior.model=2 or 3)
        results.tsfe  = t-values spatial fixed effects (if prior.model=1 or 3)
        results.ttfe  = t-values time period fixed effects (if prior.model=2 or 3)
        results.con   = intercept
        results.con   = t-value intercept
        results.lik   = log likelihood
        results.nobs  = # of observations
        results.nvar  = # of explanatory variables in x
        results.tnvar = # fixed effects
        results.iter  = # of iterations taken
        results.rmax  = 1/max eigenvalue of W (or rmax if input)
        results.rmin  = 1/min eigenvalue of W (or rmin if input)
        results.lflag = lflag from input
        results.fe    = fe from input
        results.liter = info.iter option from input
        results.order = info.order option from input
        results.limit = matrix of [rho lower95,logdet approx, upper95] intervals
                        for the case of lflag = 1
        results.time1 = time for log determinant calcluation
        results.time2 = time for eigenvalue calculation
        results.time4 = time for MCMC sampling
        results.time  = total time taken
        results.lndet = a matrix containing log-determinant information
                         (for use in later function calls to save time)
--------------------------------------------------
 NOTES: if you use lflag = 1 or 2, info.rmin will be set = -1
                                   info.rmax will be set = 1
        For number of spatial units < 500 you should use lflag = 0 to get
        exact results,
        Fixed effects and their t-values are calculated as the deviation
        from the mean intercept
---------------------------------------------------
```

Turning to use of the feature that allows for different spatial weight matrices for each time period of our panel data, we illustrate this in the following demonstration program. This demonstration is meant to serve as a warning regarding issues that can arise when allowing for different spatial weight matrices during each time period in your model formulation.

The demonstration program forms two weight matrices, one based on 10 nearest neighbors and another based on only 3 nearest neighbors. These are constructed from the same random latitude-longitude vectors, with the *make_neighborsw()* function from my econometrics toolbox. Another function from my toolbox *blockdiag()* is used to place these two $W-$matrices on the diagonal of a larger matrix named *Wtime* that is consistent with the 6 time periods for the panel data set. The two $W-$matrices are alternated each time period, so we have a matrix *Wtime* that allows for 10 neighbors during time period 1, 3 neighbors during time period 2, and so on for all six time periods.

The estimation function is then called with this time-varying set of matrices in *Wtime* to produce estimates, for a $y-$variable generated using this type of spatial weight matrix.

The default behavior of the *prt_panel* function is to printout scalar summary estimates for the coefficient estimates along with the direct, indirect and total effects estimates.

The program also calculates *observation-level effects* estimates based on the true parameters β, ρ rather than the *scalar summary effects* estimates. LeSage and Pace (2009) argue that we can summarize the $N \times N$ matrix of partial derivatives (in the case of a cross-sectional model) using an average of the main diagonal elements from this matrix of derivatives to produce a *scalar* summary measure of the own-partial derivatives (direct effects). They also argue that we can use the average of the cumulative off-diagonal elements from the $N \times N$ matrix of partial derivatives to calculate a *scalar* summary estimate of the cross-partial derivatives (indirect effects).

Elhorst (2013) argues that in a *static panel data model* context, where all NT data observations are used to produce a single set of estimates for the parameters β, ρ, and the small $N \times N$ matrix W is repeated for each time period, the scalar summary effects calculations from LeSage and Pace (2009) are valid. The observation-level partial derivatives take the form of an $NT \times NT$ matrix $(I_{NT} - \rho(I_T \otimes W))^{-1} I_{NT}\beta$, in this case where the matrix W does not change over time periods.

However, in our case where the matrix *Wtime* does change over time periods, we should examine the observation-level effects in the $NT \times NT$ matrix $(I_{NT} - \rho Wtime)^{-1} I_{NT}\beta$, and compare these to the scalar summary estimates, based on the LeSage and Pace (2009) proposal to use the average of the main diagonal for direct effects and the average of the cumulative off-diagonal elements for indirect effects.

The code snippet below calculates scalar summary estimates (for variable x_1 only) in the fashion proposed by LeSage and Pace (2009) as well as the observation-level effects, which are constructed from the total effects based on a sum of all NT rows, the direct effects based on the diagonal elements of the $NT \times NT$ matrix of partial derivatives, and the indirect effects using the difference between these two $NT \times 1$ vectors, e.g., indirect = total - direct effects.

```
% code snippet (taken from sar_panel_gd2.m file
% calculate true observation-level effects estimates (for x1 variable only)
% and scalar summary effects estimates
V = (speye(n*t) - rho*Wtime)\speye(n*t);
bmat  = eye(n*t)*beta(1);          % (for x1 variable only)
total = sum(V*bmat,2);             % Observation-level true values
direct = diag(V*bmat);             % Observation-level true values
indirect = total - direct;         % Observation-level true values
direct_mean = mean(direct);        % Scalar summary true values
indirect_mean = mean(indirect);    % Scalar summary true values
total_mean = mean(total);          % Scalar summary true values
[direct_mean indirect_mean total_mean] % displays the true scalar summaries
```

```
% sar_panel_gd2 demo file
clear all;
rng(10203444);
```

```
n = 200;
t = 6;

rho = 0.7;
k = 2;
x = randn(n*t,k);
beta = ones(k,1);
sige = 1;
evec = randn(n*t,1)*sqrt(sige);

latt = rand(n,1);
long = rand(n,1);

W1 = make_neighborsw(latt,long,10);
W2 = make_neighborsw(latt,long,3);

Wtime = blockdiag(W1,W2,W1,W2,W1,W2);

% add fixed effects to the DGP
tts = (1:n)*(1/n);
SFE = kron(ones(t,1),tts');
ttt = (1:t)*(1/t);
TFE = kron(ttt',ones(n,1));

y = (speye(n*t) - rho*Wtime)\(x*beta + SFE + TFE + evec);

% calculate true observation-level effects estimates (for x2 variable only)
% and scalar summary effects estimates
V = (speye(n*t) - rho*Wtime)\speye(n*t);
bmat  = eye(n*t)*beta(1);      % (for x1 variable only)
total = sum(V*bmat,2);         % Observation-level true values
direct = diag(V*bmat);         % Observation-level true values
indirect = total - direct;     % Observation-level true values
direct_mean = mean(direct);    % Scalar summary true values
indirect_mean = mean(indirect);% Scalar summary true values
total_mean = mean(total);      % Scalar summary true values

[direct_mean indirect_mean total_mean]

ndraw = 2500;
nomit = 500;
info.novi_flag = 1;
info.model = 3;

result1 = sar_panel_FE_g(y,x,Wtime,t,ndraw,nomit,info);
prt_panel(result1);
% posterior mean estimates based on MCMC draws
rho = result1.rho;
beta = result1.beta(1,1);

V = (speye(n*t) - rho*Wtime)\speye(n*t);
bmat  = eye(n*t)*beta(1);       (for x1 variable only)
total2 = sum(V*bmat,2);         % Observation-level estimates
direct2 = diag(V*bmat);         % Observation-level estimates
indirect2 = total2 - direct2;   % Observation-level estimates
```

```
direct_mean = mean(direct2);     % scalar summary estimates
indirect_mean = mean(indirect2); % scalar summary estimates
total_mean = mean(total2);       % scalar summary estimates

tt=1:n*t;
subplot(2,1,1),
plot(tt,direct,'.b',tt,direct2,'.r');
legend('true','estimate');
xlabel('observation-level direct effects');
subplot(2,1,2),
plot(tt,indirect,'.b',tt,indirect2,'.r');
legend('true','estimate');
xlabel('observation-level indirect effects');
```

The estimation results presented below do a reasonable job of estimating the scalar summary effects estimates calculated based on the true parameter values β, ρ as indicated in the printout. The true values for the scalar summary estimates of direct, indirect and total are shown to be: 1.166, 2.166, and 3.333, respectively. The estimated scalar summary estimates are shown as having corresponding values of 1.185, 2.112, and 3.298. (Of course, since the true parameter for β_2 associated with the variable x_2 the direct,indirect and total effects for this variable should equal those for the x_1 variable.)

```
>> output from: sar_panel_gd2
% true scalar summary direct, indirect, total effects
direct    indirect  total
1.1669    2.1664    3.3333

Homoscedastic model
MCMC SAR model with both region and time period fixed effects
R-squared          =    0.8774
corr-squared       =    0.6608
sigma^2            =    0.9189
Nobs,Nvar,#FE      =   1200,     2,   206
ndraw,nomit        =   2500,   500
rvalue             =      0
min and max rho    =   -1.0000,   1.0000
total time in secs =    1.2150
time for lndet     =    0.0200
time for MCMC draws =    1.0140
Pace and Barry, 1999 MC lndet approximation used
order for MC appr  =     50
iter  for MC appr  =     30
***************************************************************
Variable        Coefficient  Asymptot t-stat    z-probability
variable 1         1.022800       32.036979         0.000000
variable 2         0.994902       34.254267         0.000000
rho                0.689075       43.819695         0.000000

Direct          Coefficient           t-stat           t-prob         lower 05         upper 95
variable 1         1.185391       29.989935         0.000000         1.264369         1.108712
variable 2         1.153038       32.377203         0.000000         1.222753         1.083251

Indirect        Coefficient           t-stat           t-prob         lower 05         upper 95
variable 1         2.112722       12.454978         0.000000         2.454282         1.806681
```

```
variable 2         2.054856        12.851577         0.000000         2.373588         1.751417

Total            Coefficient         t-stat           t-prob          lower 05         upper 95
variable 1         3.298113        16.579918         0.000000         3.702934         2.938570
variable 2         3.207894        17.315650         0.000000         3.571280         2.848126
```

A plot of the true versus estimated observation-level direct and indirect effects is shown in Figure 1.3, where we see a clear impact of the changes in the matrix W taking place over time. The change from 10 to 3 nearest neighbors over the 6 time periods produce observation-level effects estimates that vary to reflect these differences. The spillover (indirect) effects will be larger for the time periods where we have 10 neighbors and smaller for those where there are only 3 neighbors.

The scalar summary estimates average over the set of low and high values for the observation-level estimates, which is what the method proposed by LeSage and Pace (2009) should do. The question we have to ask is — would the user draw valid conclusions regarding the partial derivative impacts arising from changes in the x_1-variable on the dependent variable by looking *only* at the printed scalar summary estimates? Relying on the scalar summary estimates would clearly obscure important variation over time in the direct and indirect effects that might be of substantive interest in the spatial regression relationship being analyzed.

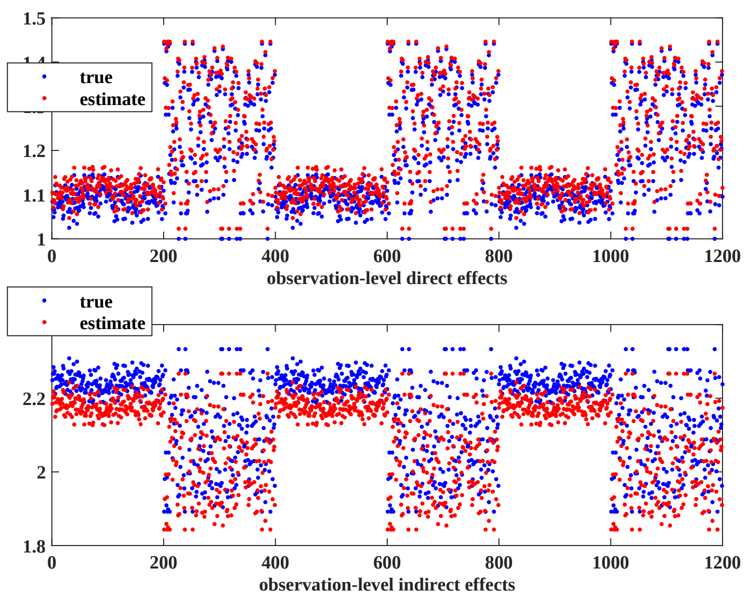

Figure 1.3: Observation-level effects estimates for x1 variable

Another point illustrated by the plot of the observation-level direct and indirect effects is that small estimation errors for the coefficients β, ρ will be magnified by the non-linear nature of the partial derivatives, which involves the infinite series expansion : $I_{NT} - \rho Wtime)^{-1} = I_{NT} + \rho Wtime + \rho^2 Wtime^2 + \ldots$. The magnification is more apparent for the indirect effects which rely on cumulating all off-diagonal elements from the matrix powers: $Wtime, Wtime^2, Wtime^3$ etc.

Given that we have MCMC draws for the parameters β, ρ, we could use these draws to produce an estimate of the standard deviation or the 0.05 and 0.95 quantiles of the observation-level estimates shown in Figure 1.3. This would allow us to answer the question — is there a significant different in the effects estimates for time periods based on 10 versus 3 nearest neighbors in the weight matrix.

The code snippet below from the *sar_panel_gd3.m* file does this. We loop over the ndraw-nomit = 2000 MCMC draws and use all draws for β, ρ to calculate a series of $2000 \times n * t$ direct and indirect effects estimates. Given these, we use the *plims()* function from my toolbox to find the lower 0.05, upper 0.95 and the 0.5 quantiles.

```
% code snippet (taken from sar_panel_gd3.m file
% calculate mean and 0.05, 0.95 intervals for the effects estimates
% using 2000 MCMC draws

total2 = zeros(ndraw-nomit,n*t);
direct2 = zeros(ndraw-nomit,n*t);
indirect2 = zeros(ndraw-nomit,n*t);


for iter=1:ndraw-nomit
rho = result1.pdraw(iter,1);
beta = result1.bdraw(iter,1);

V = (speye(n*t) - rho*Wtime)\speye(n*t);
bmat  = eye(n*t)*beta(1);
total2(iter,:) = (sum(V*bmat,2))';
direct2(iter,:) = diag(V*bmat);
indirect2(iter,:) = (total2(iter,:) - direct2(iter,:))';
end

direct_int = plims(direct2);
indirect_int = plims(indirect2);
% plims returns an n*t x 5 matrix with
% p quantiles from columns of direct2
% p =[0.005, 0.025, 0.5, 0.975, 0.995];

tt=1:2*n;
subplot(2,1,1),
plot(tt,direct(tt,1),'.b',tt,direct_int(tt,3),'.r',tt,direct_int(tt,2),'-g',tt,direct_int(tt,4),'-g');
legend('true','estimate','upper0.95','lower0.05');
xlabel('observation-level direct effects');
subplot(2,1,2),
plot(tt,indirect(tt,1),'.b',tt,indirect_int(tt,3),'.r',tt,indirect_int(tt,2),'-g',tt,indirect_int(tt,4),'-g');
legend('true','estimate','upper0.95','lower0.05');
xlabel('observation-level indirect effects');
```

Figure 1.4 shows a plot of the direct and indirect medians along with the lower 0.05 and upper 0.95 intervals, for only the first 400 observations, representing the first two time periods. This was done to provide a clearer view of the results.

From the figure we see that there are statistically significant differences between both the direct and indirect effects estimates from time period 1 (based on a weight matrix with 10 neighbors) and time period 2 (where the weight matrix was based on 3 neighbors). We draw this conclusion from

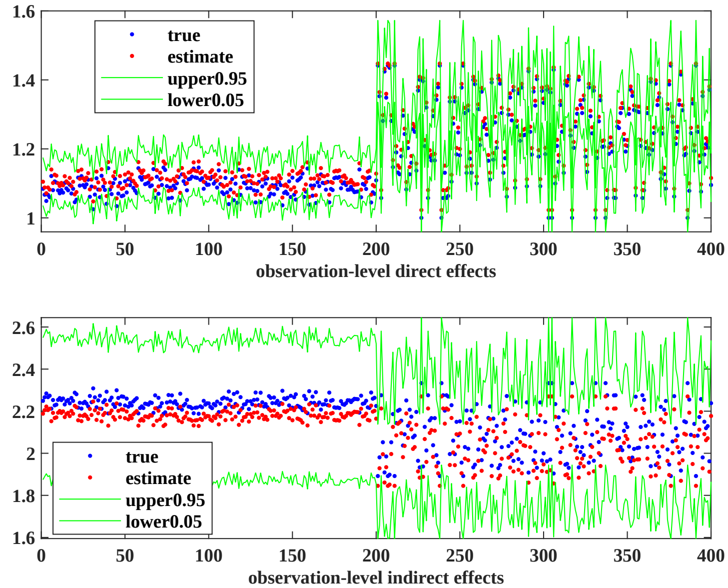

Figure 1.4: Observation-level effects with lower 0.05 and upper 0.95 limits

the fact that the lower 0.05 interval for the second time period direct effects lies above the upper 0.95 interval for the first time period for may of the direct effects. It is also the case that for the indirect effects, we see an upper 0.95 interval for the second time period that lies above many of the indirect effects estimates from the first time period.

The results of this analysis indicate that we should not rely on scalar summary effects estimates that remain the same over all time periods and regions because there is significant variation in the direct and indirect effects over time periods that arise from changes in the spatial weight matrix over time periods. What would represent a valid way to proceed given the results presented here? We should probably present *two* sets of scalar summary measures. One would be based on averaging the *observation-level* effects over years 1, 3, 5 to produce one set of scalar summary estimates for the direct, indirect, and total effects. The second set of scalar summary estimates would take the same approach, averaging the *observation-level* effects over years 2, 4, 6. These two sets of estimates would clearly provide inferences regarding the relationship in which we are interested that is closer to the true nature of the estimation results and what they imply about the relationship.

1.5 Chapter summary

We outlined two of the model estimation functions contained in the panel data portion of the Toolbox. In addition to mathematical descriptions of these model specifications, illustrations for use of functions to produce parameter estimates for the ordinary regression (OLS) and spatial autoregressive model (SAR) were provided.

The input options for selecting the time of fixed effects to be used, choice of robust/heteroscedastic or homoscedastic estimates, assigning Bayesian normal prior means and variances or variance-covariance structure for the parameters β, assigning inverse-gamma prior parameters for the noise variance parameter σ^2 are harmonized over all functions in the toolbox.

I recommend use of the robust/heteroscedastic estimates are this approach to estimation can avoid issues that arise from outliers or heteroscedastic/non-constant variance over both regions and time periods. This is something that maximum likelihood estimates cannot do.

We also illustrated a feature of the MCMC estimation function for the SAR model (*sar_panel_FE_g()* that allows users to input spatial weight matrices that varying with each time period in the panel data set. Reviewers of my applied work on spatial regression models frequently raise this criticism of *static panel* spatial regression models. However, reviewers may be naive about the potential challenges that arise from taking this approach to model specification.

To illustrate these issues, we showed how a set of MCMC draws can be used to produce observation-level effects estimates that can be compared to the scalar summary estimates for the direct, indirect and total effects printed out by the toolbox *prt_panel()* function using the MATLAB structure variable returned by the estimation functions. The name of the structure variable can be anything you wish, e.g.,

```
president_trump = sar_panel_FE_g(y,x,W,ndraw,nomit,prior);
prt_panel(president_trump)
```

would work. It probably makes sense to pick intuitive names like *results1, results2*, etc. The same is true for the structure variable inputs, for example, in place of my use of a structure variable named *prior*, note that I used:

```
info.novi_flag = 1;
info.model = 3;
result1 = sar_panel_FE_g(y,x,Wtime,t,ndraw,nomit,info);
```

in the example above.

Discussion of more spatial regression models that can be estimated using the panel data part of the Toolbox continues in Chapter 2.

1.6 Chapter references

Geweke, J. (1993). Bayesian Treatment of the Independent Student t Linear Model, *Journal of Applied Econometrics*, 8, 19-40.

LeSage, J.P. and R.K. Pace (2009) *Introduction to Spatial Econometrics*, Taylor & Francis, CRC Press.

Lange, K.L., R.J.A. Little and J.M.G. Taylor (1989). Robust Statistical Modeling Using the t Distribution, *Journal of the American Statistical Association*, 84, 881-896.

Chapter 2

The SDM, SEM, SDEM, SLX models

This chapter discusses more estimation functions available in the Toolbox from both a mathematical and applied perspective.

2.1 The SDM model

The spatial Durbin (SDM) model is shown in (2.1), where it should be clear that this model extends the SAR specification to include *spatial lags* of the explanatory variables, which consists of the $NT \times K$ matrix WX, with associated $K \times 1$ vector of parameters θ.

$$
\begin{aligned}
y &= \rho W y + X\beta + WX\theta + \iota_T \otimes \mu + \nu \otimes \iota_N + \varepsilon && (2.1)\\
\varepsilon &\sim N(0_{NT}, \sigma^2 V) \\
V &= \begin{pmatrix} v_1 & 0 & \dots & 0 \\ 0 & v_2 & \dots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & \dots & 0 & v_{NT} \end{pmatrix} \\
\pi(\beta) &\sim N(c, C) && (2.2)\\
\pi(r/v_i) &\sim iid\ \chi^2(r), i = 1, \dots, N*T && (2.3)\\
\pi(\sigma^2) &\sim IG(a, b) && (2.4)\\
\pi(\rho) &\sim U(1/\lambda_{min}, 1/\lambda_{max}) && (2.5)
\end{aligned}
$$

In addition to a function *sdm_panel_FE_g()* that produces estimates for this model with MCMC and allowing for use of Bayesian prior distributions, heterocedastic/outlier robust estimates based on the variance scalars v_{it}, there is also a maximum likelihood estimation function *sdm_panel_FE()*. The addition of the extension *_g* to designate MCMC estimation comes from use of the term *Gibbs sampling* in the Bayesian MCMC literature. This term has been used to refer to a special type of MCMC sampling, where the conditional distributions for all of the model parameters take the form of *known* distributions, e.g., a normal, inverse-gamma, chi-squared, etc. This is not technically true of our MCMC sampler which relies on a Metropolis-Hastings or draw-by-inversion approach to sampling for the spatial dependence parameter ρ. Nonetheless, I use the file naming convention *_g* to designate an MCMC estimation function.

As in the case of the SAR model function, the MCMC estimation function allows the user to input either a small $N \times N$ spatial weight matrix W to the function, or optionally, a larger $NT \times NT$ matrix W that allows for different weight matrices for each time period. The function will determine which type of W matrix has been input and do the appropriate thing. For example, if you use an $N \times N$ spatial weight matrix W, the function creates $I_T \times W$ for you, whereas if you use a larger $NT \times NT$ matrix W, the function uses this for estimation.

The reader might wonder why we need a separate function for estimating the SDM model, when it would see we could use the SAR model function and just input a modified matrix $\tilde{X} = \begin{pmatrix} X & WX \end{pmatrix}$. The answer is that the partial derivatives used to calculate the direct and indirect effects estimates for this model are different from those of the SAR model specification. Specifically, they take the form shown of an $NT \times NT$ matrix in (2.12).

$$\partial E(y)/\partial X^{r\prime} \quad = \quad (I_{NT} - \hat{\rho} W)^{-1}(\hat{\beta} + W\hat{\theta}) \tag{2.6}$$

A call to the SAR MCMC (or ML) estimation function with the matrix $\tilde{X}$ will result in the function printing out the wrong direct, indirect and total effects estimates.

Some applied examples of using the SDM model follows.

2.1.1 Using the *sdm_panel_FE()*, *sdm_panel_FE_g()* functions

The program below illustrates ML and MCMC estimation of an SDM model, where the reader should note that we enter *only* the matrix X of explanatory variables, letting the function create the *spatial lag* of the explanatory variables matrix.

A point to note is that if you wish to apply a normal prior distribution to the coefficients β for this model, you need to input a $2k \times 1$ vector of prior means for both β and θ, the coefficients associated with both the $X-$ and WX-variables. Similarly, the variance-covariance matrix for the multivariate normal prior distribution must be a $2k \times 2k$ matrix, where k is the number of explanatory variables in the matrix X.

The empirical model estimated uses a panel of 48 US states for 51 weeks during the year 2020, starting during the second week of 2020. The dependent variable $y = uclaims$ is the annual growth rate of continuing claims for unemployment insurance, calculated using $log(uclaims2020) - log(uclaims2019)$, where the same weeks during 2020 and 2019 was used. (This data is from the US Department of Labor). The two explanatory variables are job openings and a measure of social distancing reflecting the median percentage of population spending time at home. The job posting data comes from the Opportunity Insights Economic Tracker, a large public database of 40,000 online job boards, and reflects an index that measures changes each week during 2020 relative to the weeks during January 2020, prior to the Covid-19 pandemic. The time at home information comes from the Bureau of Transportation Statistics (BTS) daily survey from 2019 and 2020, which is based on a national panel of anonymous mobile devices, with weights applied to the panel to create estimates for the statewide number of people staying at home versus not staying at home.

The program reads the data from 3 different sheets of a single Excel file, using the MATLAB *xlsread()* function. The first sheet has columns with the 48 state names (two-letter postal codes) and simple row-labels 'week1', 'week2', etc.

These row- and column-labels are converted to string vectors, one containing the state names and the other the time-period names, using the *strvcat()* function from my toolbox. We need to start in row 2, column 1 of the matrix b that contains row- and column-labels read from the Excel file to skip the header label. Similarly, we need to start in row 1, column 2 when creating the time-period labels to avoid this header label.

The input data represent three $N \times T$ matrices, which are converted to vectors using the MATLAB *vec()* function.

Another Excel file contains a 48×48 binary contiguity matrix for the 48 states, with values of 1 for neighboring states (those with borders touching) and zeros otherwise. The function from my toolbox *normw()* is used to produce a row-normalized spatial weight matrix, where each row has values that sum to one.

The function produces: 1) a set of ML estimates, 2) a set of MCMC estimates for the homoscedastic model, and 3) a set of MCMC estimates for the heteroscedastic model, all of which should be reasonably similar. In the face of homoscedastic disturbances for our sample data generated model, the variance scalars v_{it} should be fairly close to one, which will not impact the estimates. On the other hand, if there is a heteroscedasticity problems, the robust model estimates based on assigning a value of $r = 5$ to the chi-squared prior for the v_{it} scalars should properly adjust the estimates to accommodate this type of problem. This suggests that we should routinely use this type of prior for MCMC estimation.

```
% file: sdm_panel_gd.m demo file
clear all;
[uclaims,b] = xlsread('../demo_data/weekly.xlsx',1);
% read data from sheet 1 of Excel spreadsheet
% growth rate of unemployment 2019-2020 from same week, previous year
snames = strvcat(b(2:end,1)); % 48 state names
tnames = strvcat(b(1,2:end)); % 51 week labels
[N,T] = size(uclaims);
[jobposts,b] = xlsread('../demo_data/weekly.xlsx',2);
% read data from sheet 2 of Excel spreadsheet
% change in job offers from 1st week of 2020
[athome,b] = xlsread('../demo_data/weekly.xlsx',3);
% read data from sheet 3 of Excel spreadsheet
% growth rate of percent population at home
% 2019-2020 from same week, previous year
[a,b] = xlsread('../demo_data/Wcont48.xlsx');
% 48 x 48 contiguity matrix for states
W = normw(a);

y = vec(uclaims);
x = [vec(jobposts) vec(athome)];

info.model = 3;
result1 = sdm_panel_FE(y,x,W,T,info);
vnames = strvcat('uclaims','jobposts','athome');
prt_panel(result1,vnames);

ndraw = 2500;
nomit = 500;
prior.novi_flag = 1;
prior.model = 3;
result2 = sdm_panel_FE_g(y,x,W,T,ndraw,nomit,prior);
prt_panel(result2,vnames);
```

```
prior2.rval = 5;
prior2.model = 3;
prior2.fe = 1;
result3 = sdm_panel_FE_g(y,x,W,T,ndraw,nomit,prior2);
prt_panel(result3,vnames,snames,tnames);

tt=1:T;
plot(tt,result3.tfe);
xlabel('weeks during 2020');
ylabel('time fixed effects estimates');

vmat = reshape(result3.vmean,N,T);
vmean = mean(vmat,1);
plot(tt,vmean,'o');
xlabel('Weeks during 2020');
ylabel('Mean of time v_{it} estimates');
```

The program uses the state names and time-period names when printing out the estimates, which is set using the option *prior2.fe=1*, and the region-specific variable names, *snames*, and time-specific labels, *tnames* when calling the *prt_panel()* function.

The program plots the time-specific effects for the 51 weeks, as this might reflect an interesting source of variation during the pandemic weeks of lock down. Using *help sdm_panel_FE_g* would show that these fixed effects estimates are returned in the structure variable field *.tfe*.

The program also recovers the posterior mean v_{it} estimates from the *result3* structure variable field *.vmean* and reshapes this $NT \times 1$ vector to an $N \times T$ matrix, using the MATLAB *reshape()* function. The average over all states for each time period is calculated using the MATLAB *mean()* function which allows the user to indicate whether the mean is calculated over columns *mean(x,1)*, or rows *mean(x,2)*, with our choice begin to average down the columns (over all states) for each each time period.

Estimation results are shown below, where we see ML and MCMC homoscedastic estimates that are similar, along with robust MCMC estimates that are slightly different. We note that in terms of inference which would be based on the direct and indirect effects estimates, there would be no difference in our conclusions. This is also true of our inference regarding spatial dependence, which is around $\hat{\rho} = 0.18$ and significant at the 99% level in all three sets of estimates.

The own-state (direct) impact of changes in job postings on continuing claims for unemployment is negative and significant at the 99% level for all three sets of estimates. This result matches our intuition, as we should expect that more job openings in the typical state i would reduce continuing claims for unemployment insurance in the typical state i. The direct effect of population staying at home is positive as we would expect, more social distancing in the typical state i leads to more continuing claims for unemployment insurance in the typical state i.

```
% results from: sdm_panel_gd
Homoscedastic model
MaxLike SDM model with both region and time period fixed effects
Dependent Variable =         uclaims
R-squared          =    0.9478
corr-squared       =    0.0157
sigma^2            =    0.0520
Nobs,Nvar,#FE      =   2448,     4,    99
log-likelihood     =        134.87487
```

```
# of iterations    =     15
min and max rho     =   -1.0000,   1.0000
total time in secs  =    0.1360
time for lndet      =    0.0130
Pace and Barry, 1999 MC lndet approximation used
order for MC appr  =     50
iter  for MC appr  =     30
***************************************************************
Variable       Coefficient  Asymptot t-stat    z-probability
jobposts         -0.174599       -4.133059         0.000036
athome            0.682434        4.465409         0.000008
W-jobposts        0.010628        0.131012         0.895766
W-athome         -0.933454       -4.049607         0.000051
rho               0.180985        6.804117         0.000000

Direct         Coefficient          t-stat           t-prob         lower 05         upper 95
jobposts         -0.172832       -4.030322         0.000198        -0.261797        -0.089991
athome            0.645966        4.358292         0.000069         0.354022         0.939853

Indirect       Coefficient          t-stat           t-prob         lower 05         upper 95
jobposts         -0.022185       -0.230708         0.818522        -0.216058         0.168172
athome           -0.953022       -3.673754         0.000601        -1.474685        -0.466266

Total          Coefficient          t-stat           t-prob         lower 05         upper 95
jobposts         -0.195017       -1.753141         0.085963        -0.404988         0.029323
athome           -0.307056       -1.408596         0.165400        -0.758508         0.112817

Homoscedastic model
MCMC SDM model with both region and time period fixed effects
Dependent Variable =         uclaims
R-squared          =    0.9478
corr-squared       =    0.0157
sigma^2            =    0.0521
Nobs,Nvar,#FE      =   2448,     2,    99
ndraw,nomit        =   2500,   500
rvalue             =      0
min and max rho     =   -1.0000,   1.0000
total time in secs  =    2.1920
time for MCMC draws =    1.2250
Pace and Barry, 1999 MC lndet approximation used
order for MC appr  =     50
iter  for MC appr  =     30
***************************************************************
Variable       Coefficient  Asymptot t-stat    z-probability
jobposts         -0.175969       -4.245189         0.000022
athome            0.684132        4.357569         0.000013
W-jobposts        0.011332        0.139931         0.888715
W-athome         -0.935979       -4.078070         0.000045
rho               0.180371        7.025468         0.000000

Direct         Coefficient          t-stat           t-prob         lower 05         upper 95
jobposts         -0.176910       -4.227126         0.000025        -0.260435        -0.094539
athome            0.647674        4.268288         0.000020         0.342358         0.942458

Indirect       Coefficient          t-stat           t-prob         lower 05         upper 95
```

```
jobposts          -0.024117        -0.252342         0.800798        -0.222461         0.155683
athome            -0.955232        -3.836535         0.000128        -1.458683        -0.451972

Total            Coefficient          t-stat           t-prob         lower 05         upper 95
jobposts          -0.201027        -1.824144         0.068252        -0.419438         0.018191
athome            -0.307558        -1.471534         0.141275        -0.729211         0.100418

Heterocedastic model
MCMC SDM model with both region and time period fixed effects
Dependent Variable =         uclaims
R-squared          =    0.9479
corr-squared       =    0.0158
sigma^2            =    0.0346
Nobs,Nvar,#FE      =   2448,     2,    99
ndraw,nomit        =   2500,   500
rvalue             =      5
min and max rho    =   -1.0000,   1.0000
total time in secs  =    3.5160
time for MCMC draws =    2.6550
Pace and Barry, 1999 MC lndet approximation used
order for MC appr  =     50
iter  for MC appr  =     30
***************************************************************
Variable       Coefficient  Asymptot t-stat    z-probability
jobposts          -0.160821        -3.962871         0.000074
athome             0.599252         3.906682         0.000094
W-jobposts         0.022847         0.291396         0.770749
W-athome          -0.812927        -3.628539         0.000285
rho                0.199132         8.188863         0.000000

Direct           Coefficient          t-stat           t-prob         lower 05         upper 95
jobposts          -0.161523        -4.008740         0.000063        -0.240070        -0.082423
athome             0.573789         3.955593         0.000079         0.297556         0.868561

Indirect         Coefficient          t-stat           t-prob         lower 05         upper 95
jobposts          -0.015586        -0.170951         0.864277        -0.197634         0.164855
athome            -0.844215        -3.506196         0.000463        -1.312068        -0.359739

Total            Coefficient          t-stat           t-prob         lower 05         upper 95
jobposts          -0.177109        -1.681919         0.092712        -0.386722         0.033177
athome            -0.270425        -1.301998         0.193039        -0.662680         0.140060
```

The indirect (spatial spillover) impact of job offers in the typical state i is not significant, suggesting workers do not respond to job postings in neighboring states j. The indirect impact of population staying at home in neighboring states j is negative and significant at the 99% level. This could arise from either product market impacts of population staying at home in neighboring states on neighboring states economies, or perhaps networks between suppliers in neighboring states and own-states. It might also reflect a simple spatial co-movement of social distancing between own-state and neighboring-states over the weekly time periods. We know from popular press reports that social distancing at the state level exhibited spatial clustering over time, with states in the southern US (e.g., FL, GA, NC, Sc), and upper midwestern states (e.g, MT, ND, SD, WY) engaging in lower levels of social distancing behavior, while east and west coast states engaged in higher levels of social distancing.

Plots of the fixed effects estimates for the 51 weekly time-periods are shown in Figure 2.1 and a plot of the v_{it} variance scalar estimates (averaged over all states) for each of the 51 weeks are shown in Figure 2.2. From the time-specific effects we see large positive effects for weeks 11 to 20 during the height of the pandemic lock down period, starting around the third week of March. These effects are centered on zero, and become negative again around week 40 during the summer of 2020. The variance scalar estimates (averaged over all states) also show that starting around week 11 the noise variances exhibited an increase, reflecting a non-constant variance for the disturbances for these weeks. The figure also shows a rise in the averaged v_{it} scalars for weeks at the end of the year, when a second wave of high Covid-19 infections began to take place.

We could of course also examine the state-specific effects and the average for v_{it} over time periods for each state using similar MATLAB commands. However, the most interesting variation for this model and dataset revolves around the time-dimension of the pandemic.

2.2 The SEM model

The spatial Error (SEM) model is shown in (2.7), which does not include a spatial lag of the dependent variable vector (Wy). This model allows for spatial dependence in the disturbances. There are no *spatial spillovers* in this model. LeSage and Pace (2014) argue that we should not label the spatial impacts arising from dependence in the disturbances as reflecting spatial spillovers. They reserve the term spatial spillovers to describe situations where changes in a variable x_j from region j impacts outcomes y_i in another region i. They label the situation where shocks to the disturbances u_j impact disturbances in other regions $u_i, i \neq j$ as *global spatial shocks* to the disturbances.

$$\begin{aligned}
y &= X\beta + \iota_T \otimes \mu + \nu \otimes \iota_N + u && (2.7)\\
u &= \rho W u + \varepsilon \\
\varepsilon &\sim N(0_{NT}, \sigma^2 V) \\
V &= \begin{pmatrix} v_1 & 0 & \dots & 0 \\ 0 & v_2 & \dots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & \dots & 0 & v_{NT} \end{pmatrix} \\
\pi(\beta) &\sim N(c, C) && (2.8)\\
\pi(r/v_i) &\sim iid\ \chi^2(r), i = 1, \dots, N*T && (2.9)\\
\pi(\sigma^2) &\sim IG(a, b) && (2.10)\\
\pi(\rho) &\sim U(1/\lambda_{min}, 1/\lambda_{max}) && (2.11)
\end{aligned}$$

To see why they use the term *global* when referring to the shocks, note that we can write: $(I_{NT} - \rho W)^{-1} = \varepsilon = I_{NT}\varepsilon + \rho W\varepsilon + \rho^2 W^2 \varepsilon + \dots$, so that shocks to a single region i can impact disturbances from neighboring states $\rho W \varepsilon$, disturbances from neighbors to the neighboring states $\rho^2 W^2 \varepsilon$, and so on for higher-order neighbors captured by terms like $\rho^j W^j \varepsilon$ in the infinite series expansion of $(I_{NT} - \rho W)^{-1}$. It should also be noted that in our static panel data setting the matrix W is block diagonal, with a series of $N \times N$ blocks on the main diagonal of the matrix $Wbig = I_T \otimes W$. This means that shocks to disturbances from one time period cannot impact

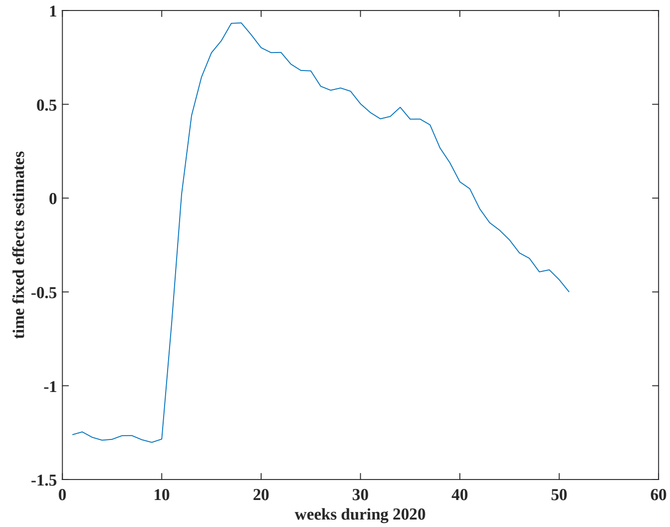

Figure 2.1: Time-period fixed effects estimates

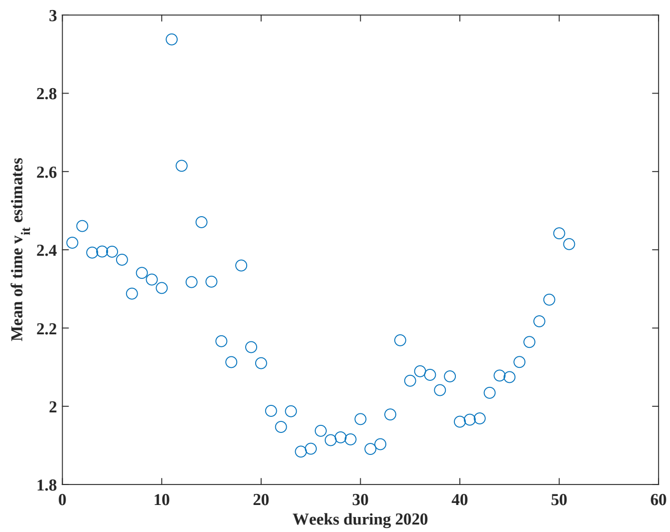

Figure 2.2: Mean (over states) v_{it} estimates

disturbances in other time periods, because the matrix inverse: $(I_{NT} - \rho W)^{-1}$ will also be block diagonal.

In addition to a function *sem_panel_FE_g()* that produces estimates for this model with MCMC and allowing for use of Bayesian prior distributions, heterocedastic/outlier robust estimates based on the variance scalars v_{it}, there is also a maximum likelihood estimation function *sem_panel_FE()*, where again the naming convention of the extension *_g* is used to designate MCMC estimation.

The partial derivatives for this model are the same as those for the OLS model, allowing us to interpret the coefficient estimates for the parameters β as partial derivative impacts arising from changes in each variable x_i for the ith region on outcomes y_i in the same region. As noted, there are not spatial spillover impacts from changes in region i characteristics ($x-$variables) on outcomes in other regions. Formally, for the $r-$th explanatory variable we have:

$$\partial E(y)/\partial X^{r\prime} \quad = \quad \hat{\beta}_r \tag{2.12}$$

Some applied examples of using the SEM model follows.

2.2.1 Using the *sem_panel_FE()*, *sem_panel_FE_g()* functions

The program below estimates both an OLS and SEM panel model using the same empirical model of y =unemployment claims growth and x = index of job posts and annual growth rates of population at home.

LeSage and Pace (2009) argue that if the model specification is correct, OLS and SEM models should produce the same estimates for the coefficients β, but in the presence of spatial dependence in the disturbances, the SEM model estimates will be more efficient, reflected in larger $t-$statistics. If the model specification is not correct, perhaps suffering from omitted variables bias, or incorrect functional form, or perhaps a non-linear relationship instead of the assumed linear specification, then these two sets of estimates will be different.

```
% file: sem_panel_gd
clear all;
[uclaims,b] = xlsread('../demo_data/weekly.xlsx',1);
% read data from sheet 1 of Excel spreadsheet
% growth rate of unemployment 2019-2020 from same week, previous year
snames = strvcat(b(2:end,1)); % 48 state names
tnames = strvcat(b(1,2:end)); % 51 week labels
[N,T] = size(uclaims);
[jobposts,b] = xlsread('../demo_data/weekly.xlsx',2);
% read data from sheet 2 of Excel spreadsheet
% change in job offers from 1st week of 2020
[athome,b] = xlsread('../demo_data/weekly.xlsx',3);
% read data from sheet 3 of Excel spreadsheet
% growth rate of percent population at home
% 2019-2020 from same week, previous year
[a,b] = xlsread('../demo_data/Wcont48.xlsx');
% 48 x 48 contiguity matrix for states
W = normw(a);

y = vec(uclaims);
x = [vec(jobposts) vec(athome)];

vnames = strvcat('y=unclaims','jobposts','athome');

info.model = 3;
ndraw = 2500;
nomit = 500;
info.novi_flag = 1;
result1 = ols_panel_FE_g(y,x,T,ndraw,nomit,info);
prt_panel(result1,vnames);
```

```
prior.novi_flag = 1;
prior.model = 3;
result2 = sem_panel_FE_g(y,x,W,T,ndraw,nomit,prior);
prt_panel(result2,vnames);
```

The resulting estimates are presented below, where we see significant spatial dependence in the disturbances, indicated by the estimate $\hat{\rho} = 0.1835$, that is significant at the 99% level. The estimates for the coefficients on the *jobposts* variable are very similar, including the $t-$statistics. In contrast, the estimates for the coefficients on the *athome* variable from the SEM model are $1.3282 = (0.3509/0.2642)$ times as large as those from OLS. An interesting question is whether this difference between the two estimates for this variable are *significantly* different from a statistical viewpoint.

```
Homoscedastic model
MCMC OLS model with both region and time period fixed effects
Dependent Variable =       y=unclaims
R-squared           =    0.9460
corr-squared        =    0.0097
sigma^2             =    0.0539
Nobs,Nvar,#FE       =   2448,     2,    99
log-likelihood     =        104.02482
prior rvalue       =      0
total time in secs  =    0.2240
ndraws,nomit        =   2500,   500
time for MCMC draws =    0.1810
***************************************************************
Variable        Coefficient  Asymptot t-stat    z-probability
joboffers         -0.177623        -4.146793         0.000034
athome             0.264266         2.268440         0.023302

Homoscedastic model
MCMC SEM model with both region and time period fixed effects
Dependent Variable =       y=unclaims
R-squared           =    0.9460
corr-squared        =    0.0096
sigma^2             =    0.0523
Nobs,Nvar,#FE       =   2448,     2,    99
prior rvalue       =      0
min and max rho     =   -1.0000,   1.0000
ndraws,nomit        =   2500,   500
total time in secs  =   75.9700
time for MCMC draws =   73.1610
Pace and Barry, 1999 MC lndet approximation used
order for MC appr  =     50
iter  for MC appr  =     30
***************************************************************
Variable        Coefficient  Asymptot t-stat    z-probability
joboffers         -0.173744        -4.140312         0.000035
athome             0.350999         2.734304         0.006251
rho                0.183583         6.817240         0.000000
```

We can test the hypothesis that the OLS versus SEM estimates for the coefficient on the *athome* variable are significantly different using the MCMC draws produced during estimation. Here we illustrate this test graphically, using the following MATLAB code snippet to produce posterior distributions for the two β estimates as well as a posterior distribution for the difference between the two estimates. MCMC draws provide a valid basis for constructing posterior distributions, and we use a kernel density function *pltdens()* from my econometrics toolbox functions to produce these.

```
% code snippet from sem_gd.m file
betao = result1.bdraw(:,2);
betas = result2.bdraw(:,2);
beta_diff = betas - betao;

[h1,f1,y1] = pltdens(betao);
[h2,f2,y2] = pltdens(betas);
[h3,f3,y3] = pltdens(beta_diff);

subplot(2,1,1),
plot(y1,f1,'.-r',y2,f2,'.-b');
ylabel('\beta posteriors');
xlabel('\beta values');
legend('\beta_o','\beta_s');
subplot(2,1,2),
plot(y3,f3,'.-g');
ylabel('Posterior for \beta_s - \beta_o');
xlabel('\beta_s - \beta_o values');
zipi = find(y3 > 0);
line([0 0],[0 f3(zipi(1,1))]);
legend('\beta_s - \beta_o','zero');

% trapezoid rule integration
sum_all = trapz(y3,f3);
sum_positive = trapz(y3(zipi,1),f3(zipi,1));
prob = sum_positive/sum_all
```

The posterior distributions for the OLS and SEM parameters associated with the *athome* variable are shown in Figure 2.2, where we see that the differences are small relative to the dispersion of the two posterior distributions. The posterior distribution for the difference between the SEM minus the OLS parameters illustrates that this difference is nearly centered on zero.

Formally, the probability that the two β estimates are significantly different would be the area under the distribution of $\beta_s - \beta_o$ where the $x-$axis is positive, or to the right of the vertical line at zero shown in the figure. We integrate to find this area using simple trapezoid integration (with the last lines of code above). This returns a value of 0.7192, far below the 0.95 level we would needed to conclude that the difference in the two coefficients is significantly different from zero at the 95% level. Note that unless you fix the random number generator seed value, every run of this program will produce slightly different estimates along with slightly different integration results. Of course, if there are no problems with convergence of the MCMC estimates, these differences should not produce a substantive change in the estimates and inferences.

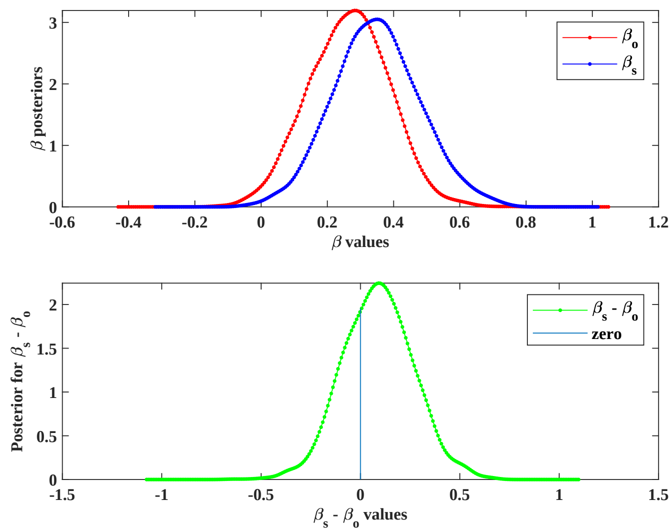

Figure 2.3: Posterior distributions for β_o, β_s

2.3 The SDEM model

The spatial Durbin Error (SDEM) model is shown in (2.13), which does not include a spatial lag of the dependent variable vector (Wy), but has a spatial lag of the explanatory variables WX. This model allows for spatial dependence in the disturbances and *local spatial spillovers*. By this we mean that spillovers from immediate neighbors arise when the coefficients θ on the spatial lag of the explanatory variable are non-zero. These are based only on immediate neighbors, which contrast with what LeSage and Pace (2014) label *global spatial spillovers*. The global spillovers arise when we have a spatial autoregressive process that involves a spatial lag of the dependent variable vector, so that: $E(y) = (I_{NT} - \hat{\rho}W)^{-1}\hat{\beta} = I_{NT}\hat{\beta} + \hat{\rho}WI_{NT}\hat{\beta} + \hat{\rho}^2W^2I_{NT}\hat{\beta} + \ldots$. This allows for spillovers from neighbors (W), neighbors to the neighbors (W^2), and so on, for all higher order neighboring regions $W^j, j = 3, 4, \ldots$.

Recall that LeSage and Pace (2014) argue that we should not label the spatial impacts arising from dependence in the disturbances as reflecting spatial spillovers. They reserve the term spatial spillovers to describe situations where changes in a variable x_j from region j impacts outcomes y_i in another region i. The SDEM specification does allow this type of spillover impact to arise from changes in explanatory variables of immediately neighboring regions, reflected by the spatial lag of the explanatory variables WX.

$$\begin{aligned} y &= X\beta + WX\theta + \iota_T \otimes \mu + \nu \otimes \iota_N + u \\ u &= \rho W u + \varepsilon \\ \varepsilon &\sim N(0_{NT}, \sigma^2 V) \end{aligned} \tag{2.13}$$

$$
\begin{aligned}
V &= \begin{pmatrix} v_1 & 0 & \dots & 0 \\ 0 & v_2 & \dots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & \dots & 0 & v_{NT} \end{pmatrix} \\
\pi(\beta) &\sim N(c, C) && (2.14) \\
\pi(r/v_i) &\sim iid\ \chi^2(r), i = 1, \dots, N*T && (2.15) \\
\pi(\sigma^2) &\sim IG(a, b) && (2.16) \\
\pi(\rho) &\sim U(1/\lambda_{min}, 1/\lambda_{max}) && (2.17)
\end{aligned}
$$

To see why we use the term *local* when referring to the spillovers from this model, note we have partial derivatives for changes in the rth explanatory variable that take the form in (2.18).

$$
\begin{aligned}
\partial E(y)/\partial X^r &= \hat{\beta}_r + W\hat{\theta}_r && (2.18) \\
&= \hat{\beta}_r + \hat{\theta}_r
\end{aligned}
$$

The last line follows from the fact that the matrix W has row-sums of one. This means we can interpret the coefficient estimates for $\hat{\beta}$ as direct (own-region) effects and the estimates for $\hat{\theta}$ as the indirect (other-region) effects arising from changes in WX, explanatory variables of the neighbors. The $t-$statistics will provide a valid inference regarding the significance of the direct and indirect effects estimates. However, if we wish to conduct inference on the statistical significant of the *total effect* estimate, this would need to be based on the sum of the two coefficients $\hat{\beta} + \hat{\theta}$, and a proper measure for the dispersion of this sum of coefficients. This can be constructed from the MCMC draws, which is what the MCMC estimation function does.

Some applied examples of using the SDEM model follows.

2.3.1 Using the *sdem_panel_FE()*, *sdem_panel_FE_g()* functions

There are maximum likelihood and MCMC estimation functions, where again the MCMC functions allow for non-constant variance scalars v_{it}, and for use of prior distributions applied to the parameters β, σ^2. You might wonder why we need a separate function to produce estimates of the SDEM model, when we could produce estimates by simply calling the SEM model function with an augmented explanatory variables matrix $\tilde{X} = \begin{pmatrix} X & WX \end{pmatrix}$. The answer is that the partial derivatives used to calculate the total effects estimates for this model would not be produced by the SEM model function used in this fashion. This is a minor issue, but SDEM provides a clearer printout of the estimates and properly labels them as direct, indirect and total effects estimates.

The ML estimation routine *sdem_panel_FE()* simply prints estimates for the coefficients β, θ and associated $t-$statistics. The MCMC estimation routine *sdem_panel_FE_g()* prints out the estimates β as direct effects, those for θ as indirect effects, and uses the MCMC draws for the sum of these two sets of estimates to produce a total effect and associated $t-$statistic as well as 0.05 and 0.95 intervals for the total effect.

A warning is that if you attempt to use the same structure variable *prior* when calling the estimation function again, all *fields* from the prior call will be still present. That is the motivation for creating new structure variables, *prior2,prior3,prior4* on subsequent calls to the MCMC estimation function. This avoids possible errors.

The demonstration program produces ML estimates as well as MCMC estimates for:

1) a homoscedastic model with no prior, which should be equivalent to the ML estimates,

2) a heteroscedastic/robust set of estimates which should also be equivalent to ML estimates because the DGP used to create the sample data was based on a scalar noise variance,

3) a homoscedastic model where a prior distribution was assigned to the parameters β, θ that was centered on a zero prior mean, but relies on a diagonal variance-covariance matrix with large variances equal to 10 on the diagonal and zero off-diagonal elements. Given the large prior variances assigned, these estimates should also be close to the ML estimates.

4) a heteroscedastic model where a prior distribution was assigned to the parameters β, θ that was centered on a prior mean equal to one, and a diagonal variance-covariance matrix with very small prior variances equal to 0.001 on the diagonal and zero off-diagonal elements. Given the small prior variances assigned here, these estimates should be biased away from the ML estimates, towards the prior mean values of one. Note that the default behavior of the MCMC estimation function is to estimate the parameters v_{it}, or what I have been labeling a heteroscedastic/robust set of estimates.

```
% file sdem_panel_gd.m demo file
clear all;
rng(10203040);

n = 100;
t = 20;
rho = 0.7;
k = 2;
x = randn(n*t,k);
beta = ones(k,1);
beta = [beta
        -ones(k,1)];
sige = 0.1;
evec = randn(n*t,1)*sqrt(sige);

latt = rand(n,1);
long = rand(n,1);
W = make_neighborsw(latt,long,5);
Wbig = kron(eye(t),W);

xo = x;
x = [x Wbig*x];

% add fixed effects to the DGP
tts = (1:n)*(1/n);
SFE = kron(ones(t,1),tts');
ttt = (1:t)*(1/t);
TFE = kron(ttt',ones(n,1));

u = (speye(n*t) - rho*Wbig)\evec;
y = (x*beta + SFE + TFE + u);
```

```
info.model = 3;
result1 = sdem_panel_FE(y,xo,W,t,info);
vnames = strvcat('y','x1','x2');
prt_panel(result1,vnames);

ndraw = 2500;
nomit = 500;
prior.novi_flag = 1;
prior.model = 3;
result2 = sdem_panel_FE_g(y,xo,W,t,ndraw,nomit,prior);
prt_panel(result2,vnames);

prior2.rval = 5;
prior2.model = 3;
result3 = sdem_panel_FE_g(y,xo,W,t,ndraw,nomit,prior2);
prt_panel(result3,vnames);

prior3.novi_flag = 1;
prior3.model = 3;
k = size(x,2);
prior3.beta = zeros(k,1);
prior3.bcov = eye(k)*10;
result4 = sdem_panel_FE_g(y,xo,W,t,ndraw,nomit,prior3);
prt_panel(result4,vnames);

prior4.model = 3;
prior4.beta = ones(4,1);
prior4.bcov = eye(4)*0.001;
result5 = sdem_panel_FE_g(y,xo,W,t,ndraw,nomit,prior4);
prt_panel(result5,vnames);
```

The estimation results below show that ML and MCMC estimates from model specifications: 1), 2) and 3) above are all close to the ML estimates. The last set of estimates for θ exhibit bias towards away from the true values of -1 towards the tightly imposed prior mean of one. The estimates for the parameters β in this case are also biased away from the true values of one, having larger values around 1.25. This likely results from the large amount of bias introduced by the inaccurate prior of 1 on θ, which will also impact the estimates of the other parameters β in the model. Note also, that there will be likely correlation between economic explanatory variables X and the spatial lag of these variables WX, in applied modeling. Pace, LeSage and Zhu (2012) show correlations for typically used census and BEA variables based on counties and census tracts are often above 0.9 which is a result of spatial clustering of regional economies and regional demographics.

```
% output from running sdem_panel_gd.m file
MaxLike SDEM model with both region and time period fixed effects
Homoscedastic model
Dependent Variable =              y
R-squared          =    0.9367
corr-squared       =    0.9297
sigma^2            =    0.0908
Nobs,Nvar,#FE      =   2000,     2,   120
min and max rho    =   -0.9900,   0.9900
total time in secs =    1.3110
time for lndet     =    0.0140
```

```
Pace and Barry, 1999 MC lndet approximation used
order for MC appr  =     50
iter  for MC appr  =     30
***************************************************************
Variable      Coefficient  Asymptot t-stat    z-probability
x1               0.988452       122.210624         0.000000
x2               1.009443       127.690684         0.000000
W-x1            -1.045909       -42.205156         0.000000
W-x2            -0.945451       -39.851527         0.000000
rho              0.687976        40.903251         0.000000

Homoscedastic model
MCMC SDEM model with both region and time period fixed effects
Dependent Variable =              y
R-squared          =    0.9367
corr-squared       =    0.9297
sigma^2            =    0.0905
Nobs,Nvar,#FE      =   2000,     2,   120
ndraw,nomit        =   2500,   500
rvalue             =      0
min and max rho    =   -1.0000,   1.0000
total time in secs  =    1.2360
time for eigs      =    0.0200
time for MCMC draws =    1.1710
Pace and Barry, 1999 MC lndet approximation used
order for MC appr  =     50
iter  for MC appr  =     30
***************************************************************
Variable      Coefficient  Asymptot t-stat    z-probability
x1               0.988254       127.046500         0.000000
x2               1.009033       125.610907         0.000000
W-x1            -1.046859       -41.559864         0.000000
W-x2            -0.947228       -40.869724         0.000000
rho              0.695006        40.926907         0.000000

Direct        Coefficient           t-stat           t-prob         lower 05          upper 95
x1               0.988254       127.046500         0.000000         1.003434          0.972386
x2               1.009033       125.610907         0.000000         1.024870          0.993420

Indirect      Coefficient           t-stat           t-prob         lower 05          upper 95
x1              -1.046859       -41.559864         0.000000        -0.997983         -1.096925
x2              -0.947228       -40.869724         0.000000        -0.900532         -0.991293

Total         Coefficient           t-stat           t-prob         lower 05          upper 95
x1              -0.058605        -1.936193         0.052985        -0.000036         -0.118759
x2               0.061805         2.170462         0.030089         0.117767          0.006431

Heterocedastic model
MCMC SDEM model with both region and time period fixed effects
Dependent Variable =              y
R-squared          =    0.9368
corr-squared       =    0.9298
sigma^2            =    0.0907
Nobs,Nvar,#FE      =   2000,     2,   120
ndraw,nomit        =   2500,   500
```

```
rvalue              =      5
min and max rho     =   -1.0000,   1.0000
total time in secs  =    2.4090
time for eigs       =    0.0190
time for MCMC draws =    2.3760
Pace and Barry, 1999 MC lndet approximation used
order for MC appr  =     50
iter  for MC appr  =     30
***************************************************************
Variable      Coefficient  Asymptot t-stat    z-probability
x1               0.988649       106.024188         0.000000
x2               1.010841       110.798805         0.000000
W-x1            -1.044411       -38.138143         0.000000
W-x2            -0.938817       -35.421805         0.000000
rho              0.690958        41.763079         0.000000

Direct        Coefficient           t-stat           t-prob         lower 05         upper 95
x1               0.988649       106.024188         0.000000         1.006449         0.970117
x2               1.010841       110.798805         0.000000         1.028408         0.993651

Indirect      Coefficient           t-stat           t-prob         lower 05         upper 95
x1              -1.044411       -38.138143         0.000000        -0.988700        -1.099752
x2              -0.938817       -35.421805         0.000000        -0.889938        -0.991937

Total         Coefficient           t-stat           t-prob         lower 05         upper 95
x1              -0.055762        -1.679633         0.093185         0.009151        -0.123436
x2               0.072024         2.254337         0.024283         0.133591         0.007948

Homoscedastic model
MCMC SDEM model with both region and time period fixed effects
Dependent Variable =               y
R-squared           =    0.9367
corr-squared        =    0.9297
sigma^2             =    0.0907
Nobs,Nvar,#FE       =   2000,      2,   120
ndraw,nomit         =   2500,   500
rvalue              =      0
min and max rho     =   -1.0000,   1.0000
total time in secs  =    0.9530
time for eigs       =    0.0170
time for MCMC draws =    0.9210
Pace and Barry, 1999 MC lndet approximation used
order for MC appr  =     50
iter  for MC appr  =     30
***************************************************************
Variable       Prior Mean    Std Deviation
x1               0.000000         3.162278
x2               0.000000         3.162278
W-x1             0.000000         3.162278
W-x2             0.000000         3.162278

***************************************************************
Variable      Coefficient  Asymptot t-stat    z-probability
x1               0.988905       121.851570         0.000000
x2               1.009405       123.605788         0.000000
```

```
W-x1            -1.046108       -42.394019          0.000000
W-x2            -0.945963       -38.868594          0.000000
rho              0.691721        41.540899          0.000000

Direct        Coefficient           t-stat            t-prob          lower 05          upper 95
x1               0.988905       121.851570          0.000000          1.004702          0.973256
x2               1.009405       123.605788          0.000000          1.026178          0.993606

Indirect      Coefficient           t-stat            t-prob          lower 05          upper 95
x1              -1.046108       -42.394019          0.000000         -0.998546         -1.095795
x2              -0.945963       -38.868594          0.000000         -0.899052         -0.993285

Total         Coefficient           t-stat            t-prob          lower 05          upper 95
x1              -0.057203        -1.902357          0.057268          0.000849         -0.117664
x2               0.063442         2.125625          0.033657          0.122085          0.005140

Heterocedastic model
MCMC SDEM model with both region and time period fixed effects
Dependent Variable =               y
R-squared          =    0.6398
corr-squared       =    0.6781
sigma^2            =    0.3589
Nobs,Nvar,#FE      =   2000,     2,   120
ndraw,nomit        =   2500,   500
rvalue             =      5
min and max rho    =   -1.0000,   1.0000
total time in secs =    1.0720
time for eigs      =    0.0160
time for MCMC draws =    1.0240
Pace and Barry, 1999 MC lndet approximation used
order for MC appr  =     50
iter  for MC appr  =     30
***************************************************************
Variable       Prior Mean    Std Deviation
x1               1.000000         0.031623
x2               1.000000         0.031623
W-x1             1.000000         0.031623
W-x2             1.000000         0.031623

***************************************************************
Variable      Coefficient  Asymptot t-stat    z-probability
x1               1.234189        90.718946          0.000000
x2               1.242133        95.537788          0.000000
W-x1             0.432583        13.793900          0.000000
W-x2             0.428508        13.922923          0.000000
rho              0.798371        60.986945          0.000000

Direct        Coefficient           t-stat            t-prob          lower 05          upper 95
x1               1.234189        90.718946          0.000000          1.259704          1.208160
x2               1.242133        95.537788          0.000000          1.267701          1.216677

Indirect      Coefficient           t-stat            t-prob          lower 05          upper 95
x1               0.432583        13.793900          0.000000          0.492006          0.370950
x2               0.428508        13.922923          0.000000          0.487048          0.367068
```

```
Total        Coefficient       t-stat       t-prob       lower 05       upper 95
x1              1.666771    43.410420     0.000000       1.740801       1.590095
x2              1.670642    44.167937     0.000000       1.742791       1.597402
```

2.4 The SLX model

The spatial lag of X model is shown in (2.19), which does not include a spatial lag of the dependent variable vector (Wy), but has a spatial lag of the explanatory variables WX. This model does not allow for spatial dependence in the disturbances, but does allow for *local spatial spillovers*. Theoretically, if there is spatial dependence in the disturbances and the model is correctly specified, with no omitted variables or functional form problems, the SLX estimates should equal those from SDEM, with SDEM being more efficient.

$$
\begin{aligned}
y &= X\beta + WX\theta + \iota_T \otimes \mu + \nu \otimes \iota_N + \varepsilon & (2.19)\\
\varepsilon &\sim N(0_{NT}, \sigma^2 V)\\
V &= \begin{pmatrix} v_1 & 0 & \dots & 0 \\ 0 & v_2 & \dots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & \dots & 0 & v_{NT} \end{pmatrix}\\
\pi(\beta) &\sim N(c, C) & (2.20)\\
\pi(r/v_i) &\sim iid\ \chi^2(r), i = 1, \dots, N*T & (2.21)\\
\pi(\sigma^2) &\sim IG(a, b) & (2.22)\\
& & (2.23)
\end{aligned}
$$

Maximum likelihood estimates for this model can be produced by OLS, so there is no function to produce maximum likelihood estimates. There is a function *slx_panel_FE_g()* that allows for use of the heteroscedastic/robust estimates as well as use of priors for the parameters β, θ, σ^2. of course, this could be accomplished by calling the function *ols_panel_FE_g()*, but that function would not produce total effects estimates that are printed out by the *slx_panel_FE_g()* function.

2.4.1 Using the *slx_panel_FE_g()* function

The program below estimates the pandemic labor market model used to demonstrate the SEM model in Section 2.2.1. We estimate the SLX model specification as well as the SDEM model, which should produce equivalent estimates if there are no model mis-specification issues. We use the homoscedastic noise variance option.

The program compares the estimates for the variable *athome* measuring social distancing, using both the $\beta + \theta$ estimates. We are interested if the total effect of the *athome* variable are the same from both sets of estimates, which consists of the direct effects (β) parameter for this variable, plus the indirect effects (θ) parameter.

The MCMC draws can be used to produce a posterior distribution for the total effect of the *athome* variable for the SLX and SDEM models. This can be done by simply adding the MCMC draws for the two parameters $\beta+\theta$, and using the *pltdens()* function from my econometrics toolbox.

```
% file: slx_panel_gd
clear all;
[unclaims,b] = xlsread('../demo_data/weekly.xlsx',1);
% read data from sheet 1 of Excel spreadsheet
% growth rate of unemployment 2019-2020 from same week, previous year
snames = strvcat(b(2:end,1)); % 48 state names
tnames = strvcat(b(1,2:end)); % 51 week labels
[N,T] = size(unclaims);
[jobposts,b] = xlsread('../demo_data/weekly.xlsx',2);
% read data from sheet 2 of Excel spreadsheet
% change in job offers from 1st week of 2020
[athome,b] = xlsread('../demo_data/weekly.xlsx',3);
% read data from sheet 3 of Excel spreadsheet
% growth rate of percent population at home
% 2019-2020 from same week, previous year
[a,b] = xlsread('../demo_data/Wcont48.xlsx');
% 48 x 48 contiguity matrix for states
W = normw(a);

y = vec(unclaims);
x = [vec(jobposts) vec(athome)];
vnames = strvcat('y=unclaims','jobposts','athome');

info.model = 3;
ndraw = 2500;
nomit = 500;
info.novi_flag = 1;
result1 = slx_panel_FE_g(y,x,W,T,ndraw,nomit,info);
prt_panel(result1,vnames);

prior.novi_flag = 1;
prior.model = 3;
result2 = sdem_panel_FE_g(y,x,W,T,ndraw,nomit,prior);
prt_panel(result2,vnames);

betao = result1.bdraw(:,2);
betas = result2.bdraw(:,2);
beta_diff = betas - betao;

thetao = result1.bdraw(:,4);
thetas = result2.bdraw(:,4);
theta_diff = thetas - thetao;

[h1,f1,y1] = pltdens(betao+thetao);
[h2,f2,y2] = pltdens(betas+thetas);
[h3,f3,y3] = pltdens(beta_diff+theta_diff);

subplot(2,1,1),
plot(y1,f1,'.-r',y2,f2,'.-b');
ylabel('\beta + \theta posteriors');
xlabel('total effects values');
legend('\beta_o+\theta_o','\beta_s+\theta_o');
subplot(2,1,2),
plot(y3,f3,'.-g');
ylabel('Posterior total effects differences');
```

```
xlabel('total effects values');
zipi = find(y3 > 0);
line([0 0],[0 f3(zipi(1,1))]);
legend('total differences','zero');

% trapezoid rule integration
sum_all = trapz(y3,f3);
sum_positive = trapz(y3(zipi,1),f3(zipi,1));
prob = sum_positive/sum_all
```

The estimation result from running the program are shown below, where we see evidence of significant spatial dependence in the disturbances, since $\hat{\rho} = 0.1812$, which is significant at the 99% level. The estimates from the two models are very close, pointing to no model specification issues.

As in the case of the SEM model estimates, the direct effect impact of job offers on continuing unemployment claims is negative, reducing these, while the impact of social distancing captured by the proportion of population at home is positive, raising continuing claims for unemployment insurance. The indirect effect of job offers is not significantly different from zero, suggesting that neighboring state job offers have no impact on own-state unemployment claims. Neighboring state social distancing has a negative impact on unemployment claims, implying that more social distancing in state j reduces (continuing) unemployment insurance claims in state i. This result is not entirely expected, but the reader should note that the relationship being estimated is an unidentified reduced form relationship, which could reflect both shifts in labor demand and supply. Social distancing in state j could impact labor demand in state i as a result of a decline in cross-border shopping, which would impact labor demand because of reduced product market demand, which would reduced labor demand. This would of course, result in an increase, not decrease in unemployment claims. On the other hand, increased social distancing in State j could reflect a reduction in labor supply in state j relative to state i. An increase in labor supply in state i (relative to neighboring states j) could result in a decrease in unemployment claims.

A more formal examination of the question regarding significant differences between SLX and SDEM estimates can be based on the posterior distribution for the total effects. We focus only on the total effects associated with the *athome* variable, since the indirect effects of job offers is not significantly different from zero.

```
% output from running slx_panel_gd.m file
Homoscedastic model
MCMC SLX model with both region and time period fixed effects
Dependent Variable =       y=unclaims
R-squared            =    0.9463
corr-squared         =    0.0155
sigma^2              =    0.0536
Nobs,Nvar,#FE        =   2448,     3,    99
log-likelihood      =         111.1882
prior rvalue        =      0
total time in secs  =    0.2600
time for MCMC draws =    0.2020
***************************************************************
Variable          Coefficient  Asymptot t-stat    z-probability
jobposts            -0.173515       -4.028189         0.000056
athome               0.648057        4.225041         0.000024
W-jobposts          -0.006979       -0.083408         0.933527
```

```
W-athome           -0.875091       -3.738658        0.000185

Direct            Coefficient          t-stat          t-prob        lower 05        upper 95
jobposts           -0.174024       -4.039993        0.000055       -0.254785       -0.090399
athome              0.646168        4.212722        0.000026        0.344772        0.944722

Indirect          Coefficient          t-stat          t-prob        lower 05        upper 95
jobposts           -0.007535       -0.090055        0.928251       -0.172922        0.155034
athome             -0.876400       -3.744252        0.000185       -1.330662       -0.412436

Total             Coefficient          t-stat          t-prob        lower 05        upper 95
jobposts           -0.181559       -1.932440        0.053420       -0.368393        0.004255
athome             -0.230232       -1.269063        0.204539       -0.581188        0.128230

Homoscedastic model
MCMC SDEM model with both region and time period fixed effects
Dependent Variable =       y=unclaims
R-squared          =    0.9463
corr-squared       =    0.0154
sigma^2            =    0.0521
Nobs,Nvar,#FE      =   2448,     2,    99
ndraw,nomit        =   2500,   500
rvalue             =      0
min and max rho    =   -1.0000,   1.0000
total time in secs =   69.1540
time for eigs      =    3.1660
time for MCMC draws =   65.9290
Pace and Barry, 1999 MC lndet approximation used
order for MC appr  =     50
iter  for MC appr  =     30
*************************************************************
Variable          Coefficient  Asymptot t-stat    z-probability
jobposts           -0.177115       -4.134536        0.000036
athome              0.627619        4.232993        0.000023
W-jobposts         -0.037995       -0.438811        0.660798
W-athome           -0.869383       -3.801019        0.000144
rho                 0.182301        6.951916        0.000000

Direct            Coefficient          t-stat          t-prob        lower 05        upper 95
jobposts           -0.177115       -4.134536        0.000037       -0.260944       -0.094399
athome              0.627619        4.232993        0.000024        0.323024        0.905946

Indirect          Coefficient          t-stat          t-prob        lower 05        upper 95
jobposts           -0.037995       -0.438811        0.660837       -0.203135        0.128159
athome             -0.869383       -3.801019        0.000148       -1.315483       -0.425761

Total             Coefficient          t-stat          t-prob        lower 05        upper 95
jobposts           -0.215110       -2.113839        0.034630       -0.414329       -0.020512
athome             -0.241765       -1.220109        0.222541       -0.625583        0.160024

prob =  0.4853
```

Figure 2.4 shows the posterior distribution for total effects associated with the *athome* variable, constructed by summing the MCMC draws for the β_2 parameter on own-state *athome* and the MCMC draws for the θ_2 parameter on neighboring state *athome*. The difference between these total effects from the SLX and SDEM estimates are then used to construct a posterior distribution, shown in the second part of the plot. This distribution is centered very near zero, and the trapezoid rule integration of the area above zero relative to the entire area, produced a probability of 0.4775, consistent with what we see in the posterior distribution. This allows us to reject that the difference in total effects estimates are significant.

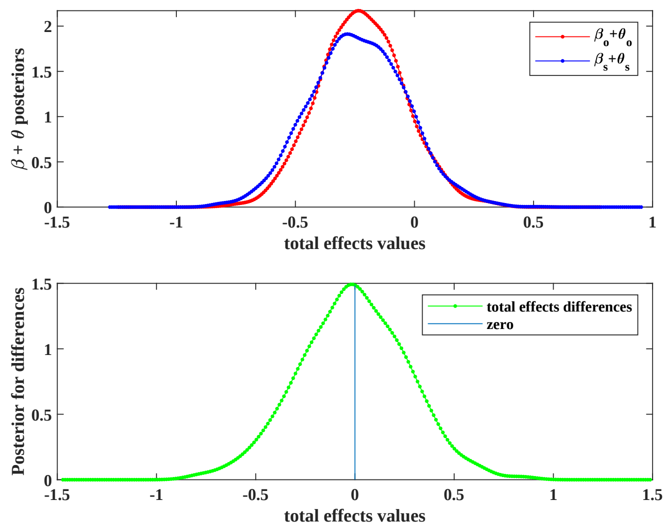

Figure 2.4: Posterior distributions for total effects

2.5 Chapter summary

This chapter provided illustrations for use of the SDM, SEM, SDEM and SLX model estimation functions. It also showed how MCMC draws can be useful for hypothesis testing, although formally, Bayesian do not engage in hypothesis testing, but rather calculate posterior probabilities regarding any questions concerning parameter magnitudes.

The toolbox functions described here that produce MCMC estimates printout posterior means for the parameters as well as the direct, indirect and total effects estimates needed for interpreting how changes in the explanatory variables impact the dependent (or outcome) variable. The posterior means should be equivalent to point estimates from ML estimation if no strongly imposed prior distributions are assigned to the model parameters. The $t-$statistics printed by the *prt_panel()* function represent the posterior mean divided by the posterior standard deviation, both of which are calculated from the retained MCMC draws. The retained draws are those starting after the omitted draws used to adjust for start-up of the MCMC sampler, which relies on arbitrary initial values for the parameters. In the notation used here, the retained draws would number *(ndraw-nomit)*.

An article by Gelfand et. al (1990) provides theoretical proofs that inference based on combining MCMC draws for different model parameters can be used to construct valid intervals regarding additive, multiplicative and even non-linear combinations of parameters for linear regression models. Their results are applicable for MCMC estimates from spatial regression models of the type considered here, which greatly simplifies inference regarding combinations of parameters.

2.6 Chapter references

Gelfand, A.E., S.E. Hills, A. Racine-Poon and A.F.M. Smith (1990). Illustration of Bayesian Inference in Normal Data Models Using Gibbs Sampling, *Journal of the American Statistical Association*, 85, 972-985.

LeSage, J. P. and R. K. Pace (2014). Interpreting Spatial Econometric Models, *Handbook of Regional Science*, M. M. Fischer and Peter Nijkamp (Eds.) Springer, Berlin 2014, pp. 1535-1552.

Pace, R. K., J. P. LeSage and S. Zhu (2012). Spatial Dependence in Regressors, *Advances in Econometrics*, Volume 30, Thomas B. Fomby, R. Carter Hill, Ivan Jeliazkov, Juan Carlos Escanciano and Eric Hillebrand (Series Eds., Volume editors: Dek Terrell and Daniel Millimet), Emerald Group Publishing Limited, 2012, pp. 257-295.

Chapter 3

Convex combinations of spatial weight matrix models

This chapter discusses some recent models that allow a convex combination of spatial weight matrices to be used for estimation of the SAR, SEM, SDM, and SDEM panel data model specifications. The convex combination takes the form: $W_c = \gamma_1 W_1 + \gamma_2 W_2 + \ldots + \gamma_m W_m$, where the scalar weights assigned to each of the spatial weight matrices must be positive and sum to unity. This of course means that $\gamma_m = 1 - \gamma_1 - \gamma_2 - \ldots - \gamma_{m-1}$.

Spatial regression models typically rely on a single weight matrix constructed based on spatial proximity (neighboring regions with borders in common), or some number (say m) of nearest neighboring regions, or points (e.g., firms, consumers, houses) located in space. This approach has two advantages: 1) geographical location of observations is objective and easy to determine, and 2) weight matrices based on geographical space can be viewed as fixed over time and in most cases exogenous.

There has been a great deal of criticism of weight matrices based solely on spatial location of observations, Corrado and Fingleton (2012). This criticism in part derives from application of spatial regression models to broader contexts involving interregional flows of: goods (e.g., trade), population (e.g., migration), knowledge (e.g., patent citations); student peer groups, social networks, etc., where geographical location of observations does not seem intuitively or theoretically appealing.

There are a limited number of studies where weight matrices reflecting connectivity of observations have been motivated by underlying theoretical considerations. For example Behrens, Ertur and Koch (2012) derive a quantity-based structural gravity equation system where both trade flows and error terms are cross-sectionally correlated based on population shares of regions in the sample, and Koch and LeSage (2015) show that the multilateral resistance concept from trade theory Anderson and van Wincoop (2003,2004) can be viewed as a simultaneous autoregressive dependence structure involving gross domestic product shares of the sample regions as well as other types of generalized distance factors.

Debarsy and LeSage (2018) note that once we open the door to non-spatial metrics as a way to specify dependence between cross-sectional observations, a host of issues arise, which are discussed in LeSage and Pace (2011). Blankmeyer et. al (2011) state:

> A single weight matrix, based on a multivariate similarity criterion (generalized distance)

requires a norm to prevent scale differences from influencing the weight placed on the various measures of similarity. (This is unlike the case of spatial proximity where Euclidian distance provides a natural scaling.)

The convex combination approach avoids the issue of scaling for different metrics used in a generalized measure of distance as each connectivity matrix entering the convex combination is normalized beforehand. Note that the convex combination of weights W_c will preserve the row-normalization. Estimates for the parameters $\gamma_m, m = 1, \ldots, M$ provide an inference on the relative importance of different types of connectivity structures for the empirical model being explored.

There were some early efforts to produce estimates for models based on combinations of spatial weight matrices, for example Pace and LeSage (2002), Blankmeyer et. al (2011), and Hazir, et. al (2018). A number of computational issues need to be addressed in order to produce estimates for this type of model, and and Debarsy and LeSage (2018,2021) extended ideas from Pace and LeSage (2002) to the case of MCMC estimation for these models. The computationally difficult part of estimating these models involves calculation of the logged determinant: $log\ |I_{NT} - \rho W|$, which Debarsy and LeSage (2021) tackle with a Taylor-series expansion, following the approach taken by Pace and LeSage (2002).

The MCMC estimation routines in this toolbox rely on the Taylor-series expansion approach, which might be inaccurate for panel data sets involving small N. Practitioners use of the functions in this toolbox should help shed light on this issue.

In order to exploit a computationally efficient approach to MCMC estimation of these models, no prior information is allowed, and the noise disturbances are assumed to obey homoscedastic, scalar variance over both time and space (after taking into account fixed effects for both space and time).

3.1 The SAR convex combination of W model

The SAR convex combination model is shown in (3.1), where each $NT \times NT$ matrix W_m represents some sort of connectivity between regions, and takes a block diagonal form: $I_T \otimes w_m$, where w_m is an $N \times N$ weight matrix, with main diagonal elements equal to zero and row-sums equal to one. For example, consider the case of two matrices w_1, w_2, where w_1 might be a first-order spatial contiguity matrix, and w_2 an $N \times N$ weight matrix constructed based on commodity flows between the N regions. The motivation for the convex combination of weights models is to allow different types of connectivity, for example spatial versus economic connectivity in the case of the two weight matrices above.

$$\begin{aligned} y &= \rho W_c(\Gamma) y + X\beta + \varepsilon, \quad \varepsilon \sim N(0, \sigma^2 I_n), \\ W_c(\Gamma) &= \sum_{m=1}^{M} \gamma_m W_m, \quad 0 \leq \gamma_m \leq 1, \ \sum_{m=1}^{M} \gamma_m = 1, \\ \Gamma &= (\gamma_1, \ldots, \gamma_M)' \end{aligned} \tag{3.1}$$

The $NT \times k$ matrix X in (3.1) contains exogenous explanatory variables, with β being the associated $k \times 1$ vector of parameters. The $NT \times 1$ vector ε represents a constant variance normally distributed disturbance term, and the $NT \times 1$ dependent variable vector y takes the same form as in our models from Chapters 1 and 2.

The SAR model in (3.1) can be re-expressed as in (3.2), which is a more computationally convenient expression that isolates the parameters $\rho, \gamma_m, m = 1, \ldots, M$ in the $(M+1) \times 1$ vector ω.

$$\begin{aligned}
\tilde{y}\omega &= X\beta + \varepsilon \\
\tilde{y} &= (y,\ W_1 y,\ W_2 y,\ \ldots,\ W_M y) \\
\omega &= \begin{pmatrix} 1 \\ -\rho\Gamma \end{pmatrix}, \\
\Gamma &= \begin{pmatrix} \gamma_1 \\ \gamma_2 \\ \vdots \\ \gamma_M \end{pmatrix} \\
\gamma_M &= 1 - \gamma_1 - \gamma_2 - \ldots - \gamma_{M-1}
\end{aligned} \tag{3.2}$$

The value of isolating the parameter vector ω is that we can pre-calculate the $NT \times (M+1)$ matrix $\tilde{y}$ prior to the beginning of the MCMC sampling loop, since this matrix contains only sample data.

The likelihood of the model in (3.2) is shown in (3.3), where $\mathcal{W} = W_1, \cdots, W_M$, and we use the notation: $W_c(\Gamma)$ to indicate that the convex combination matrix W_c depends on the parameters in the vector Γ.

$$\begin{aligned}
f(y|X, \mathcal{W}; \rho, \Gamma, \sigma^2, \beta) &= |R(\omega)| \left(2\pi\sigma^2\right)^{-NT/2} \exp\left(-\frac{e'e}{2\sigma^2}\right) \\
e &= \tilde{y}\omega - X\beta \\
R(\omega) &= I_{NT} - \rho W_c(\Gamma)
\end{aligned} \tag{3.3}$$

Note that the likelihood function contains the term: $|R(\omega)|$ which represents the Jacobian of the transformation from the disturbances to the dependent variable vector. As noted above, this complicates MCMC (or maximum likelihood) estimation of the model because changes in the parameters (ρ, Γ) during optimization (or MCMC sampling) requires that we calculate the (log of) this determinant.

The functions for MCMC estimation in this toolbox restrict $\rho \in (-1, 1)$, so that $R(\omega)^{-1} = \sum_{j=0}^{\infty} \rho^j W_c^j(\Gamma)$ exhibits an underlying stationary process. The true lower bound on the parameter space for ρ is a function of the smallest eigenvalue of $W_c(\Gamma)$, which we label λ_N, with $\rho \in (-\lambda_N^{-1}, 1)$.[1] Of course, this would create additional problems for efficient estimation because the smallest eigenvalue would need to be re-calculated anytime the parameters in Γ change.

If you are interested in the computational details regarding MCMC estimation for this model, see (Debarsy and LeSage, 2021). The focus here is on use of these models in an applied setting, and issues pertaining to extension of the models to the case of SDM, SDEM which are not discussed in Debarsy and LeSage (2021).

[1]Note that because the matrices w_m that make up the block diagonal matrix W_c are of dimension $N \times N$, we have only N eigenvalues.

There are advantages to the convex combination approach to incorporating multiple weight matrices in a spatial regression setting relative to models that have been labeled *higher-order* models. A higher-order model is shown in (3.4) (see Lacombe, 2014).

$$\begin{aligned} y &= \rho_1 W_1 y + \rho_2 W_2 y + \rho_3 W_3 y + X\beta + \varepsilon \\ \varepsilon &\sim N(0, \sigma^2 I) \end{aligned} \tag{3.4}$$

There are a number of theoretical and computational issues that arise for this type of model. One is that bounds on the dependence parameter ρ_1 depend in a non-linear way on values taken by the parameters ρ_2, ρ_3, which produces computational issues. Another issue is that the matrix of partial derivatives for this model takes the form: $\partial E(y)/\partial x_r = (I_{NT} - \hat{\rho}_1 W_1 - \hat{\rho}_2 W_2 - \hat{\rho}_3 W_3)^{-1} I_{NT} \hat{\beta}_r$, which requires special software code.

Note that in the case of our convex combination model, we have: $\partial E(y)/\partial x_r = (I_{NT} - \hat{\rho} W_c(\hat{\Gamma}))^{-1} I_{NT} \hat{\beta}_r$, so given estimates for the parameters $, \hat{\rho}, \hat{\Gamma}$, we can rely on computationally efficient ways to calculate the scalar effects estimates described in LeSage and Pace (2009).

The next section illustrates use of the SAR convex combination estimation functions.

3.1.1 Using the *sar_conv_panel_FE_g()* function

The program below reads an Arcview shapefile using the function *shape_read()* from my *arc_map_version1.0* toolbox functions available for download (separate from my econometrics toolbox), as a zip file from *www.spatial-econometrics.com*, under the 'download' tab. Unzip the file and add the (unzipped) folder named *arc_mat_ver1.0* to your MATLAB path, using the *set path* tool. The downloaded files contain the Arcview shape file for the 3,111 US counties that I am using in the demonstration.

The program extracts the latitude-longitude coordinates for the centroids of the counties and uses a Delaunay triangle routine from MATLAB to produce a spatial contiguity weight matrix, named *Wcont* here using the function *xy2cont()* from my econometrics toolbox.

A second weight matrix named *W6* is created that contains six equally weighted neighbors using the latitude-longitude coordinates and the function *make_neighborsw()* from my econometrics toolbox.

A third matrix named *Wdist* is created based on inverse distances with a cut-off of 6 nearest neighboring counties. This is done using the *distance()* function from my toolbox to produce an $N \times N$ matrix of pairwise distances, which is then converted to inverse distances using element-by-element division, *(ones(n,n)./Wdist)*. Finally, element-by-element multiplication by the *W6* matrix should zero-out distances for all neighbors that are not in the set of 6 nearest. Note the final result is row-normalized using the *normw()* function from my toolbox.

The true DGP relies only on two of the three $W-$matrices, specifically *Wcont* and *W6*, because the true $\gamma_m, m = 1, 2, 3$ are assigned values of $\gamma_1 = 0.3, \gamma_2 = 0.7, \gamma_3 = 0.0$.

A matrix named *Wtrue* is constructed based on $W_c = 0.3Wcont + 0.7W6$ and used to produce ML estimates by calling the *sar_panel_FE()* function.

A second set of estimates feeds down all three $W-$matrices to the function *sar_conv_panel_g()* which produces estimates for the model parameters $\beta, \rho, \gamma_1, \gamma_2, \gamma_3, \sigma^2$.

A third set of estimates is produced using the function *sar_conv_panel_g()* called using only the two true $W-$matrices, *Wcont* and *W6*.

```
% sar_conv_panel_gd demo file
clear all;
rng(10203444);
% read an Arcview shape file for 3,111 US counties
map_results = shape_read('../demo_data/uscounties_projected');
latt = map_results.data(:,3);
long = map_results.data(:,4);
n = length(latt);
t = 20;

% plot(long,latt,'.');

[junk,Wcont,junk] = xy2cont(latt,long); % Delaunay contiguity W

W6 = make_neighborsw(latt,long,6);% 6 nearest neighbors W

Wdist = distance(latt,long) + eye(n);
% create inverse distance with a 6 neighbor cut-off W
Wcut = (ones(n,n)./Wdist).*W6;
Wdist = normw(Wcut);

rho = 0.6;
k = 2;
x = randn(n*t,k);
beta = ones(k,1);
sige = 1;
evec = randn(n*t,1)*sqrt(sige);

gamma1 = 0.3;
gamma2 = 0.7;
gamma3 = 0.0;

Wc = gamma1*kron(eye(t),Wcont) + gamma2*kron(eye(t),W6) + gamma3*kron(eye(t),Wdist);

% add fixed effects to the DGP
tts = (1:n)*(1/n);
SFE = kron(ones(t,1),tts');
ttt = (1:t)*(1/t);
TFE = kron(ttt',ones(n,1));

y = (speye(n*t) - rho*Wc)\(x*beta + SFE + TFE + evec);

ndraw = 20000;
nomit = 2000;
prior.model = 3;
prior.novi_flag = 1;
prior.thin = 4;
prior.plt_flag = 1;

Wtrue = gamma1*Wcont + gamma2*W6;

result1 = sar_panel_FE(y,x,Wtrue,t,prior);
vnames = strvcat('y','x1','x2');
fprintf(1,'sar_panel_FE based on true Wc \n');
prt_panel(result1,vnames);
```

```
Wmatrices = [kron(eye(t),Wcont) kron(eye(t),W6) kron(eye(t),Wdist)];

fprintf(1,'sar_conv_panel_g with 3 W-matrices \n');
result3 = sar_conv_panel_g(y,x,Wmatrices,n,t,ndraw,nomit,prior);
vnames = strvcat('y','x1','x2');
prt_panel(result3,vnames);

Wmatrices = [Wcont W6];

fprintf(1,'sar_conv_panel_g with 2 true W-matrices \n');
result4 = sar_conv_panel_g(y,x,Wmatrices,n,t,ndraw,nomit,prior);
vnames = strvcat('y','x1','x2');
prt_panel(result4,vnames);
```

The function call to *sar_conv_panel_g()* sets user options for a model with both region- and time-specific fixed effects (*prior.model=3*), and a homoscedastic model *prior.novi_flag = 1;*, where I note that the heteroscedastic/robust model is not currently available, but may be implemented in the future. The option *prior.thin = 4;* indicates that we will produce posterior estimates based on skipping every 4th MCMC draw. Since we are using 20,000 MCMC draws with the first 2,000 omitted for start-up, this results in 18,000/4 = 4,500 MCMC draws returned in the results structure for things like the parameters $(\beta, \sigma, \rho, \gamma)$ as well as the direct, indirect, total effects scalar summary effects estimates. Another input option is *prior.plt_flag = 1;*, which produces plots of the MCMC parameter draws in groups of 1,000. That is for every 1,000 MCMC draws a plot appears and the plot is updated after the next 1,000 MCMC draws.

Figure 3.1 shows a plot of the last 1,000 MCMC draws, where we see plots of the γ parameter draws, the ρ draws, the β draws, and σ^2. Stationary plots (such as those shown in the figure) are indicative of successful MCMC sampling, whereas non-stationary plots where the parameter draws wander over sampling point to a problem.

The three sets of estimation results are shown below, where we see that ML estimates based on the true matrix W_c produce accurate estimates for the parameters β, ρ, σ^2.

Of course, in practice we wish to estimate the parameters for the convex combination weights $\gamma_i, i = 1, 2, 3$, which are presented in the second set of estimates presented, which were produced by the *sar_conv_panel_g()* function. The estimates for the parameters β, ρ are close to the true values of 1.0 (for both $\beta_k, k = 1, 2$ and that for $\rho = 0.6110$ is close to the true value of 0.6 for this parameter. The $\hat{\gamma}_1 = 0.3239$ close to the true value of 0.30, while $\hat{\gamma}_2 = 0.6576$, close to the true value of 0.7. The estimate for $\hat{\gamma}_3 = 0.0183$ and is not significantly different from zero based on the $t-$statistic of 1.5114.

```
% results from running sar_conv_panel_gd.m
sar_panel_FE based on true Wc
Homoscedastic model
MaxLike SAR model with both region and time period fixed effects
Dependent Variable =               y
R-squared          =    0.8065
corr-squared       =    0.6668
sigma^2            =    0.9571
Nobs,Nvar,#FE      =  62220,     3,    3131
log-likelihood     =       -89124.943
# of iterations    =     16
```

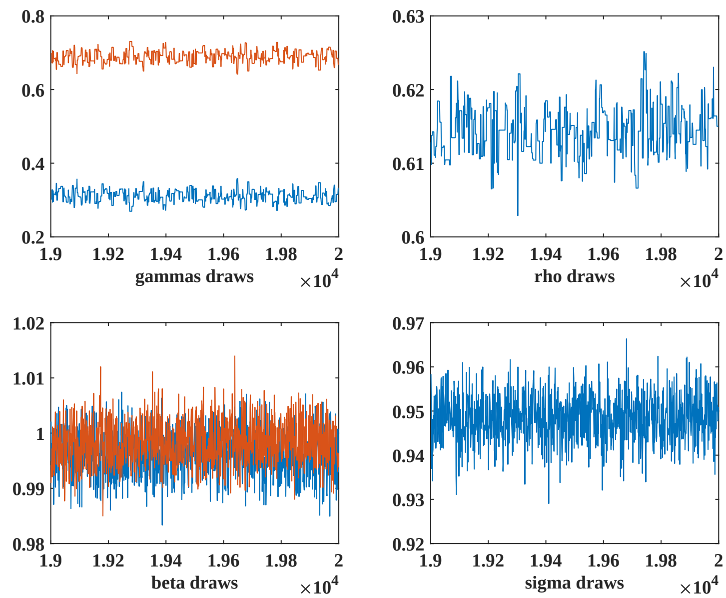

Figure 3.1: MCMC monitoring draws

```
min and max rho      =   -1.0000,    1.0000
total time in secs   =    0.7320
time for lndet       =    0.0580
time for MCMC draws =    0.0480
Pace and Barry, 1999 MC lndet approximation used
order for MC appr   =     50
iter  for MC appr   =     30
***************************************************************
Variable      Coefficient  Asymptot t-stat    z-probability
x1               1.002952       248.164276         0.000000
x2               1.005512       248.853611         0.000000
rho              0.594966       187.250463         0.000000

Direct        Coefficient           t-stat           t-prob         lower 05         upper 95
x1               1.083038       239.300525         0.000000         1.091608         1.073812
x2               1.085925       248.191672         0.000000         1.094264         1.077033

Indirect      Coefficient           t-stat           t-prob         lower 05         upper 95
x1               1.393332        77.208260         0.000000         1.428575         1.357175
x2               1.397043        78.088544         0.000000         1.433187         1.362902

Total         Coefficient           t-stat           t-prob         lower 05         upper 95
x1               2.476370       120.083746         0.000000         2.514884         2.435976
x2               2.482968       122.455556         0.000000         2.522891         2.443266

sar_conv_panel_g with 3 W-matrices
MCMC SAR convex combination W model with both region and time period fixed effects
Bayesian spatial autoregressive convex W model
```

```
Dependent Variable =              y
Log-marginal        = -110513.2487
Log-marginal MCerror=  0.020279
R-squared           =    0.8074
corr-squared        =    0.6668
mean of sige draws  =    0.9527
posterior mode sige =    0.9526
Nobs, Nvars         =  62220,     2
ndraws,nomit        =  50000, 20000
time for effects    =    9.3910
time for sampling   =   54.4340
time for Taylor     =    4.2195
thinning for draws  =        10
min and max rho     =   -1.0000,   1.0000
***************************************************************
      MCMC diagnostics ndraws = 3000
Variable        mode         mean     MC error        tau     Geweke
x1            1.0007       1.0008   0.00007307   0.978535   0.999588
x2            1.0031       1.0031   0.00007750   0.949026   0.999841
rho           0.6114       0.6111   0.00005269   0.964930   0.999786
gamma1        0.3254       0.3240   0.00042176   1.761557   0.991865
gamma2        0.6667       0.6577   0.00042102   1.752002   0.995793
gamma3        0.0078       0.0183   0.00018593   1.124896   0.992024

***************************************************************
      Posterior Estimates
Variable      Coefficient  Asymptot t-stat    z-probability
x1               1.000774       242.576507         0.000000
x2               1.003065       251.131934         0.000000
rho              0.611082       187.512217         0.000000
gamma1           0.323983        20.293881         0.000000
gamma2           0.657686        35.541550         0.000000
gamma3           0.018331         1.511470         0.130669

Direct        Coefficient           t-stat           t-prob        lower 05        upper 95
x1               1.087104       239.213211         0.000000        1.077934        1.095975
x2               1.089593       245.691705         0.000000        1.080488        1.098125

Indirect      Coefficient           t-stat           t-prob        lower 05        upper 95
x1               1.486293        71.954341         0.000000        1.445704        1.527136
x2               1.489698        71.700354         0.000000        1.448612        1.529063

Total         Coefficient           t-stat           t-prob        lower 05        upper 95
x1               2.573397       111.183174         0.000000        2.527817        2.619088
x2               2.579291       110.910699         0.000000        2.533995        2.623346

sar_conv_panel_g with 2 true W-matrices
MCMC SAR convex combination W model with both region and time period fixed effects
Homoscedastic model
Bayesian spatial autoregressive convex W model
Dependent Variable =              y
Log-marginal        = -110508.6217
Log-marginal MCerror=  0.015417
R-squared           =    0.8074
corr-squared        =    0.6668
```

```
mean of sige draws  =    0.9526
posterior mode sige =    0.9527
Nobs, Nvars         =  62220,     2
ndraws,nomit        =  50000, 20000
time for effects    =    9.5070
time for sampling   =   58.5050
time for Taylor     =    1.2406
thinning for draws  =       10
min and max rho     =   -1.0000,   1.0000
***************************************************************
      MCMC diagnostics ndraws = 3000
Variable          mode         mean     MC error         tau     Geweke
x1              1.0007       1.0006   0.00006682    1.017429   0.999779
x2              1.0031       1.0031   0.00006082    1.084098   0.999711
rho             0.6115       0.6116   0.00005941    0.967024   0.999817
gamma1          0.3276       0.3275   0.00034587    1.098705   0.994609
gamma2          0.6724       0.6725   0.00034587    1.098705   0.997370

***************************************************************
      Posterior Estimates
Variable      Coefficient  Asymptot t-stat    z-probability
x1               1.000644       248.616927         0.000000
x2               1.003123       248.035927         0.000000
rho              0.611566       187.347321         0.000000
gamma1           0.327534        21.023024         0.000000
gamma2           0.672466        43.162742         0.000000

Direct        Coefficient           t-stat           t-prob         lower 05         upper 95
x1               1.086957       242.915572         0.000000         1.078072         1.095764
x2               1.089650       244.781683         0.000000         1.080945         1.098398

Indirect      Coefficient           t-stat           t-prob         lower 05         upper 95
x1               1.489319        71.415644         0.000000         1.450240         1.530811
x2               1.493007        72.123102         0.000000         1.452769         1.533342

Total         Coefficient           t-stat           t-prob         lower 05         upper 95
x1               2.576276       110.181526         0.000000         2.532257         2.622853
x2               2.582657       111.751236         0.000000         2.537427         2.626681
```

Another point to note regarding the printed output shown above is that a set of diagnostics for convergence of the MCMC sampling estimates for the SAR convex combination model are presented. When estimating complex models using MCMC methods, there is often concern about whether the sampling estimation procedure gets stuck in part of the parameter space and produces non-converged estimates. There is a large literature on alternative approaches to diagnosing the issue of convergence of the MCMC sampling procedure, and I have incorporated some of the more popular convergence diagnostics into the printed results for convex combination models. The methods used to produce MCMC estimates for these models are sophisticated and rely on numerous numerical approximations making it more likely that users might encounter convergence issues. Convergence of the MCMC sampler for the non-convex combination of weight matrix static panel spatial regression models is not a problem, unless you rely on extremely poorly scaled variable vectors or collinear sets of explanatory variables.

The diagnostics show the mean and mode, which provide an indication of how *symmetric* the distribution of MCMC draws for the model parameters are. Note that since the parameters γ_1, γ_2 are restricted to lie in the $(0, 1)$ interval, estimates close to the boundaries of 0 and 1 will likely be skewed. For the second set of estimates above where we know that the true value for $\gamma_3 = 0$, we see that this is the case because the mean estimate is 0.0183, far above the modal value of 0.0078, pointing to a right-skewed set of MCMC draws. We will have more to say about this later.

The diagnostics also report and estimate of the accuracy/error in the estimates due to MCMC sampling, which should be small relative to the estimated parameter values. This measure labeled *MC error* in the printout is based on the standard deviation (a measure of dispersion) from batched means calculated based on the sequence of MCMC draws.

The *tau* diagnostic is an estimate of the autocorrelation of the draws. For the case of independent MCMC draws with no correlation, the variance would be a function of (1/ndraws), so more draws reduces the uncertainty in the estimates. For correlated draws, the variance is a function of (tau/ndraws), so we want to see estimated values for tau close to one, with larger values (say 10) meaning that we require ndraws/tau to have an effective sample of ndraws. Formally, tau is the number of draws needed before the MCMC (chain) “forgets” where it started. So, if tau = 10, and we carry out 20,000 MCMC draws, we have only 2,000 “effective” draws on which to calculate the means, standard deviations etc. of our estimated parameters. From the printed results for the model based on all three $W-$matrices, we see tau values close to one for the β parameter draws (0.9785 and 0.9490), a value of 0.9649 for the ρ draws, and ranging between 1.1248 and 1.7615 for the γ parameter draws. Since we have set the input option to “thin” the MCMC draws by using only every 4th draw, this helps avoid autocorrelation in the draws, but the *tau* diagnostic indicates there is still autocorrelation in the MCMC draws for the γ parameters. More will be said about this later.

The *Geweke* statistic in the printed output is a test for for equality of the means of the first 10% of the MCMC draws and the last 50% of the draws, which should be interpreted as 1 - *Geweke* in favor of equality. So, a value of 0.05 would be an indication that we cannot reject equality of the draws at the 95% level. Of course, equality of the draws is an indication of convergence, so we want large values of *Geweke* that allow us to conclude MCMC convergence. In the results shown below, most of the *Geweke* statistics are 0.99, pointing to convergence at the 99% level or above, for all parameters.

If there are concerns regarding convergence we can produce estimates based on a larger number of MCMC draws. Note that for our sample of $N = 3111, T = 20$, where $N \times T = 62,220$ it took around 35 seconds to produce 20,000 MCMC draws, and 75 seconds to produce the estimates presented here based on the set of 50,000 MCMC draws.

Figure 3.2 shows a histogram of the 3,000 retained MCMC draws from the run based on *ndraw=50,000, nomit = 20,000, prior.thin=10.* This of course is consistent with the skew of the distribution implied by the mean versus median values.

The points to a possible problem with inclusion in the convex combination of weights matrices models where having many redundant or irrelevant matrices will result in non-zero estimates for the γ parameters associated with these weights that are truly zero. This will have a small impact on the γ estimates for weight matrices that are truly relevant if there is only one or two irrelevant weight matrices. However, if there are a large number of such matrices, the small impacts could build up to have a significant impact on the non-zero *gamma* estimates, impacting our conclusions. Debarsy and LeSage (2021) show how to calculate posterior model probabilities for models based on all possible combinations of two or more weight matrices. The estimation routines print out

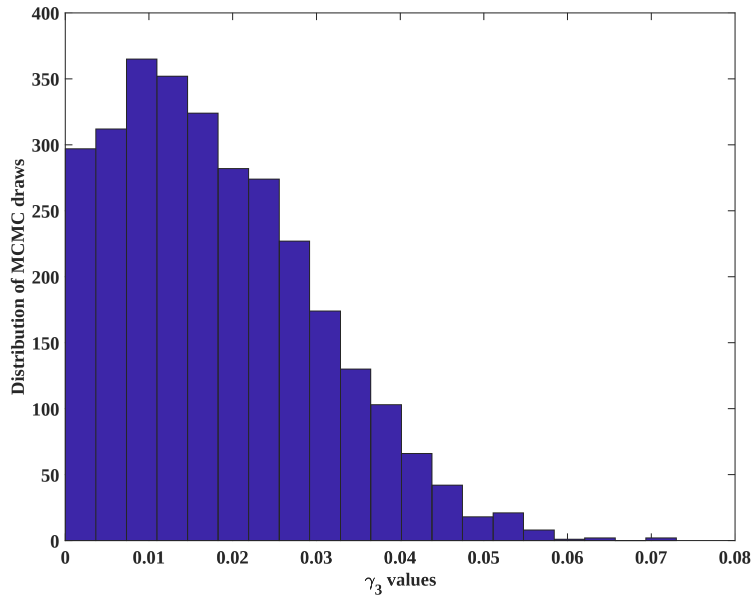

Figure 3.2: Distribution of MCMC draws for γ_3

an estimate of the log-marginal likelihood for each model, which can be used to decide which model is the best, given our sample data and alternative candidate weight matrices. Given the log-marginal likelihood for a model M_q, that we denote $LogM_q$, we can calculate its associated posterior probability for the qth model, namely $prob(M_q) = \exp(LogM_q)/\sum_{q=1}^{Q} \exp(LogM_q)$ (in the case of Q different models). The highest $prob(M_q)$ answers the question — which model is most consistent with the sample data $(y, X, W_1, W_2, \ldots, W_q)$? — *unconditional* on any parameter value/estimates. The answer is unconditional with respect to the parameter values because the parameters have been integrated out of the joint likelihood to produce the log-marginal likelihood and associated model probabilities. We will demonstrate this later.

For the true 2 $W-$matrices model, the log-marginal likelihood is -110508.6217, and for the model based on 3 $W-$matrices it is -110513.2487, which leads to a probability of 0.9903 in favor of the (true) 2 $W-$model and 0.0097 in favor of the (incorrect) 3 $W-$model. These probabilities can be found using the *model_probs()* function from the toolbox that takes a vector of log-marginal likelihoods and returns model probabilities, e.g.,

```
lmarg = [-110508.6217
         -110513.2487];

model_probs(lmarg)
    0.9903
    0.0097
```

3.1.2 Limitations of the convex combination of weights models

In the remainder of this chapter we describe how convex combinations of weight matrices models can be used in other spatial regression specifications. However, before we do this, it is important to note some potential issues that can arise when using these models in applied settings.

One point is that when attempting to draw distinctions between different spatial weight matrices, we need a sufficient sample size of regions N to allow meaningful statistical conclusions. The time dimension (T) of our panel is not likely to be as important as the N dimension when attempting to determine the relative importance or unimportance of different matrices W_m.

A second point is that in the absence of spatial dependence, it will be impossible to determine which weight matrix is relevant, because a lack of spatial dependence implies no need for *any* weight matrix.

A third point is that attempting to draw distinctions between two weight matrices that are very similar, with minor differences arising in a few row- or column-elements is not likely to be possible.

We illustrate this issues in this section using applied examples, where the data is generated with varying: 1) sample size N, 2) levels of spatial dependence ρ, and 3) similarity of weight matrices.

3.1.3 The role played by N, the number of regions

The program below generates y using the SAR specification based on 4 $W-$matrices with a sample of $N = 48$ states, and $T = 20$ years. The W_1 is based on state-to-state commodity flows during 2017, with W_2 reflecting miles of border in common between states, and W_3 a contiguity matrix assigning equal weight to all states with borders that touch, and W_4 based on inverse distance weights given to the 6 nearest neighboring states.

A question we need to ask is — how similar are these matrices? LeSage and Pace (2014) propose converting the weight matrix to a vector using a matrix-vector product involving the weight matrices W_a, W_b and random normal vector u, then finding the correlation between $corr(W_a u, W_b u)$. This is done in *sar_conv_panel_gd2* program, producing results shown below.

```
Correlation        Wcom    Wborder      Wcont      Wdist
Wcom             1.0000     0.7647     0.8187     0.5824
Wborder          0.7647     1.0000     0.9213     0.8411
Wcont            0.8187     0.9213     1.0000     0.7246
Wdist            0.5824     0.8411     0.7246     1.0000
```

Consider that if two matrices W_a, W_b are identical and we attempt to estimate the $\gamma, 1-\gamma$ parameter weights assigned to W_a, W_b, an answer of $\gamma = 0.5$ would produce the same W_c matrix as would the answer of $\gamma = 1$, or $\gamma = 0$. This means that we have an ill-defined problem that has no solution, e.g., we can say that the parameter γ in this case is *unidentified*. Intuitively, this suggests that for cases where the matrices W_a, W_b are *very similar*, we will have difficulty producing valid estimates for the parameter γ. We will explore this in more detail in the next section.

Here we note that the correlation between $Wborder$ and $Wcont$ is 0.9213, which makes sense because the contiguity matrix provides equal weight to all bordering states, while the border miles matrix assigns varying weights to each contiguous state based on miles of border in common. The non-zero elements in both matrices are identical, but the weights given to the non-zero elements are different. There is also a high correlation between the weight matrix based on inverse distance ($Wdist$), with a 6 nearest neighbors cut-off and $Wborder, Wcont$. On average the number of neighbors to each state is around 6, so the inverse-distance matrix would of course be highly correlated with the contiguity ($Wcont$) and border miles matrices ($Wborder$), which we see as 0.7246 and 0.8411, respectively.

Another point is that you should use *sparse* weight matrices, those that contain a large number of zero elements. An inverse distance matrix would contain non-zero elements for entries in the matrix W, and this will adversely impact the use of the 4th-order Taylor series approximation used to calculate log-determinants when estimating these models. Simply put, don't use full $W-$matrices, as they are likely to produce erroneous estimation results.

The program produces three different sets of estimates, one based on the specification estimated by the *sar_panel_FE_g()* function, where we rely on the true W_c matrix, which of course would not be known in applied practice. The estimates from this model are included to show that given accurate estimates of the parameters $\gamma_m, m = 1, \ldots, 4$, we would be able to produce accurate estimates of the model parameters β, ρ, σ^2 and the associated direct and indirect effects estimates.

The second set of estimates attempts to produce estimates for the parameters $\gamma_m, m = 1, \ldots, 4$, using the *sar_conv_panel_g()* estimation function. The third set of estimates reflects a situation where a user omits two of the four (true) weight matrices that were involved in the data generating process (DGP). The results from this set of estimates should shed light on the impact of excluding relevant weight matrices. We already noted in the previous section that inclusion of irrelevant weight matrices has a potential to bias the γ estimates, so this set of estimates is focused on the opposite case where we exclusion relevant weight matrices from consideration.

```
% sar_conv_panel_gd2 demo file
clear all;
rng(10203040);

map_results = shape_read('../demo_data/usstates49');
latt = [map_results.data(1:8,2)
       map_results.data(10:end,2)]; % skip Washington DC
long = [map_results.data(1:8,3)
       map_results.data(10:end,3)];

n = length(latt);
t = 20;

W6 = make_neighborsw(latt,long,6);% 6 nearest neighbors W

Wdist = distance(latt,long) + eye(n);
Winv_distance = zeros(n,n);
dmax = max(max(Wdist));
for i=1:n
    for j=1:n
        if Wdist(i,j) ~= 0
            Winv_distance(i,j) = 1/Wdist(i,j);
        else
            Winv_distance(i,j) = 1/dmax;
        end
    end
end
% create inverse distance with a 6 neighbor cut-off W
Wtmp = Winv_distance.*W6;
Winv_distance = normw(Wtmp);
```

```
% state-to-state commodity flows, 2017
[a,b] = xlsread('../demo_data/cflows_2017.xlsx');
% set main diagonal (intrastate flows) to zero
diaga = diag(a);
W = a - diag(diaga);

Wcom_flows = normw(W); % row-normalize
% eliminate small elements
for i=1:n
    for j=1:n
        if Wcom_flows(i,j) < 0.005
            Wcom_flows(i,j) = 0;
        end
    end
end

Wcom_flows = normw(Wcom_flows);

% miles of common borders between states
[a,b] = xlsread('../demo_data/states_borders.xlsx');
snames = strvcat(b(2:end,:));
% only upper triangle
Wmiles = a(:,2:end);
% make it symmetric
for i=1:48
    for j=1:48
        if Wmiles(i,j) > 0
            Wmiles(j,i) = Wmiles(i,j);
        end
    end
end
Wborder_miles = normw(Wmiles); % row-normalize

% 48 x 48 binary contiguity matrix for states
[a,b] = xlsread('../demo_data/Wcont48.xlsx');
Wcontiguity = normw(a);

rho = 0.7;
k = 2;
x = randn(n*t,k);
beta = ones(k,1);
sige = 0.1;
evec = randn(n*t,1)*sqrt(sige);

gamma1 = 0.2;
gamma2 = 0.5;
gamma3 = 0.1;
gamma4 = 0.2;

Wc = gamma1*kron(eye(t),Wcom_flows) + gamma2*kron(eye(t),Wborder_miles) + ...
    gamma3*kron(eye(t),Wcontiguity)+ gamma4*kron(eye(t),Winv_distance);

% add fixed effects to the DGP
```

```
tts = (1:n)*(1/n);
SFE = kron(ones(t,1),tts');
ttt = (1:t)*(1/t);
TFE = kron(ttt',ones(n,1));

y = (speye(n*t) - rho*Wc)\(x*beta + SFE + TFE + evec);

ndraw = 20000;
nomit = 2000;
prior.model = 3;
prior.novi_flag = 1;
prior.thin = 4;
prior.plt_flag = 1;

Wtrue = gamma1*Wcom_flows + gamma2*Wborder_miles + ...
    gamma3*Wcontiguity + gamma4*Winv_distance;

result1 = sar_panel_FE_g(y,x,Wtrue,t,ndraw,nomit,prior);
vnames = strvcat('y','x1','x2');
fprintf(1,'Estimates based on true Wc \n');
prt_panel(result1,vnames);

Wmatrices = [kron(eye(t),Wcom_flows) kron(eye(t),Wborder_miles) kron(eye(t),Wcontiguity) ...
 kron(eye(t),Winv_distance)];

result2 = sar_conv_panel_g(y,x,Wmatrices,n,t,ndraw,nomit,prior);
fprintf(1,'Estimates based on 4 W-matrices \n');
prt_panel(result2,vnames);

Wmatrices = [kron(eye(t),Wcom_flows) kron(eye(t),Wborder_miles)];

result3 = sar_conv_panel_g(y,x,Wmatrices,n,t,ndraw,nomit,prior);
fprintf(1,'Estimates based on 2 W-matrices \n');
prt_panel(result3,vnames);
```

The printed results are shown below, where we see estimates based on the true $W-$matrix are very close to the true values from the DGP. The second set of estimates that rely on the *sar_conv_panel_g()* function are more different from the true values (which I have inserted into the printout), than in the previous example where the $N = 3,111$. In addition, the $r-$squared reported for these estimates is very high (around 0.97), for this data generated sample, whereas the $r-$squared reported for the previous estimates based on $N = 3,111$ was around 0.80.

In terms of specifics, the estimation approach taken by the function *sar_conv_panel_g()* relies on Taylor series approximations for terms involving the log-determinant terms needed for estimation that are a function of the $W-$matrices. These approximations work better for large N than for small N samples. Increases in the time dimension of the sample data (T) do not improve the accuracy of the Taylor series approximations.

```
% results from sar_conv_panel_gd2.m
Estimates based on true Wc
Homoscedastic model
MCMC SAR model with both region and time period fixed effects
Dependent Variable =              y
R-squared          =    0.9784
```

```
corr-squared        =    0.9379
sigma^2             =    0.0947
Nobs,Nvar,#FE       =    960,     2,    68
ndraw,nomit         =  20000,  2000
rvalue              =      0
min and max rho     =   -1.0000,   1.0000
total time in secs  =    6.7100
time for lndet      =    0.0380
time for MCMC draws =    5.9220
Pace and Barry, 1999 MC lndet approximation used
order for MC appr  =     50
iter  for MC appr  =     30
***************************************************************
Variable      Coefficient  Asymptot t-stat    z-probability
x1               0.983175       93.064061         0.000000
x2               1.003347       96.953417         0.000000
rho              0.703282       60.620586         0.000000

Direct        Coefficient           t-stat           t-prob         lower 05         upper 95
x1               1.133271       78.778148         0.000000         1.105363         1.161901
x2               1.156522       81.160259         0.000000         1.128949         1.184767

Indirect      Coefficient           t-stat           t-prob         lower 05         upper 95
x1               2.185303       17.544238         0.000000         1.954461         2.440836
x2               2.230136       17.574211         0.000000         1.993706         2.488598

Total         Coefficient           t-stat           t-prob         lower 05         upper 95
x1               3.318573       24.620975         0.000000         3.069923         3.594427
x2               3.386658       24.699504         0.000000         3.131133         3.666953

Estimates based on 4 W-matrices
MCMC SAR convex combination W model with both region and time period fixed effects
Homoscedastic model
Bayesian spatial autoregressive convex W model
Dependent Variable  =                y
Log-marginal        = -631.2005
Log-marginal MCerror=  0.045167
R-squared           =    0.9785
corr-squared        =    0.9418
mean of sige draws  =    0.0940
posterior mode sige =    0.0932
Nobs, Nvars         =    960,     2
ndraws,nomit        =  20000,  2000
time for effects    =    0.4210
time for sampling   =    3.2100
time for Taylor     =    0.0979
thinning for draws  =         4
min and max rho     =   -1.0000,   1.0000
***************************************************************
      MCMC diagnostics ndraws = 4500
Variable         mode         mean     MC error        tau     Geweke
x1             0.9800       0.9803   0.00018545   0.967230   0.999135
x2             1.0024       1.0026   0.00017096   0.953358   0.999276
rho            0.6602       0.6597   0.00129213  13.312864   0.998332
gamma1         0.1040       0.1006   0.00224078  14.172586   0.949563
```

```
gamma2         0.5951       0.5885   0.00321036  12.963978   0.987634
gamma3         0.1180       0.1242   0.00213079   7.373106   0.913128
gamma4         0.1828       0.1868   0.00059554   2.113492   0.990473

***************************************************************
      Posterior Estimates
Variable      Coefficient  Asymptot t-stat    z-probability
x1               0.980289       93.825674         0.000000
x2               1.002580       96.393547         0.000000
rho              0.659715       33.084306         0.000000     True values
gamma1           0.100572        2.797013         0.005158     gamma1 = 0.2;
gamma2           0.588474       11.285973         0.000000     gamma2 = 0.5;
gamma3           0.124188        2.746916         0.006016     gamma3 = 0.1;
gamma4           0.186767        7.107260         0.000000     gamma4 = 0.2;

Direct        Coefficient          t-stat           t-prob         lower 05         upper 95
x1               1.125965       78.901695         0.000000         1.098878         1.154334
x2               1.151571       79.804464         0.000000         1.123808         1.180083

Indirect      Coefficient          t-stat           t-prob         lower 05         upper 95
x1               1.764863       10.648888         0.000000         1.479030         2.122434
x2               1.804977       10.667407         0.000000         1.512525         2.168811

Total         Coefficient          t-stat           t-prob         lower 05         upper 95
x1               2.890828       16.538864         0.000000         2.590671         3.272787
x2               2.956547       16.582821         0.000000         2.648076         3.339394

Estimates based on 2 W-matrices
MCMC SAR convex combination W model with both region and time period fixed effects
Homoscedastic model
Bayesian spatial autoregressive convex W model
Dependent Variable  =              y
Log-marginal        = -656.4057
Log-marginal MCerror=  0.020039
R-squared           =    0.9770
corr-squared        =    0.9377
mean of sige draws  =    0.1005
posterior mode sige =    0.0999
Nobs, Nvars         =    960,     2
ndraws,nomit        =  20000,  2000
time for effects    =    0.4110
time for sampling   =    2.7570
time for Taylor     =    0.1459
thinning for draws  =         4
min and max rho     =   -1.0000,   1.0000
***************************************************************
      MCMC diagnostics ndraws = 4500
Variable         mode         mean     MC error        tau     Geweke
x1             0.9799       0.9798   0.00015719   0.944789   0.998916
x2             1.0031       1.0031   0.00011286   0.995368   0.999751
rho            0.6304       0.6287   0.00083946   7.617887   0.994979
gamma1         0.1407       0.1369   0.00166745   7.792185   0.942754
gamma2         0.8593       0.8631   0.00166745   7.792185   0.990906

***************************************************************
```

Posterior Estimates

Variable	Coefficient	Asymptot t-stat	z-probability	
x1	0.979824	90.246319	0.000000	
x2	1.003101	93.717640	0.000000	
rho	0.628689	31.159792	0.000000	True values
gamma1	0.136938	3.604256	0.000313	gamma1 = 0.2;
gamma2	0.863062	22.716097	0.000000	gamma2 = 0.5;

Direct	Coefficient	t-stat	t-prob	lower 05	upper 95
x1	1.111668	79.869587	0.000000	1.084308	1.138778
x2	1.138077	82.091504	0.000000	1.110969	1.165349

Indirect	Coefficient	t-stat	t-prob	lower 05	upper 95
x1	1.534994	10.965408	0.000000	1.286224	1.825756
x2	1.571431	11.001428	0.000000	1.317222	1.870888

Total	Coefficient	t-stat	t-prob	lower 05	upper 95
x1	2.646662	17.916197	0.000000	2.383662	2.955041
x2	2.709507	18.012026	0.000000	2.438515	3.024371

The last set of estimates exhibit the expected impact of biasing downward the γ_1 estimate, and biasing upward the γ_2 coefficient estimate. We will have more to say later about how to determine relevant and irrelevant weight matrices for these models.

To illustrate the point made earlier about not using a full inverse-distance matrix, the estimates shown below were produced using this matrix in conjunction with the other three matrices (see: sar_conv_panel_gd2a.m file). The full inverse-distance matrix was used in the DGP to produce the $y-$variable outcomes, so if the estimation procedure were working well, we should see estimates close to the true parameter values used in the DGP. Instead, we see a set of estimates far from the true values of the parameters.

Posterior Estimates

Variable	Coefficient	Asymptot t-stat	z-probability	
x1	0.815589	52.512236	0.000000	
x2	0.833507	52.746188	0.000000	
rho	0.767922	41.324731	0.000000	
gamma1	0.052483	2.037284	0.041622	gamma1 = 0.2;
gamma2	0.391602	12.521630	0.000000	gamma2 = 0.5;
gamma3	0.042280	1.624184	0.104337	gamma3 = 0.1;
gamma4	0.513635	23.159930	0.000000	gamma4 = 0.2;

3.1.4 The role played by ρ, spatial dependence

It should be clear that if $\rho = 0$, then no spatial weight matrix is appropriate, so estimation of the convex combination weights $\gamma_m, m = 1, \ldots, M$ would be unidentified. The demonstration program here illustrates the point that in the face of low levels of spatial dependence (e.g., $\rho = 0.1$ or $\rho = 0.2$, we will have less success in estimating the parameters γ_m.

The program produces two sets of estimates, one based on $\rho = 0.1$ and a second with $\rho = -0.2$, where we would expect improved estimates associated with the larger level of spatial dependence of $\rho = -0.2$. The DGP uses the sample of $N = 3,111$ US counties, which produced very accurate estimates in the case of $\rho = 0.6$ and the *sar_conv_panel_gd.m* demonstration file, when the correct three weight matrices were used as input to the function.

```
% sar_conv_panel_gd3 demo file
clear all;
rng(10203444);

% read an Arcview shape file for 3,111 US counties
map_results = shape_read('../demo_data/uscounties_projected');
latt = map_results.data(:,3);
long = map_results.data(:,4);
n = length(latt);
t = 20;

[j,Wcont,j] = xy2cont(latt,long); % Delaunay contiguity W

W6 = make_neighborsw(latt,long,6);% 6 nearest neighbors W

Wdist = distance(latt,long) + eye(n);
% create inverse distance with a 6 neighbor cut-off W
Wcut = (ones(n,n)./Wdist).*W6;
Wdist = normw(Wcut);

rho = 0.1;
k = 2;
x = randn(n*t,k);
beta = ones(k,1);
sige = 1;
evec = randn(n*t,1)*sqrt(sige);

gamma1 = 0.3;
gamma2 = 0.6;
gamma3 = 0.1;

Wc = gamma1*kron(eye(t),Wcont) + gamma2*kron(eye(t),W6) + gamma3*kron(eye(t),Wdist);

% add fixed effects to the DGP
tts = (1:n)*(1/n);
SFE = kron(ones(t,1),tts');
ttt = (1:t)*(1/t);
TFE = kron(ttt',ones(n,1));

y = (speye(n*t) - rho*Wc)\(x*beta + SFE + TFE + evec);

ndraw = 50000;
nomit = 20000;
prior.model = 3;
prior.novi_flag = 1;
prior.thin = 10;
% prior.plt_flag = 1;

Wmatrices = [kron(eye(t),Wcont) kron(eye(t),W6) kron(eye(t),Wdist)];

result1 = sar_conv_panel_g(y,x,Wmatrices,n,t,ndraw,nomit,prior);
vnames = strvcat('y','x1','x2');
fprintf(1,'sar_conv_panel_g with rho = 0.1 \n');
prt_panel(result1,vnames);
```

```
rho = -0.2;
k = 2;
x = randn(n*t,k);
beta = ones(k,1);
sige = 1;
evec = randn(n*t,1)*sqrt(sige);

gamma1 = 0.3;
gamma2 = 0.6;
gamma3 = 0.1;

Wc = gamma1*kron(eye(t),Wcont) + gamma2*kron(eye(t),W6) + gamma3*kron(eye(t),Wdist);

% add fixed effects to the DGP
tts = (1:n)*(1/n);
SFE = kron(ones(t,1),tts');
ttt = (1:t)*(1/t);
TFE = kron(ttt',ones(n,1));

y = (speye(n*t) - rho*Wc)\(x*beta + SFE + TFE + evec);

Wmatrices = [kron(eye(t),Wcont) kron(eye(t),W6) kron(eye(t),Wdist)];

result2 = sar_conv_panel_g(y,x,Wmatrices,n,t,ndraw,nomit,prior);
vnames = strvcat('y','x1','x2');
fprintf(1,'sar_conv_panel_g with rho = -0.2 \n');
prt_panel(result2,vnames);
```

The results printed below show that for the case of $\rho = 0.1$, the estimates for the parameters $\gamma_m, m = 1,2,3$ are far from the true values (which I inserted into the printout shown below). Nonetheless, the estimates are reasonable, preserving the rank ordering of γ_3 being the smallest estimate, γ_2 the largest, with γ_1 in between. We also see a right-skewed distribution for the smallest $\gamma_3 = 0.1$ estimate, where the mean of the retained, thinned, MCMC draws was 0.1828, which was above the modal estimate of 0.1631.

For the estimation results based on a DGP that set $\rho = -0.2$, the accuracy of the estimates for $\gamma_m, m = 1,2,3$ improved, coming closer to the true values of 0.3, 0.6, 0.1 for $\gamma_m, m = 1,2,3$, respectively.

Of course with a larger sample size N, we are able to produce reasonably accurate estimates for the γ parameters, than if we were dealing with a small N. There is of course, a trade-off between the three facets of the underlying model we are discussing here. In addition, there is an important role played by the signal/noise ratio of the model relationship.

```
sar_conv_panel_g with rho = 0.1
MCMC SAR convex combination W model with both region and time period fixed effects
Homoscedastic model
Bayesian spatial autoregressive convex W model
Dependent Variable  =               y
Log-marginal        = -108519.5982
Log-marginal MCerror=  0.022787
R-squared           =    0.7065
corr-squared        =    0.6684
mean of sige draws  =    0.9557
posterior mode sige =    0.9556
```

```
Nobs, Nvars         =  62220,     2
ndraws,nomit        =  50000, 20000
time for effects    =    9.2970
time for sampling   =   98.1310
time for Taylor     =    4.2535
thinning for draws  =        10
min and max rho     =   -1.0000,   1.0000
****************************************************************
      MCMC diagnostics ndraws = 3000
Variable         mode         mean     MC error        tau    Geweke
x1             1.0022       1.0023   0.00005532   0.953481   0.999664
x2             1.0047       1.0047   0.00004954   1.013990   0.999915
rho            0.1038       0.1030   0.00010267   1.158201   0.996025
gamma1         0.3935       0.3902   0.00231873   1.363870   0.986113
gamma2         0.4434       0.4270   0.00261366   1.566423   0.984810
gamma3         0.1631       0.1828   0.00130947   1.122705   0.994076

****************************************************************
      Posterior Estimates
Variable      Coefficient  Asymptot t-stat    z-probability
x1               1.002252       250.137218         0.000000
x2               1.004707       247.023927         0.000000
rho              0.102998        21.020123         0.000000
gamma1           0.390168         3.769692         0.000163 (true = 0.3)
gamma2           0.427039         3.558998         0.000372 (true = 0.6)
gamma3           0.182793         1.994414         0.046107 (true = 0.1)

Direct        Coefficient           t-stat           t-prob        lower 05        upper 95
x1               1.003981       249.829351         0.000000        1.011689        0.996361
x2               1.006440       246.964584         0.000000        1.014431        0.998196

Indirect      Coefficient           t-stat           t-prob        lower 05        upper 95
x1               0.113388        19.010144         0.000000        0.125097        0.101967
x2               0.113665        19.045567         0.000000        0.125402        0.102285

Total         Coefficient           t-stat           t-prob        lower 05        upper 95
x1               1.117369       147.136916         0.000000        1.132109        1.103243
x2               1.120104       148.234541         0.000000        1.134544        1.105272

sar_conv_panel_g with rho = -0.2
MCMC SAR convex combination W model with both region and time period fixed effects
Homoscedastic model
Bayesian spatial autoregressive convex W model
Dependent Variable  =              y
Log-marginal        = -108682.1053
Log-marginal MCerror=  0.016688
R-squared           =    0.6997
corr-squared        =    0.6658
mean of sige draws  =    0.9567
posterior mode sige =    0.9565
Nobs, Nvars         =  62220,     2
ndraws,nomit        =  50000, 20000
time for effects    =    9.1270
time for sampling   =   98.2520
time for Taylor     =    4.3758
```

```
thinning for draws  =        10
min and max rho     =   -1.0000,   1.0000
***************************************************************
      MCMC diagnostics ndraws = 3000
Variable         mode         mean     MC error        tau     Geweke
x1             0.9946       0.9946   0.00007915   1.037830   0.999270
x2             1.0023       1.0022   0.00006583   0.975545   0.999808
rho           -0.2033      -0.2026   0.00011495   1.059074   0.999973
gamma1         0.2562       0.2539   0.00124730   1.535486   0.988550
gamma2         0.6813       0.6664   0.00171263   1.524198   0.999144
gamma3         0.0625       0.0797   0.00105468   1.049602   0.957236

***************************************************************
      Posterior Estimates
Variable      Coefficient  Asymptot t-stat    z-probability
x1               0.994645       249.222430         0.000000
x2               1.002217       250.934610         0.000000
rho             -0.202623       -37.230492         0.000000
gamma1           0.253906         4.699115         0.000003 (true = 0.3)
gamma2           0.666404        10.828052         0.000000 (true = 0.6)
gamma3           0.079690         1.834610         0.066564 (true = 0.1)

Direct        Coefficient           t-stat           t-prob         lower 05          upper 95
x1               1.000405       249.132908         0.000000         1.008235          0.992561
x2               1.008020       251.398312         0.000000         1.015933          1.000186

Indirect      Coefficient           t-stat           t-prob         lower 05          upper 95
x1              -0.173324       -42.460226         0.000000        -0.165199         -0.181094
x2              -0.174643       -42.702625         0.000000        -0.166615         -0.182609

Total         Coefficient           t-stat           t-prob         lower 05          upper 95
x1               0.827081       162.991019         0.000000         0.837200          0.817124
x2               0.833377       160.942916         0.000000         0.843655          0.823428
```

3.1.5 The role played by similarity of the $W_m, m = 1, \ldots, M$ weight matrices

The program below shows two sets of estimates, one based on a model where 2 $W-$matrices were used constructed one based on 6 nearest neighbors and the other on 7 nearest neighbors. A second set of estimates based on a model where the 2 $W-$matrices were used constructed one based on 6 nearest neighbors and the other on 20 nearest neighbors. The correlation between the three matrices was calculated and shown below (and in the printed results from the program).

What we see is a high correlation between $W6, W7$ and a lower correlation between $W6, W20$.

```
Correlation         W6         W7        W20
W6              1.0000     0.9287     0.5459
W7              0.9287     1.0000     0.5849
W20             0.5459     0.5849     1.0000
```

```
% file: sar_conv_panel_gd4
clear all;
rng(10203444);

% read an Arcview shape file for 3,111 US counties
```

```
map_results = shape_read('../demo_data/uscounties_projected');
latt = map_results.data(:,3);
long = map_results.data(:,4);
n = length(latt);
t = 20;

W6 = make_neighborsw(latt,long,6);% 6 nearest neighbors W
W7 = make_neighborsw(latt,long,7);% 7 nearest neighbors W
W20 = make_neighborsw(latt,long,20);% 20 nearest neighbors W

u = randn(n,1);
corr= corrcoef([W6*u W7*u W20*u]);

% find correlation between W-matrices
inc.cnames = strvcat('W6','W7','W20');
inc.rnames = strvcat('Correlation','W6','W7','W20');
mprint(corr,inc);

rho = 0.6;
k = 2;
x = randn(n*t,k);
beta = ones(k,1);
sige = 100;
evec = randn(n*t,1)*sqrt(sige);

gamma1 = 0.3;
gamma2 = 0.7;

Wc = gamma1*kron(eye(t),W6) + gamma2*kron(eye(t),W7);

% add fixed effects to the DGP
tts = (1:n)*(1/n);
SFE = kron(ones(t,1),tts');
ttt = (1:t)*(1/t);
TFE = kron(ttt',ones(n,1));

y = (speye(n*t) - rho*Wc)\(x*beta + SFE + TFE + evec);

ndraw = 20000;
nomit = 10000;
prior.model = 3;
prior.novi_flag = 1;
prior.thin = 4;
% prior.plt_flag = 1;

Wmatrices = [kron(eye(t),W6) kron(eye(t),W7)];
result1 = sar_conv_panel_g(y,x,Wmatrices,n,t,ndraw,nomit,prior);
vnames = strvcat('y','x1','x2');
fprintf(1,'sar_conv_panel_g with W6, W7 \n');
prt_panel(result1,vnames);

Wc = gamma1*kron(eye(t),W6) + gamma2*kron(eye(t),W20);
y = (speye(n*t) - rho*Wc)\(x*beta + SFE + TFE + evec);

Wmatrices = [kron(eye(t),W6) kron(eye(t),W20)];
```

```
result2 = sar_conv_panel_g(y,x,Wmatrices,n,t,ndraw,nomit,prior);
vnames = strvcat('y','x1','x2');
fprintf(1,'sar_conv_panel_g with W6, W20 \n');
prt_panel(result2,vnames);
```

Estimation results from the two models (and two separate DGP's) are shown in the printout results from running the program below. For the case of matrices $W6, W7$ the correlation between $W6 \times u$, $W7 \times u$, is 0.9287, and the resulting estimates for $\gamma_1 = 0.2191, \gamma_2 = 0.7809$ are not equal to the true values used in the DGP, which were $\gamma_1 = 0.3, \gamma_2 = 0.7$. Of course, attempting to distinguish between very similar W matrices creates a problem as motivated earlier because the strength of identification diminishes.

The resulting estimates based on a DGP that involved $W6, W20$, where the correlation between $W6, W20$ was only 0.5459 are very accurate. These estimates are shown in the second set of printed output.

```
% results from sar_conv_panel_gd4.m
Correlation         W6         W7        W20
W6              1.0000     0.9287     0.5459
W7              0.9287     1.0000     0.5849
W20             0.5459     0.5849     1.0000

sar_conv_panel_g with W6, W7
MCMC SAR convex combination W model with both region and time period fixed effects
Homoscedastic model
Bayesian spatial autoregressive convex W model
Dependent Variable  =               y
Log-marginal        = -253623.8274
Log-marginal MCerror=  0.021563
R-squared           =    0.2924
corr-squared        =    0.0202
mean of sige draws  =   95.2656
posterior mode sige =   95.2720
Nobs, Nvars         =  62220,     2
ndraws,nomit        =  20000, 10000
time for effects    =    9.1910
time for sampling   =   37.9660
time for Taylor     =    1.2763
thinning for draws  =         4
min and max rho     =   -1.0000,   1.0000
***************************************************************
      MCMC diagnostics ndraws = 2500
Variable         mode         mean     MC error        tau     Geweke
x1             1.0139       1.0141   0.00089876   0.938208   0.994143
x2             1.0436       1.0423   0.00082928   1.025756   0.998273
rho            0.6252       0.6252   0.00010213   1.404443   0.999277
gamma1         0.2192       0.2191   0.00051931   1.732610   0.985503
gamma2         0.7808       0.7809   0.00051931   1.732610   0.995956
***************************************************************
      Posterior Estimates
Variable      Coefficient  Asymptot t-stat    z-probability
x1               1.014084       24.930706         0.000000
x2               1.042304       25.748123         0.000000
rho              0.625159      123.624690         0.000000
gamma1           0.219083        6.025388         0.000000   (true = 0.3)
```

```
gamma2           0.780917        21.477343        0.000000   (true = 0.7)

Direct        Coefficient           t-stat          t-prob        lower 05         upper 95
x1               1.097121        24.930653        0.000000        1.182512         1.012940
x2               1.127652        25.738334        0.000000        1.212086         1.043493

Indirect      Coefficient           t-stat          t-prob        lower 05         upper 95
x1               1.608719        22.130649        0.000000        1.749509         1.464813
x2               1.653494        22.631002        0.000000        1.797154         1.513060

Total         Coefficient           t-stat          t-prob        lower 05         upper 95
x1               2.705840        23.759206        0.000000        2.926767         2.484961
x2               2.781147        24.411944        0.000000        2.999419         2.563710

sar_conv_panel_g with W6, W20
MCMC SAR convex combination W model with both region and time period fixed effects
Homoscedastic model
Bayesian spatial autoregressive convex W model
Dependent Variable  =              y
Log-marginal        = -252629.0650
Log-marginal MCerror=  0.018947
R-squared           =    0.1916
corr-squared        =    0.0204
mean of sige draws  =   95.5756
posterior mode sige =   95.5814
Nobs, Nvars         =  62220,     2
ndraws,nomit        =  20000, 10000
time for effects    =   14.8390
time for sampling   =   37.5810
time for Taylor     =    2.9291
thinning for draws  =        4
min and max rho     =   -1.0000,   1.0000
***************************************************************
      MCMC diagnostics ndraws = 2500
Variable         mode         mean     MC error        tau     Geweke
x1             1.0156       1.0154   0.00087517   1.011294   0.999831
x2             1.0446       1.0446   0.00076213   0.939938   0.999535
rho            0.6207       0.6203   0.00023069   1.964437   0.999998
gamma1         0.2999       0.3002   0.00041083   2.033840   0.999969
gamma2         0.7001       0.6998   0.00041083   2.033840   0.999987
***************************************************************
      Posterior Estimates
Variable      Coefficient  Asymptot t-stat    z-probability
x1               1.015410        24.730612        0.000000
x2               1.044645        26.153533        0.000000
rho              0.620329        80.342719        0.000000
gamma1           0.300183        20.934138        0.000000  (true = 0.3)
gamma2           0.699817        48.803794        0.000000  (true = 0.7)

Direct        Coefficient           t-stat          t-prob        lower 05         upper 95
x1               1.053595        24.697950        0.000000        1.136681         0.973123
x2               1.083927        26.155394        0.000000        1.163654         1.003891

Indirect      Coefficient           t-stat          t-prob        lower 05         upper 95
x1               1.622027        18.914418        0.000000        1.788573         1.459656
```

```
x2              1.668636      19.936362       0.000000       1.827665       1.509185

Total        Coefficient         t-stat         t-prob       lower 05       upper 95
x1              2.675622      21.817339       0.000000       2.919073       2.445351
x2              2.752562      23.180227       0.000000       2.978779       2.528143
```

3.2 Chapter Summary

We introduced a SAR model specification that allows for use of convex combinations of spatial weight matrices. Additional specifications such as the SDM, SEM, and SDEM models will be introduced with examples in the next chapter.

These models hold the possibility to greatly relax the typical assumption made in spatial regression models that focus on interaction/dependence between regions located in space to more general notions of connectivity, for example connections based on trade flows, migration flows, etc. Of course, in reality there may be numerous types of connectivity at work, and the convex combination of multiple weight matrices specification described here allows us to model multiple sources of connectivity.

Since there are new methods that require more effort to estimate users must be careful not to misuse the methods. Some feel for issues that can arise with these models has been provided in this chapter, with more discussion of issues and useful solutions in the next chapters. Things discussed here were: 1) sample size requirements, 2) strength of dependence, and 3) the introduction of similar matrices in these models. We demonstrated that an adequate sample size (specifically the $N-$dimension) is important if you wish to produce accurate estimates and inferences regarding the relative importance of different types of connectivity. The level of dependence is also a factor that determines whether these models can be estimated successfully. Finally, it makes no sense to introduce multiple weight matrices that are similar, just as it makes no sense to introduce multiple explanatory variables that are similar. We demonstrated that similarity of weight matrices can be determined by creating vectors: $W_a u, W_b u$ consisting of matrix-vector products, where W_a, W_b are $N \times N$ matrices and u is an $N \times 1$ random normal vector. Given the two vectors, $W_a u, W_b u$ a simple correlation coefficient can be used to pass judgment on similarity of the matrices.

3.3 Chapter references

Anderson, J.E., van Wincoop E. 2003. Gravity with gravitas: A solution to the border puzzle *American Economic Review*, 93, 170-192.

Anderson, J.E., van Wincoop E. 2004. Trade costs, *Journal of Economic Literature*, 42, 691-751.

Behrens, K., Ertur, C., Koch, W. 2012. Dual gravity: using spatial econometrics to control for multilateral resistance. *Journal of Applied Econometrics*, 27, 773-794.

Blankmeyer, E., J.P. LeSage, J.R. Stutzman, K.J. Knox, R.K. Pace, 2011. Peer group dependence in salary benchmarking: A statistical model, *Managerial and Decision Economics*, 32(2), 91-104.

Debarsy N., J.P. LeSage (2018). Flexible dependence modeling using convex combinations of different types of connectivity structures, *Regional Science and Urban Economics* , 69, 48-68.

Debarsy, N. and LeSage, J.P. 2021. Bayesian model averaging for spatial autoregressive models based on convex combinations of different types of connectivity matrices, *Journal of Business & Economic Statistics*

Hazir, C.S., J.P. LeSage, C. Autant-Bernard. 2018. The role of R&D collaboration networks on regional innovation performance, *Papers in Regional Science*, 97, 549-567.

Koch, W., J.P. LeSage. 2015. Latent multilateral trade resistance indices: theory and evidence, *Scottish Journal of Political Economy*, 62(3) 264-290.

Lacombe, D. (2004). Does Econometric Methodology Matter? An Analysis of Public Policy Using Spatial Econometric Techniques, *Geographical Analysis*, 36, 87-89.

LeSage, J. P. and R. K. Pace. 2014. The biggest myth in spatial econometrics, *Econometrics*, Volume 2, Issue 4, pp. 217-249.
doi:10.3390/econometrics2040217. http://www.mdpi.com/2225-1146/2/4/217

Pace, R.K., J.P. LeSage. (2002). Semiparametric maximum likelihood estimates of spatial dependence, *Geographical Analysis*, 34(1), 75-90.

LeSage, J.P., R.K. Pace. 2011. Pitfalls in higher order model extensions of basic spatial regression methodology, *Review of Regional Studies*, 41(1), 13-26.

Chapter 4

Convex combinations for SDM, SEM, SDEM models

This chapter discusses additional convex combination models, specifically, SDM, SEM, SDEM model specifications. Questions arise regarding models such as the SDM and SDEM specifications that include spatial lags of the explanatory variables, e.g., WX variables.

The SDEM and SDM model specifications raise a potential problem regarding use of the convex combination matrix W_c in the matrix product W_cX that represents characteristics of the explanatory variables from broadly defined neighbors, where $W_c = \gamma_1 W_1 + \ldots + \gamma_M W_M$. To see the nature of the problem, consider the SDEM model shown below, where I have assumed only two W_1, W_2 matrices for simplicity.

$$\begin{aligned}
y &= X\beta + W_c X\theta + \iota_T \otimes \mu + \nu \otimes \iota_N + u && (4.1)\\
u &= \rho W_c u + \varepsilon \\
&= X\beta + \gamma_1 W_1 X\theta + \gamma_2 W_2 X\theta + \iota_T \otimes \mu + \nu \otimes \iota_N + u \\
&= X\beta + W_1 X\pi_1 + W_2 X\pi_2 + \iota_T \otimes \mu + \nu \otimes \iota_N + u && (4.2)\\
W_c &= \gamma_1 W_1 + \gamma_2 W_2 \\
& \quad \pi_1 = \gamma_1\theta, \pi_2 = \gamma_2\theta
\end{aligned}$$

What these expressions show is that the coefficient estimates for the explanatory variables matrices W_1X, W_2X (that I have labeled π_1, π_2) in (4.2) will be restricted to have a relationship. The relationship is such that the estimates for the explanatory variables W_1X will be equal to $\gamma_1\theta$, and those for W_2X will be equal to $\gamma_2\theta$, which is very restrictive. By this, I mean that if we simply estimated the model shown below in (4.3) using the matrices W_1, W_2 to create the *spatial lags* of the explanatory variables, without these (implied) restrictions, the probability of having similar estimated coefficients is zero. That is, the estimated $\hat{\theta}_1, \hat{\theta}_2$ would never be close to the estimated $\hat{\pi}_1, \hat{\pi}_2$.

$$y = X\beta + W_1 X\theta_1 + W_2 X\theta_2 + \iota_T \otimes \mu + \nu \otimes \iota_N + u \qquad (4.3)$$

Furthermore, the approach we use to isolate the dependence parameters and sample from the joint posterior distribution of these no longer works. This is because the parameters γ_1, γ_2 now appear in the W_cX variables. It would not longer be possible to calculate the log-marginal likelihood as we currently do, since we cannot integrate out the parameters for β, σ^2 as we currently do. This is because the "β" parameters (which now include θ_1, θ_2) depend on the γ_1, γ_2 parameters.

My solution to this problem is to include W_1X, W_2X matrices into the model specification and estimate unrestricted coefficients/parameters for these explanatory variables. These are labeled θ_1, θ_2 in (4.3).

An implication of this approach is that when we go to calculate direct and indirect effects associated with changes in the X variable (variables), we have the expression in (4.4).

$$\partial E(y)/\partial X = (I_{NT}\hat{\beta} + W_1\hat{\theta}_1 + W_2\hat{\theta}_2) \tag{4.4}$$

The direct effects would be $\hat{\beta}$, while the indirect effects are $\hat{\theta}_1 + \hat{\theta}_2$ for this model.

4.1 The SDM convex combination of W model

The SDM convex combination model is shown in (4.5), where each $NT \times NT$ matrix W_m represents some sort of connectivity between regions, and takes a block diagonal form: $I_T \otimes w_m$, where w_m is an $N \times N$ weight matrix, with main diagonal elements equal to zero and row-sums equal to one.

$$\begin{aligned} y &= \rho W_c(\Gamma)y + X\beta + \sum_{m=1}^{M} W_m X\theta_m + \varepsilon, \quad \varepsilon \sim N(0, \sigma^2 I_n), \qquad (4.5) \\ W_c(\Gamma) &= \sum_{m=1}^{M} \gamma_m W_m, \quad 0 \leq \gamma_m \leq 1, \ \sum_{m=1}^{M} \gamma_m = 1, \\ \Gamma &= (\gamma_1, \ldots, \gamma_M)' \end{aligned}$$

The $NT \times k$ matrix X in (4.5) contains exogenous explanatory variables, with β being the associated $k \times 1$ vector of parameters. As will be demonstrated, the function automatically generates and includes the explanatory variables $\sum_{m=1}^{M} W_m X$ and includes them in the model. My thinking in taking this approach is that if you believe in multiple types of connectivity that involve interaction of the dependent variable y, then these different types of connectivity should also reflect *contextual effects* on which the outcomes in y depend. The $NT \times 1$ vector ε represents a constant variance normally distributed disturbance term, and the $NT \times 1$ dependent variable vector y takes the same form as in our models from Chapters 1 and 2. There is no option for use of the variance scalars v_{it} in this model, for reasons of computational speed.

As in the case, of the SAR model, (4.5) can be re-expressed as in (4.6), which is a more computationally convenient expression that isolates the parameters $\rho, \gamma_m, m = 1, \ldots, M$ in the $(M+1) \times 1$ vector ω.

$$\begin{aligned} \tilde{y}\omega &= X\beta + \sum_{m=1}^{M} W_m X\theta_m + \varepsilon \qquad (4.6) \\ \tilde{y} &= (y,\ W_1y,\ W_2y,\ \ldots,\ W_My) \end{aligned}$$

$$\begin{aligned}
\omega &= \begin{pmatrix} 1 \\ -\rho\Gamma \end{pmatrix}, \\
\Gamma &= \begin{pmatrix} \gamma_1 \\ \gamma_2 \\ \vdots \\ \gamma_M \end{pmatrix} \\
\gamma_M &= 1 - \gamma_1 - \gamma_2 - \ldots - \gamma_{M-1}
\end{aligned}$$

The value of isolating the parameter vector ω is that we can pre-calculate the $NT \times (M+1)$ matrix $\tilde{y}$ prior to the beginning of the MCMC sampling loop, since this matrix contains only sample data. This provides another motivation for *not* including the γ parameters in the explanatory variables matrix, as this would eliminate this computational speed advantage.

To estimate this model, we rely on the same Taylor series approximations used for the SAR convex combination model, so models with small N are likely to produce more inaccurate estimates.

The partial derivatives for our SDM convex combination model take the form in (4.7)

$$\partial E(y)/\partial x_r = (I_{NT} - \hat{\rho} W_c(\hat{\Gamma}))^{-1}(I_{NT}\hat{\beta}_r + W_1\hat{\theta}_1 + \ldots + W_M\hat{\theta}_M) \quad (4.7)$$

Note that $(I_{NT} - \hat{\rho} W_c(\hat{\Gamma}))^{-1}$ depends on the estimates for $\gamma_1, \ldots, \gamma_M$. Debarsy and LeSage (2021) discuss adoption of computationally efficient ways to calculate the scalar effects estimates described in LeSage and Pace (2009) for this type of model.

The next section illustrates use of the SDM convex combination estimation functions.

4.1.1 Using the *sdm_conv_panel_g()* function

The program below shows how to use the SDM estimation function for the convex combination of weights model. Note that we provide only the matrix X since the function adds the spatial lags of the explanatory variables $(W_1X, W_2X, \ldots, W_MX)$. You can also call the function with small $N \times N$ matrices W_1, W_2, etc., or with large $NT \times NT$ matrices, with the large matrix input option allowing for different $W-$matrices for each time period. Of course, you should keep in mind the potential problem regarding the scalar summary effects estimates illustrated in Chapter 1.

Estimates produced by the second call to the function *sdm_conv_panel_g()* should produce roughly the same estimates because the same set of matrices W_1, W_2 are used to create the larger $NT \times NT$ inputs, with the MATLAB kronecker product, *kron(eye(t),W1)*. This is of course a way to test for convergence of MCMC sampling estimates, since we should obtain equivalent estimates and inferences from two runs of the MCMC procedure which will rely on different random draws. Setting the random number seed using: *rng(10203040);* ensures that we can replicate the two sets of estimates if we wish to re-estimate a model at a later time.

```
% sdm_conv_panel_gd demo file
clear all;
rng(10203040);
n = 1000;
t = 10;
rho = 0.7;
k = 2;
```

```
x = randn(n*t,k);
beta = [1
        0.5];
theta1 = -0.75*ones(k,1);
theta2 = 0.5*ones(k,1);
bvec = [beta
        theta1
        theta2];

latt = rand(n,1);
long = rand(n,1);
W1 = make_neighborsw(latt,long,2);

latt = rand(n,1); % A different set of latt-long
long = rand(n,1); % coordinates
W2 = make_neighborsw(latt,long,6);

m = 2;
gamma1 = 0.2;
gamma2 = 0.8;

gamma = [gamma1
         gamma2];

Wc = gamma1*kron(eye(t),W1) + gamma2*kron(eye(t),W2);
% add fixed effects to the DGP
tts = (1:n)*(1/n);
SFE = kron(ones(t,1),tts');
ttt = (1:t)*(1/t);
TFE = kron(ttt',ones(n,1));
Wmatrices = [kron(eye(t),W1) kron(eye(t),W2)];

% the DGP x-variables must match the order in which
% the function sdm_conv_panel_g()
% processes the x, W*x variables
Wx = x;
begi = 1;
endi = n*t;
for ii=1:m
        Wx = [Wx Wmatrices(:,begi:endi)*x];
    begi = begi + n*t;
    endi = endi + n*t;
end

Wxb = Wx*bvec;

sige = 1;
evec = sqrt(sige)*randn(n*t,1);

y = (speye(n*t) - rho*Wc)\(Wxb + SFE + TFE + evec);

% calculate true direct and indirect effects estimates
Wc_small = gamma1*W1 + gamma2*W2;

direct_true = zeros(k,1);
```

```
indirect_true = zeros(k,1);
total_true = zeros(k,1);

B = (speye(n) - rho*Wc_small)\(speye(n));

for ii=1:k
tmp2 = B*(eye(n)*beta(ii,1) + W1*theta1(ii,1) + W2*theta2(ii,1));
total_true(ii,1) = mean(sum(tmp2,2));
tmp1 = B*(eye(n)*beta(ii,1) + W1*theta1(ii,1) + W2*theta2(ii,1));
direct_true(ii,1) = mean(diag(tmp1));
indirect_true(ii,1) = total_true(ii,1) - direct_true(ii,1);
end

fprintf(1,'true effects estimates \n');
in.cnames = strvcat('direct','indirect','total');
in.rnames = strvcat('variables','x1','x2');

out = [direct_true indirect_true total_true];
mprint(out,in);

Wmatrices = [W1 W2];
ndraw = 40000;
nomit = 20000;
prior.model=3;
prior.plt_flag = 0;
prior.thin = 5;
result1 = sdm_conv_panel_g(y,x,Wmatrices,n,t,ndraw,nomit,prior);
vnames = strvcat('y','x1','x2');
prt_panel(result1,vnames);
```

Results from estimating the SDM version of the convex combination of weights model are shown below, where it took around 60 seconds to produce estimates based on 40,000 MCMC draws, with the first 20,000 omitted to allow for startup of the MCMC sampler, an a thinning of the draws that selects every 5th draw. The Geweke diagnostics are all greater than 0.992, and the mean and modal estimates are all close, suggesting a symmetric (posterior) distribution for all parameters. Use of the large $N \times T$ dimension $W-$matrices or small $N \times N$ dimension matrices should not impact the time required to produce estimates. Estimates from both calls to the estimation function should produce equivalent results.

An interesting point is that although the underlying parameter estimates for β, ρ, γ are very close to the true values used to generate the $y-$vector, we see much larger differences between the true and estimated direct, indirect and total effects estimates. For example an estimate of 0.4047 for the indirect effect of X_2 compared to the true value of 0.2771. This point is raised by LeSage and Pace (2018) in a critique of spatial econometric Monte Carlo studies. They point out that few Monte Carlo studies examine the bias and mean-squared error of the direct, indirect and total effects estimates, focusing instead on the underlying parameters β, ρ. Of course, ultimately estimates and inferences are based on the *scalar summary effects estimates*, not the parameters β, ρ.

They also show that estimates of dispersion for the effects estimates can vary widely across different estimation methods. Specifically, they show that a robust GMM estimation procedure proposed by Dogan and Taspinar (2014) produced bias in the effects estimates that were 1,000 times the bias in robust MCMC (where I have labeled the variance scalar parameters v_i). The standard deviations of the effects estimates from their robust GMM method were 10 times larger than those from MCMC used here. These dramatic differences in the effects estimates arose from cases where the differences in mean-squared-error (MSE) for the parameters β, ρ were relatively small. However, small differences in bias and dispersion of the underlying β, ρ parameters can translate into very large differences in the quality of the scalar summary effects estimates on which inferences are drawn regarding the impact of explanatory variables on the outcome variable y.

```
% results from sdm_conv_panel_gd.m file
true effects estimates
variables     direct   indirect      total
x1            1.0961     1.4039     2.5000
x2            0.5562     0.2771     0.8333

MCMC SDM convex combination W model with both region and time period fixed effects
Homoscedastic model
Bayesian spatial Durbin convex W model
Dependent Variable  =              y
Log-marginal        = -17486.8471
Log-marginal MCerror=  0.010795
R-squared           =    0.8341
corr-squared        =    0.6127
mean of sige draws  =    0.8994
posterior mode sige =    0.8984
Nobs, Nvars         =  10000,     2
ndraws,nomit        =  40000, 20000
time for effects    =   30.6630
time for sampling   =   29.1090
time for Taylor     =    0.1066
thinning for draws  =         5
min and max rho     =   -1.0000,   1.0000
***************************************************************
      MCMC diagnostics ndraws = 4000
Variable         mode         mean     MC error        tau     Geweke
x1             0.9857       0.9856   0.00014770   1.013926   0.999348
x2             0.4851       0.4855   0.00013276   1.047386   0.999784
W1*x1         -0.7485      -0.7485   0.00029180   1.317256   0.997094
W1*x2         -0.7600      -0.7600   0.00020498   0.995223   0.999443
W2*x1          0.4901       0.4913   0.00049706   1.184051   0.999647
W2*x2          0.5292       0.5296   0.00043872   1.090958   0.999128
rho            0.7184       0.7178   0.00026000   2.397971   0.999103
gamma1         0.2045       0.2044   0.00017553   2.837472   0.992124
gamma2         0.7955       0.7956   0.00017553   2.837472   0.997982

***************************************************************
      Posterior Estimates
Variable      Coefficient  Asymptot t-stat    z-probability
x1               0.985643        94.627650         0.000000
x2               0.485476        47.382284         0.000000
W1*x1           -0.748531       -48.114874         0.000000
W1*x2           -0.759994       -54.090795         0.000000
W2*x1            0.491291        16.968631         0.000000
W2*x2            0.529624        19.842740         0.000000
rho              0.717816        73.194910         0.000000
gamma1           0.204426        27.137981         0.000000
gamma2           0.795574       105.613993         0.000000
```

```
Direct        Coefficient       t-stat       t-prob      lower 05     upper 95
x1               1.039120    87.008150     0.000000      1.016309     1.062748
x2               0.499498    42.085465     0.000000      0.475953     0.523414

Indirect      Coefficient       t-stat       t-prob      lower 05     upper 95
x1               1.543781    13.276935     0.000000      1.328115     1.780967
x2               0.404725     3.863680     0.000112      0.191549     0.605726

Total         Coefficient       t-stat       t-prob      lower 05     upper 95
x1               2.582902    21.014500     0.000000      2.353339     2.834547
x2               0.904223     8.078247     0.000000      0.677120     1.120150
```

If you are interested in things other than those printed by the printing function *prt_panel()*, these are contained in the structure variable returned by the estimation functions. Typing simply *result1* in the MATLAB command window produces a listing of the *field* entries in the structure variable named *result1*. The function also returns information regarding the input options selected by the user, which are useful to the *prt_panel()* function when printing results.

4.2 The SDEM convex combination of W model

The SDEM convex combination model is shown in (4.8), where each $NT \times NT$ matrix W_m represents some sort of connectivity between regions, and takes a block diagonal form: $I_T \otimes w_m$, where w_m is an $N \times N$ weight matrix, with main diagonal elements equal to zero and row-sums equal to one.

$$\begin{aligned} y &= X\beta + \sum_{m=1}^{M} W_m X \theta_m + u, \\ u &= \rho(I_T \otimes W_c)u + \varepsilon, \quad \varepsilon \sim N(0, \sigma^2 I_n), \\ W_c(\Gamma) &= \sum_{m=1}^{M} \gamma_m W_m, \quad 0 \leq \gamma_m \leq 1, \ \sum_{m=1}^{M} \gamma_m = 1, \\ \Gamma &= (\gamma_1, \ldots, \gamma_M)' \end{aligned} \tag{4.8}$$

The $NT \times k$ matrix X in (4.8) contains exogenous explanatory variables, with β being the associated $k \times 1$ vector of parameters. As will be demonstrated, the function automatically generates and includes the explanatory variables $\sum_{m=1}^{M} W_m X$ and includes them in the model. As already noted, my thinking in taking this approach is that if you believe in multiple types of connectivity that involve interaction of the dependent variable y, then these different types of connectivity should also reflect *contextual effects* on which the outcomes in y depend. This model encounters the same issue as the SDM when it comes to use of the spatial lag of the explanatory variables, suggesting we *not* use $W_c X$, but rather each of the separate $W_m, m = 1, \ldots, M$ matrices to form the spatial lags of the explanatory variables.

The $NT \times 1$ vector u follows a spatial autoregressive process, and ε represents a constant variance normally distributed disturbance term, and the $NT \times 1$ dependent variable vector y takes the same form as in our models from Chapters 1 and 2. There is no option for use of the variance scalars v_{it} in this model, for reasons of computational speed.

Unlike the case of the SAR and SDM models, the (4.8) model does not have a computationally efficient representation, so this estimation procedure takes more time.

To estimate this model, we rely on the same Taylor series approximations used for the SAR and SDM convex combination model, so models with small N are likely to produce more inaccurate estimates.

The partial derivatives for our SDEM convex combination model take the form in (4.9).

$$\partial E(y)/\partial x_r \quad = \quad (I_{NT}\hat{\beta}_r + W_1\hat{\theta}_1 + \ldots + W_M\hat{\theta}_M) \tag{4.9}$$

This means that we can interpret the parameters $\hat{\beta}_r$ as the direct impacts of changes in the rth explanatory variable on the $y-$outcomes, and the sum of the coefficients $\hat{\theta}_m, m = 1, \ldots, M$ as representing the indirect or spillover effects. This model allows for *only* local spillover effects from immediate (first-order) neighbors. Least-squares estimates of this model should be *unbiased*, but not *efficient* because they ignore the spatial autoregressive dependence in the disturbances.

The next section illustrates use of the SDEM convex combination estimation functions.

4.2.1 Using the *sdem_conv_panel_FE_g()* function

The program below shows how to use the SDEM estimation function for the convex combination of weights model. Note that we provide only the matrix X since the function adds the spatial lags of the explanatory variables $(W_1X, W_2X, \ldots, W_MX)$. You can also call the function with small $N \times N$ matrices W_1, W_2, etc., or with large $NT \times NT$ matrices, with the large matrix input option allowing for different $W-$matrices for each time period. Of course, you should keep in mind the potential problem regarding the scalar summary effects estimates illustrated in Chapter 1.

```
% sdem_conv_panel_gd demo file
clear all;
rng(19203040);
[unclaims,b] = xlsread('../demo_data/weekly.xlsx',1);
% read data from sheet 1 of Excel spreadsheet
% growth rate of unemployment claims 2019-2020 from same week, previous year
snames = strvcat(b(2:end,1)); % 48 state names
tnames = strvcat(b(1,2:end)); % 51 week labels
[N,T] = size(unclaims);
[jobposts,b] = xlsread('../demo_data/weekly.xlsx',2);
% read data from sheet 2 of Excel spreadsheet
% change in job posts from 1st week of 2020
[athome,b] = xlsread('../demo_data/weekly.xlsx',3);
% read data from sheet 3 of Excel spreadsheet
% growth rate of percent population at home
% 2019-2020 from same week, previous year
[a,b] = xlsread('../demo_data/Wcont48.xlsx');
% 48 x 48 contiguity matrix for states
Wcontiguity = normw(a);
% state-to-state commodity flows, 2017
[a,b] = xlsread('../demo_data/cflows_2017.xlsx');
% set main diagonal (intrastate flows) to zero
diaga = diag(a);
W = a - diag(diaga);
Wcom_flows = normw(W); % row-normalize
```

```
% eliminate small elements
for i=1:N
    for j=1:N
        if Wcom_flows(i,j) < 0.005
            Wcom_flows(i,j) = 0;
        end
    end
end

Wcom_flows = normw(Wcom_flows);

y = vec(unclaims);
x = [vec(jobposts) vec(athome)];

vnames = strvcat('y=unclaims','jobposts','athome');

Wmatrices = [Wcontiguity Wcom_flows];

ndraw = 25000;
nomit = 5000;
prior.model = 3;
prior.thin = 4;
result1 = sdem_conv_panel_g(y,x,Wmatrices,N,T,ndraw,nomit,prior);
prt_panel(result1,vnames);
```

Results from estimating the SDEM version of the convex combination of weights model are shown below, where it took around 20 seconds to produce estimates based on 25,000 MCMC draws, with the first 5,000 omitted to allow for startup of the MCMC sampler, an a thinning of the draws that selects every 4th draw. The Geweke diagnostics for the $\gamma_2 = 0.1187$, is 0.956605, and the estimate is not significantly different from zero. The Geweke diagnostic likely points to a *pile-up* problem with the posterior distribution of this parameter, since this is near the lower bound of zero.

The resulting estimates point to the spatial contiguity matrix receiving all of the weight, with the weight matrix constructed from the state-to-state commodity flows receiving a convex combination weight γ_2 that is zero.

The estimation results indicate that the variable *jobposts* has a negative direct effect on continuing claims for unemployment insurance, as we would expect. The variable *athome* measuring the extent of social distancing has a positive direct impact on unemployment as we would expect.

```
% results from sdem_conv_panel_gd.m file
MCMC SDEM convex combination W model with both region and time period fixed effects
Homoscedastic model
Bayesian spatial Durbin error convex W model
Dependent Variable  =       y=unclaims
Log-marginal       = -5425.6646
Log-marginal MCerror=  0.041696
R-squared           =    0.9466
Rbar-squared        =    0.0405
mean of sige draws  =    0.0520
Nobs, Nvars         =   2448,     2
ndraws,nomit        =  25000,  5000
total time in secs =   22.8008
time for sampling  =   21.9250
```

```
time for Taylor    =    0.8758
min and max lambda =   -0.9999,   0.9999
****************************************************************
      MCMC diagnostics ndraws = 5000
Variable              mean     MC error        tau     Geweke
jobposts           -0.1736   0.00053841   0.987264   0.987508
athome              0.6059   0.00248615   1.026230   0.984898
W1*jobposts         0.0220   0.00111337   0.937262   0.967012
W1*athome          -1.0970   0.00383482   0.961495   0.998473
W2*jobposts        -0.2682   0.00217081   0.873444   0.975088
W2*athome           0.6225   0.00582558   1.019626   0.985784
rho                 0.2138   0.00058503   2.829731   0.990661
gamma1              0.8813   0.00202203   2.990870   0.994316
gamma2              0.1187   0.00202203   2.990870   0.956605

****************************************************************
      Posterior Estimates
Variable           Coefficient  Asymptot t-stat    z-probability
jobposts             -0.173583        -3.990345         0.000066
athome                0.605947         4.137389         0.000035
W1*jobposts           0.021958         0.221849         0.824432
W1*athome            -1.096961        -4.139570         0.000035
W2*jobposts          -0.268195        -1.340055         0.180228
W2*athome             0.622473         1.772541         0.076305
rho                   0.213788         6.855115         0.000000
gamma1                0.881256         9.041579         0.000000
gamma2                0.118744         1.218304         0.223109

Direct             Coefficient           t-stat           t-prob          lower 05          upper 95
jobposts             -0.173583        -3.990345         0.000068         -0.258555         -0.087987
athome                0.605947         4.137389         0.000036          0.314287          0.892576

Indirect           Coefficient           t-stat           t-prob          lower 05          upper 95
jobposts             -0.246237        -1.418609         0.156140         -0.589946          0.087814
athome               -0.474488        -1.414911         0.157222         -1.118117          0.186970

Total              Coefficient           t-stat           t-prob          lower 05          upper 95
jobposts             -0.419819        -2.328930         0.019944         -0.770450         -0.074316
athome                0.131460         0.432500         0.665416         -0.458798          0.728116
```

Another point to note regarding the printed output is that the indirect (spillover) effects are not significantly different from zero. This suggests that we might consider an SEM model that excludes the WX explanatory variables. We turn attention to this model in the next section.

4.3 The SEM convex combination of W model

The SEM convex combination model is shown in (4.10), where each $NT \times NT$ matrix W_m represents some sort of connectivity between regions, and takes a block diagonal form: $I_T \otimes w_m$, where w_m is an $N \times N$ weight matrix, with main diagonal elements equal to zero and row-sums equal to one.

$$y = X\beta + u,$$

$$
\begin{aligned}
u &= \rho(I_T \otimes W_c)u + \varepsilon, \quad \varepsilon \sim N(0, \sigma^2 I_n), \\
W_c(\Gamma) &= \sum_{m=1}^{M} \gamma_m W_m, \quad 0 \le \gamma_m \le 1, \ \sum_{m=1}^{M} \gamma_m = 1, \\
\Gamma &= (\gamma_1, \ldots, \gamma_M)'
\end{aligned} \tag{4.10}
$$

The $NT \times k$ matrix X in (4.10) contains exogenous explanatory variables, with β being the associated $k \times 1$ vector of parameters. This model is subsumed as a special case of the SDEM specification.

This model does not allow for any spatial spillovers, only direct effects. Recall, that LeSage and Pace (2014) argue that we should not label the spatial impacts arising from dependence in the disturbances as reflecting spatial spillovers. They reserve the term spatial spillovers to describe situations where changes in a variable x_j from region j impacts outcomes y_i in another region i. They label the situation where shocks to the disturbances u_j impact disturbances in other regions $u_i, i \neq j$ as *global spatial shocks* to the disturbances.

The partial derivatives for this model are identical to those from OLS, because $\partial E(y)/\partial X_r = \hat{\beta}_r$. The $t-$statistics provide a valid basis for inference regarding the significance of the explanatory variables. Theoretically, we know that OLS estimates for this type of model relationship would produce unbiased, but inefficient estimates. The inefficiency would arise from ignoring the spatial dependence in the disturbances of the model.

4.3.1 Using the *sem_conv_g()* function

The program *sem_conv_gd.m* file is shown below, where we estimate an SEM version of the pandemic model used to illustrate the SDEM model specification. This model will exclude the W_1X, W_2X variables that we associated with insignificant indirect effects estimates in the printed results for the SDEM specification.

```
% sem_conv_panel_g demo file
clear all;
rng(19203040);
[unclaims,b] = xlsread('../demo_data/weekly.xlsx',1);
% read data from sheet 1 of Excel spreadsheet
% growth rate of unemployment claims 2019-2020 from same week, previous year
snames = strvcat(b(2:end,1)); % 48 state names
tnames = strvcat(b(1,2:end)); % 51 week labels
[N,T] = size(unclaims);
[jobposts,b] = xlsread('../demo_data/weekly.xlsx',2);
% read data from sheet 2 of Excel spreadsheet
% change in job posts from 1st week of 2020
[athome,b] = xlsread('../demo_data/weekly.xlsx',3);
% read data from sheet 3 of Excel spreadsheet
% growth rate of percent population at home
% 2019-2020 from same week, previous year
[a,b] = xlsread('../demo_data/Wcont48.xlsx');
% 48 x 48 contiguity matrix for states
Wcontiguity = normw(a);
% state-to-state commodity flows, 2017
[a,b] = xlsread('../demo_data/cflows_2017.xlsx');
% set main diagonal (intrastate flows) to zero
```

```
diaga = diag(a);
W = a - diag(diaga);
Wcom_flows = normw(W); % row-normalize
% eliminate small elements
for i=1:N
    for j=1:N
        if Wcom_flows(i,j) < 0.005
            Wcom_flows(i,j) = 0;
        end
    end
end

Wcom_flows = normw(Wcom_flows);

y = vec(unclaims);
x = [vec(jobposts) vec(athome)];

vnames = strvcat('y=unclaims','jobposts','athome');
Wmatrices = [Wcontiguity Wcom_flows];

ndraw = 25000;
nomit = 5000;
prior.model = 3;
prior.thin = 4;
result1 = sem_conv_panel_g(y,x,Wmatrices,N,T,ndraw,nomit,prior);
prt_panel(result1,vnames);

result2 = sdem_conv_panel_g(y,x,Wmatrices,N,T,ndraw,nomit,prior);
prt_panel(result2,vnames);
```

The SEM estimation results for the coefficient $\hat{\rho} = 0.2043$ is similar to that from the SDEM, where $\hat{\rho} = 0.2135$, and the same is true for the coefficient on *jobposts*, which was -0.1726 for the SEM and -0.1730 for the SDEM. A difference arises for the coefficient on *athome*, which was 0.3498 for SEM and 0.6067 for SDEM.

```
MCMC SEM convex combination W model with both region and time period fixed effects
Homoscedastic model
Bayesian spatial error convex W model
Dependent Variable  =       y=unclaims
Log-marginal likeli = -5427.9235
Log-marginal MCerror=  0.039053
R-squared           =    0.9460
Rbar-squared        =    0.0350
mean of sige draws  =    0.0524
Nobs, Nvars         =   2448,     2
ndraws,nomit        =  25000,  5000
total time in secs =   11.7063
time for sampling  =   11.0780
time for Taylor    =    0.6283
min and max lambda =   -0.9999,   0.9999
***************************************************************
      MCMC diagnostics ndraws = 5000
Variable           mean     MC error        tau     Geweke
jobposts        -0.1726   0.00048286   0.952757   0.988203
athome           0.3498   0.00201254   0.960440   0.982154
```

```
rho             0.2044   0.00049702   1.737929   0.978434
gamma1          0.8894   0.00229116   2.632371   0.977323
gamma2          0.1106   0.00229116   2.632371   0.822811

***************************************************************
      Posterior Estimates
Variable        Coefficient  Asymptot t-stat    z-probability
jobposts          -0.172630        -4.024668         0.000057
athome             0.349827         2.779049         0.005452
rho                0.204389         6.871731         0.000000
gamma1             0.889371         9.637786         0.000000
gamma2             0.110629         1.198845         0.230588

MCMC SDEM convex combination W model with both region and time period fixed effects
Homoscedastic model

Bayesian spatial Durbin error convex W model
Dependent Variable  =       y=unclaims
Log-marginal      = -5425.6480
Log-marginal MCerror=  0.039420
R-squared          =    0.9466
Rbar-squared       =    0.0417
mean of sige draws  =    0.0520
Nobs, Nvars        =   2448,     2
ndraws,nomit       =  25000,  5000
total time in secs =   23.9815
time for sampling  =   23.3500
time for Taylor    =    0.6315
min and max lambda =   -0.9999,   0.9999
***************************************************************
      MCMC diagnostics ndraws = 5000
Variable              mean     MC error        tau     Geweke
jobposts           -0.1731   0.00055042   0.952716   0.997698
athome              0.6067   0.00204654   0.979529   0.992644
W1*jobposts         0.0224   0.00118543   0.994505   0.857231
W1*athome          -1.0943   0.00374800   1.118881   0.996953
W2*jobposts        -0.2625   0.00230207   0.991327   0.994754
W2*athome           0.6202   0.00511606   0.987403   0.948309
rho                 0.2124   0.00064937   1.896266   0.995576
gamma1              0.8864   0.00195475   2.997233   0.996274
gamma2              0.1136   0.00195475   2.997233   0.969935

***************************************************************
      Posterior Estimates
Variable           Coefficient  Asymptot t-stat    z-probability
jobposts             -0.173081        -4.041734         0.000053
athome                0.606741         4.121820         0.000038
W1*jobposts           0.022436         0.224902         0.822056
W1*athome            -1.094320        -4.184257         0.000029
W2*jobposts          -0.262455        -1.303718         0.192330
W2*athome             0.620160         1.743555         0.081237
rho                   0.212350         6.785341         0.000000
gamma1                0.886418         9.636189         0.000000
gamma2                0.113582         1.234739         0.216928
```

```
Direct            Coefficient          t-stat          t-prob        lower 05        upper 95
jobposts            -0.173081       -4.041734        0.000055       -0.257179       -0.088067
athome               0.606741        4.121820        0.000039        0.316506        0.899569

Indirect          Coefficient          t-stat          t-prob        lower 05        upper 95
jobposts            -0.240019       -1.392115        0.164014       -0.579900        0.100158
athome              -0.474160       -1.410041        0.158655       -1.146754        0.190287

Total             Coefficient          t-stat          t-prob        lower 05        upper 95
jobposts            -0.413100       -2.307506        0.021110       -0.765433       -0.063001
athome               0.132581        0.432153        0.665668       -0.464572        0.743976
```

We can compare these estimates to see if they are significantly different, using the MCMC draws. The code snippet below (taken from the file: sem_conv_panel_gd2.m) reads the MCMC draws for the coefficient on the *athome* variable from the SEM model (from result1.bdraw) and from the SDEM model (the result2.bdraw), and produces posterior density plots as well as a posterior density for the difference between the two coefficients.

```
beta1 = result1.bdraw(:,2);
beta2 = result2.bdraw(:,2);

beta_diff = beta2 - beta1;

[h1,f1,y1] = pltdens(beta1);
[h2,f2,y2] = pltdens(beta2);
[h3,f3,y3] = pltdens(beta_diff);

subplot(2,1,1),
plot(y1,f1,'.-r',y2,f2,'.-b');
ylabel('\beta posteriors');
xlabel('\beta values');
legend('\beta_1','\beta_2');
subplot(2,1,2),
plot(y3,f3,'.-g');
ylabel('Posterior for \beta_2 - \beta_1');
xlabel('\beta_2 - \beta_1 values');
zipi = find(y3 > 0);
line([0 0],[0 f3(zipi(1,1))]);
legend('\beta_2 - \beta_1','zero');

% trapezoid rule integration
sum_all = trapz(y3,f3);
sum_positive = trapz(y3(zipi,1),f3(zipi,1));
prob = sum_positive/sum_all
% prob = 0.8966
```

We use trapezoid-rule integration to calculate the mass of the distribution that is positive, and as a test for a significant difference we would use: $1 - prob = 1 - 0.8966 = 0.1034$, which would not allow us to conclude that the coefficients from the SDEM and SEM specifications are significantly different at the 90% level.

4.4 Chapter summary

We have presented a series of convex combination of weight matrices models that extend the flexibility of the spatial regression models to allow for more than simply spatial connectivity of dependence between observations. We can for example construct weight matrices that reflect dependence based on commodity or migration flows between regions which are very different from spatial connectivity.

Once we open the door to use of multiple weight matrices, questions regarding the number of matrices and which specific matrices to incorporate in our models arise. The next chapter tackles comparison of models based on differing number of weight matrices.

Unlike the case in ordinary least-squares regression where addition of irrelevant explanatory variables produces no bias in estimates associated with relevant explanatory variables, the convex combination of weights spatial regression models can suffer from the introduction of irrelevant weight matrices. To see this, consider that irrelevant weight matrices will lead to posterior estimates for the convex combination parameters that we have labeled $\gamma_m, m = 1, \ldots, M$ that are non-zero, having distributions that pile-up at the lower and upper bounds of the 0,1 parameter space for these parameters. A large number of such irrelevant weight matrices will then subtract weight from the γ parameters associated with the relevant weights.

4.5 Chapter references

Dogan, O., Taspinar, S. (2014) Spatial autoregressive models with unknown heteroskedasticity: A comparison of Bayesian and robust GMM approach. *Regional Science and Urban Economics*, 45, 1-21.

LeSage, J. P. and R. K. Pace (2014). Interpreting Spatial Econometric Models, *Handbook of Regional Science*, M. M. Fischer and Peter Nijkamp (Eds.) Springer, Berlin 2014, pp. 1535-1552.

LeSage, J. P. and R. K. Pace (2018), Spatial econometric Monte Carlo studies: raising the bar, *Empirical Economics*, 55:1, 17-24.

Chapter 5

Model comparison

This chapter discusses model comparison for conventional static spatial panel data models with the topic of model comparison for convex combination of weight matrices models taken up in the next chapter.

For static spatial panel data models, LeSage (2014) provides a Bayesian approach to calculating the log-marginal likelihood using univariate numerical integration over the parameter ρ. This is done after analytical integration of the parameters β, σ^2 leaving us with a simple univariate integration problem over the remaining model parameter ρ. Bayesian model comparison relies on the log-marginal likelihood which requires integration over all model parameters (and prior distributions where appropriate).

An advantage of this approach to model comparison relative to approaches based on likelihood-ratio or lagrange multiplier methods is that since we have integrated over the model parameters, statements made about one model versus another *do not* depend on specific parameter values, but are valid for *all* possible parameter values. We can say that (for example), the SDM model specification is the one most consistent with the sample data set, y, X, W, irrespective of the model parameters. This is quite different from the case of likelihood ratio or lagrange multiplier tests, frequently used for model comparison. These methods involve evaluating likelihoods for *specific* values of the parameters. When comparing two models, if one is the correct specification, with associated (correct) parameter estimates, then the other model must be an incorrect specification, with associated (incorrect) parameter estimates. This is not the case for Bayesian model comparison methods.

The Bayesian solution to model choice is to calculate posterior model probabilities associated with each model. These make choice of the model most consistent with the sample data easy, it is the specification with the highest model probability. Individual model probabilities can also be used to weigh the parameter estimates from each model specification to produce a *model averaged* set of estimates. The model averaged estimates formally incorporate model uncertainty about the correct model specification into the statistical inference problem at hand.

LeSage (2014) advances a formal argument that logically implies we can simplify the task of choosing between SLX, SAR, SDM, SEM, SDEM specifications by focusing on only three model specifications. Specifically, the SLX, SDM and SDEM specifications. Without going into the details of his argument, it should be intuitively clear that the SDM specification encompasses the SAR model as a special case, and the SDEM specification subsumes the SEM specification as a special case. This rules out the need to consider the SAR and SEM models from our comparison scheme.

The SLX model results when we have an SDM specification with $\rho = 0$, so there is no spatial dependence between observations in the $y-$vector. The SDEM model specification collapses to the SLX when $\rho = 0$, so there is no dependence in the disturbances.

Our primary focus in on comparing two models, the SDM versus SDEM for the case of static spatial panel models. We can also use the methods set forth here to compare the *same* models (e.g., two SDM models) based on alternative spatial weight matrices, something that is often of interest in applied practice. For the case of two SDM models based on different spatial weight matrices, M_1 and M_2 with parameter vectors δ_1, δ_2 and data (y, X, W_1, W_2) denoted by simply y, we can use Bayes' theorem to calculate the posterior probability that M_1 is the correct model (conditional on the fact that the correct model is in the set M_1, M_2). This is given by:

$$p(M_1|y) = \frac{p(y|M_1)}{p(y|M_1) + p(y|M_2)} \times \frac{p(M_1)}{p(M_2)} \tag{5.1}$$

where $p(y|M_k)$ is the marginal likelihood of the data given M_k and $p(M_k)$ is the prior probability of the model $M_k (k = 1, 2)$. With only two SDM models, $p(M_1|y) + p(M_2|y) = 1$. The difference between prior probabilities assigned to the models and the posterior model probabilities reflects *Bayesian learning* about the model specification conditional only on the sample data. The toolbox function *lmarginal_static_panel()* described here relies on assigning no prior distributions for the model parameters, and no prior model probabilities, so the prior probabilities could be viewed as all equal, that is $p(M_1) = p(M_2)$, which means the numerator and denominator cancel in the expression (5.1) for the model probability calculation.

The marginal likelihood for model M_1 is obtained by integrating over the parameters, with (5.2) showing the integral expression for the case of model 1 and parameters δ_1.

$$\begin{aligned} p(y|M_1) &= \int p(y|\delta_1, M_1) p(\delta_1|M_1) d\delta_1 \\ &= \int (\text{likelihood x prior})\ d\delta_1 \end{aligned} \tag{5.2}$$

As parameters governing the prior distributions such as the prior variance increase, the prior distributions become more vague or diffuse and often tend to uniform distributions (Zellner, 1971), which provide the basis for many of the Bayesian econometric results found in (Zellner, 1971). However, uniform distributions are improper since they can reflect a situation where no defined limits exists for the integral of the prior distribution, leaving these undefined.

Impropriety can be a problem in some model comparison contexts and not in others. For the case of comparing weight matrices as well as models such as the the SDM versus SDEM specifications, we manage to avoid these problems. As already noted, model choice depends on ratios between model posterior probabilities. If the improper priors for the two models are different, they will not cancel each other properly. This lack of proper cancelation will also occur if the two models being compared have different numbers of parameters with improper priors. This produces a situation where the ratio of model probabilities becomes zero or infinite depending on the specific model in the numerator versus denominator (Koop. 2003, 40-42). Problems can also arise in situations where the number of parameters with improper priors is the same but explanatory variables in X are scaled differently in the two models (Koop, 2003, 42).

Since we may wish to rely on improper priors for some model parameters, we need to avoid situations where these fail to cancel during model comparison. Koop (2003, p. 42) provides guidance for us:

> When comparing models using posterior odds ratios, it is acceptable to use noninformative priors over parameters which are common to all the models. However, informative, proper priors should be used over all other parameters. (Koop, 2003, p. 42)

Since the explanatory variables matrices are the same in the SDM and SDEM panel models, or in two (say) SDM models with different weight matrices, we can rely on noninformative priors for these parameters. The single dependence parameter can be viewed as playing a different role in the two models, so we assign a proper prior for the dependence parameter to we meet these conditions. A series of Monte Carlo experiments carried out by LeSage (2014) demonstrate that the log marginal likelihoods for the SDM and SDEM models are "well-behaved" with a uniform but proper prior for ρ, e.g., $p(\rho) = 1/D$, $D = 1/\omega_{\max} - 1/\omega_{\min}$, where $\omega_{\min}, \omega_{\max}$ are minimum and maximum eigenvalues of the spatial weight matrix W. (For row normalized W, $\omega_{\max} = 1$). This type of prior requires no subjective information on the part of the practitioner as it relies on the defined dependence parameter space for these models.

A single function calculates and returns the log-marginal likelihoods and associated model probabilities for the SLX, SDM and SDEM model specifications. Use of this is illustrated in the next section.

5.1 Using the *lmarginal_static_panel()* function

We do not need to carry out estimation of the model parameters to find the log-marginal likelihood because the process of calculating this quantity involves integration over all model parameters. The function *lmarginal_static_panel()* carries out the required integration.

The program below illustrates use of the function with the pandemic panel dataset for 51 weeks during 2020 and 48 US states. An important point is that the model data needs to be transformed to take into account fixed effects (if these are desired). Note that the estimation functions discussed in Chapters 1 to Chapter 4 allow choice of the input field info.model=0,1,2,3, where 0 indicates no fixed effects, 1 region-specific effects, 2 time-specific effects and 3 is both region and time effects. The estimation functions carry out these transformation for you prior to estimation of the model parameters.

In this example, we have an untransformed sample data vector y for the dependent variable reflecting weekly 2020 continuing claims for unemployment insurance, expressed as annual growth rates relative to the same week during 2019. Our explanatory variables are also expressed as untransformed with regard to region- or time-fixed effects. The function *demean()* is used to carry out this transformation, and the demeaned variable vectors are input to the function *lmarginal_static_panel()*.

We calculate log-marginal likelihoods for: 1) a model based on no fixed effects, 2) a model based on only region-specific effects, and 3) a model based on region- and time-specific effects.

The function returns a structure variable that contains fields: *result1.lmarginal, result1.probs*, which we collect and print out with row- and column-labels, using the function *mprint()* from my toolbox.

```
% model_comparison_chapter5p1.m demo file
clear all;
[cclaims,b] = xlsread('../demo_data/weekly.xlsx',1);
% read data from sheet 1 of Excel spreadsheet
% growth rate of unemployment 2019-2020 from same week, previous year
snames = strvcat(b(2:end,1)); % 48 state names
tnames = strvcat(b(1,2:end)); % 51 week labels
[N,T] = size(cclaims);
[jobposts,b] = xlsread('../demo_data/weekly.xlsx',2);
% read data from sheet 2 of Excel spreadsheet
% change in job offers from 1st week of 2020
[athome,b] = xlsread('../demo_data/weekly.xlsx',3);
% read data from sheet 3 of Excel spreadsheet
% growth rate of percent population at home
% 2019-2020 from same week, previous year
[a,b] = xlsread('../demo_data/Wcont48.xlsx');
% 48 x 48 contiguity matrix for states
W = normw(a);

y = vec(cclaims);
x = [vec(jobposts) vec(athome)];

model = 0; % no fixed effects
[ywith,xwith,meanny,meannx,meanty,meantx]=demean(y,x,N,T,model);

info.lflag = 0; % exact log-determinant
% info.lflag = 1; uses Pace and Barry approximation
% which is faster for large data samples

result1 = lmarginal_static_panel(ywith,xwith,W,N,T,info);

fprintf(1,'no fixed effects: log marginal likelihoods and model probabilities \n');
in.cnames = strvcat('log-marginal','model probs');
in.rnames = strvcat('model','slx','sdm','sdem');
in.width = 10000;
in.fmt = '%10.4f';
out1 = [result1.lmarginal result1.probs];
mprint(out1,in);

model = 1; % state-specific fixed effects
[ywith,xwith,meanny,meannx,meanty,meantx]=demean(y,x,N,T,model);

result2 = lmarginal_static_panel(ywith,xwith,W,N,T,info);

fprintf(1,'region fixed effects: log marginal likelihoods and model probabilities \n');
in.cnames = strvcat('log-marginal','model probs');
in.rnames = strvcat('model','slx','sdm','sdem');
in.width = 10000;
in.fmt = '%10.4f';
out2 = [result2.lmarginal result2.probs];
mprint(out2,in);

model = 3; % state- and time-specific effects
[ywith,xwith,meanny,meannx,meanty,meantx]=demean(y,x,N,T,model);
```

```
result3 = lmarginal_static_panel(ywith,xwith,W,N,T,info);

fprintf(1,'region and time fixed effects: log marginal likelihoods and model probabilities \n');
in.cnames = strvcat('log-marginal','model probs');
in.rnames = strvcat('model','slx','sdm','sdem');
in.width = 10000;
in.fmt = '%10.4f';
out3 = [result3.lmarginal result3.probs];
mprint(out3,in);
```

The printed results are shown below, where we see that changes in use of fixed effects for states and time periods result in changes to the posterior model probabilities.

```
results from running: model_comparison_chapter5p1.m
no fixed effects: log marginal likelihoods and model probabilities
model log-marginal  model probs
slx     -3865.3240       0.0000
sdm     -2221.5441       1.0000
sdem    -2238.3516       0.0000

region fixed effects: log marginal likelihoods and model probabilities
model log-marginal  model probs
slx     -3673.6963       0.0000
sdm     -1331.8599       1.0000
sdem    -1346.1736       0.0000

region and time fixed effects: log marginal likelihoods and model probabilities
model log-marginal  model probs
slx      -747.3945       0.0000
sdm      -723.0018       0.4800
sdem     -722.9218       0.5200
```

There are some input options that can be specified, which can be viewed in the function documentation, printed out with *help lmarginal_static_panel* typed in the MATLAB command window. One important option is whether to use a Monte Carlo approximation to calculate the log-determinant term that appears in the expressions over which we need to carry out univariate integration, *info.lflag = 0,1* with 0 indicating the exact log-determinant and 1 the Barry and Pace (1999) and Pace and Barry (1997) approximation. If you select the approximation, there are some options, named *info.order* and *info.iter* that control the accuracy of the approximation. These work in the same fashion as the equivalent options in the spatial econometrics toolbox estimation functions that work on cross-sectional estimation for SAR, SDM, SEM, SDEM models. There is another option for setting the limits used for integrating over the parameter ρ. All of these options are set by default to operate at maximum speed. For small problems there is little difference in speed, so use of the *info.lflag=0* would be more accurate. It is also the case that running the program more than once with the approximation to the log-determinant will produce slightly different results because the approximation is based on a statistical estimate of the log-determinant that will change on each run of the program (unless you fix the random number generator seed). Setting the 'seed' should produce identical results for every run of the program.

```
 -------------------------------------------------------------
 USAGE: results = lmarginal_static_panel(y,x,W,N,T,info)
```

```
  where: y = dependent variable vector (N*T x 1)
         x = independent variables matrix, WITHOUT INTERCEPT TERM
         W = N by N spatial weight matrix (for W*y and W*e)
         N = # of cross-sectional units
         T = # of time periods
        info.lflag = 0 for full lndet computation (default = 1, fastest)
                   = 1 for MC lndet approximation (fast for very large problems)
        info.order = order to use with info.lflag = 1 option (default = 50)
        info.iter  = iterations to use with info.lflag = 1 option (default = 30)
        info.rmin  = (optional) minimum value of rho to use in search (default = -1)
        info.rmax  = (optional) maximum value of rho to use in search (default = +1)
        info.iflag = 0 for conventional W-matrix
        info.iflag = 1 for transformed W-matirx (using dmeanF())
 -------------------------------------------------------------
```

Since we have a small $N = 48$ problem, we should use the exact log-determinant term. Results from doing this are shown below, where we see a slight change in the results for the model probabilities for the case of both region- and time-specific effects estimates, which are still close to 50% for the SDM and SDEM specifications.

```
>> results from model_comparison_chapter5p2.m file
no fixed effects: log marginal likelihoods and model probabilities
model log-marginal  model probs
slx     -3865.3240       0.0000
sdm     -2261.1670       1.0000
sdem    -2279.2742       0.0000

region fixed effects: log marginal likelihoods and model probabilities
model log-marginal  model probs
slx     -3673.6963       0.0000
sdm     -1331.1294       1.0000
sdem    -1391.4137       0.0000

region and time fixed effects: log marginal likelihoods and model probabilities
model log-marginal  model probs
slx      -747.3945       0.0000
sdm      -724.4677       0.4786
sdem     -724.3822       0.5214
```

A point to note is that the ability of this Bayesian approach to model comparison to produce accurate results depends on things like sample size N, the magnitude of spatial dependence ρ, and the signal/noise ratio of the model relationship. LeSage (2014, 2015) demonstrates these issues.

It should be clear that as the level of spatial dependence approaches zero, the SDM and SDEM specifications collapse to the SLX. This means that a set of posterior model probabilities that suggest support for all three models should be interpreted as evidence in favor of the SLX model. In other words, we would not necessarily expect to see a probability of one for the SLX specification, even if it were the true specification. We demonstrate this type of result later, where we see that the SLX model receives a probability around 0.5 and the SDM and SDEM models probabilities around 0.25 each.

Regarding the pandemic model relationship and data, it should be noted that most of the variation in the data is over the 51 weekly time periods, with far less variation over the 48 states. Introducing fixed effects for the time periods (as was done in the third call to the *lmarginal_static_panel()*

function) is likely to make it more difficult to distinguish between the SDM and SDEM model specifications if we have a low level of spatial dependence. From the estimation results for the SDEM variant of the pandemic panel data model (see Chapter 4), we know that ρ is around 0.20.

I would interpret the results above to mean that if we eliminate time-specific variation in our model relationship, then it is difficult to make a decision regarding which model specification is preferred, SDM or SDEM. On the other hand, if we rely only on fixed effects for the 48 states, the SDM is the clear model choice.

Estimates from both of these models were produced using the code snippet below where both fixed effects for both states and time periods are used.

```
% code snippet from model_comparison_chapter5p1.m file
% ============================================
% estimate SDM and SDEM models
ndraw = 6000;
nomit = 1000;
prior.model = 3;
prior.novi_flag = 1;
result1 = sdm_panel_FE_g(y,x,W,T,ndraw,nomit,prior);
vnames = strvcat('cclaims','jobposts','athome');
prt_panel(result1,vnames);

result2 = sdem_panel_FE_g(y,x,W,T,ndraw,nomit,prior);
vnames = strvcat('unemp','jobposts','athome');
prt_panel(result2,vnames);
==========================    comparative estimation results =====================
            SDEM model estimates                  SDM model estimates
Variable     Coefficient  t-stat      t-prob      Coefficient  t-stat      t-prob
jobposts     -0.177856  -4.142717   0.000034     -0.173482    -4.126911   0.000037
athome        0.636863   4.385501   0.000012      0.685230     4.392607   0.000011
W-jobposts   -0.038726  -0.445963   0.655624      0.009221     0.111971   0.910846
W-athome     -0.879429  -3.729642   0.000192     -0.938773    -4.020709   0.000058
rho           0.180731   6.897159   0.000000      0.180712     6.984416   0.000000

Direct      Coefficient  t-stat      t-prob      Coefficient  t-stat       t-prob
jobposts     -0.177856  -4.142717   0.000035     -0.174523    -4.103252   0.000042
athome        0.636863   4.385501   0.000012      0.648554     4.309188   0.000017

Indirect    Coefficient  t-stat      t-prob       Coefficient  t-stat      t-prob
jobposts      -0.038726  -0.445963  0.655664     -0.026245    -0.268753   0.788143
athome        -0.879429  -3.729642  0.000196     -0.958386    -3.772160   0.000166

Total       Coefficient  t-stat      t-prob       Coefficient  t-stat      t-prob
jobposts     -0.216582  -2.115702   0.034471     -0.200768    -1.776133   0.075835
athome       -0.242566  -1.187884   0.234995     -0.309832    -1.447056   0.148009
```

The estimation results are shown above, where we see very similar estimates from both model specifications. This of course if consistent with the posterior model probabilities suggesting both models are equally consistent with the sample data. These results are consistent with what LeSage and Pace (2014) call the "biggest myth in spatial econometrics". The myth is that practitioners believe estimates and inferences from spatial regression models as highly sensitive to slight differences in weight matrices used to produce estimates. This is far from true, since differences in a handful of the n^2 elements of the matrix W are likely to result in matrix-vector products, Wy, Wx, that are highly correlated. The myth arose because people focus on estimates for the parameters β, ρ not on the proper estimates reflected by the scalar summary estimates for direct, indirect and total effects. The latter estimates reflect the true partial derivatives of the model, which cannot be said of estimates for β, ρ.

5.2 Comparing weight matrices for static panel data models

A first illustration uses the pandemic dataset and model, where we compare a weight matrix W_c based on first-order contiguity of the 48 states, where equal weights are assigned to all neighboring states, those with borders touching. A second weight matrix W_b applies weights to bordering states based on the miles of border in common with neighboring states. The only difference between the two weight matrices would be that unequal weights are assigned by the matrix W_b, and equal weights assigned in the case of W_c.

I assume that we have not decided on the appropriate model specification, so we produce log-marginal likelihoods for a set of 6 models, 3 based on the matrix W_c (for SLX, SDM, SDEM) and 3 more for the model based on W_b.

The program calls the function *lmarginal_static_panel()* using the small $n \times n$ weight matrices, as this function will *not work* for the case of a large $nt \times nt$ weight matrix.

The information in the file *states_borders.xlsx* is for miles of common borders and is upper-triangular, so we make it triangular and then row-normalize it using the *normw()* function from my toolbox.

```
% model_comparison_chapter5p2.m demo file
clear all;
[cclaims,b] = xlsread('../demo_data/weekly.xlsx',1);
% read data from sheet 1 of Excel spreadsheet
% growth rate of unemployment 2019-2020 from same week, previous year
snames = strvcat(b(2:end,1)); % 48 state names
tnames = strvcat(b(1,2:end)); % 51 week labels
[N,T] = size(cclaims);
[jobposts,b] = xlsread('../demo_data/weekly.xlsx',2);
% read data from sheet 2 of Excel spreadsheet
% change in job offers from 1st week of 2020
[athome,b] = xlsread('../demo_data/weekly.xlsx',3);
% read data from sheet 3 of Excel spreadsheet
% growth rate of percent population at home
% 2019-2020 from same week, previous year
[a,b] = xlsread('../demo_data/Wcont48.xlsx');
% 48 x 48 contiguity matrix for states
Wc = normw(a);
% miles of borders in common
[a,b] = xlsread('../demo_data/states_borders.xlsx');
Wmiles = a(:,2:end);
% only upper triangular
% so we make it symmetric
for i=1:48
    for j=1:48
        if Wmiles(i,j) > 0
            Wmiles(j,i) = Wmiles(i,j);
        end
```

```
    end
end
Wb = normw(Wmiles);

y = vec(cclaims);
x = [vec(jobposts) vec(athome)];

model = 3; % both state and time fixed effects
[ywith,xwith,meanny,meannx,meanty,meantx]=demean(y,x,N,T,model);

info.lflag = 0; % exact log-determinant
result1c = lmarginal_static_panel(ywith,xwith,Wc,N,T,info);
result1b = lmarginal_static_panel(ywith,xwith,Wb,N,T,info);

fprintf(1,'state and time fixed effects: log marginal likelihoods and model probabilities \n');
in.cnames = strvcat('Wc log-marginal','model probs');
in.rnames = strvcat('model','slx','sdm','sdem');
in.width = 10000;
in.fmt = '%10.4f';
out1 = [result1c.lmarginal result1c.probs];
mprint(out1,in);

fprintf(1,'state and time fixed effects: log marginal likelihoods and model probabilities \n');
in.cnames = strvcat('Wb log-marginal','model probs');
in.rnames = strvcat('model','slx','sdm','sdem');
in.width = 10000;
in.fmt = '%10.4f';
out2 = [result1b.lmarginal result1b.probs];
mprint(out2,in);

% compare all models
lmarginals = [result1c.lmarginal
              result1b.lmarginal];

probs = model_probs(lmarginals);
% rearrange for pretty printing
mprobs = reshape(probs,3,2);

out = [result1c.lmarginal mprobs(:,1) result1b.lmarginal mprobs(:,2)];

fprintf(1,'Comparison of Wc and Wb \n');
in.cnames = strvcat('Wc log-marginal','Wc model probs','Wb log-marginal','Wb model probs');
in.rnames = strvcat('model','slx','sdm','sdem');
in.width = 10000;
in.fmt = '%10.4f';
mprint(out,in);
```

The program produces 3 log-marginal likelihoods for the SLX, SDM and SDEM based on the matrix Wc first, then another 3 log-marginal likelihoods for the SLX, SDM and SDEM based on weight matrix Wb. An important point is that we can place these 6 log-marginal likelihoods in a vector and call the function *model_probs()* to calculate probabilities for all 6 models.

```
% results returned by: running model_comparison_chapter5p2.m
state and time fixed effects: log marginal likelihoods and model probabilities
model Wc log-marginal     model probs
```

```
slx         -747.3945          0.0000
sdm         -723.0018          0.4800
sdem        -722.9218          0.5200

state and time fixed effects: log marginal likelihoods and model probabilities
model Wb log-marginal     model probs
slx         -742.2608          0.0000
sdm         -714.8122          0.6329
sdem        -715.3568          0.3671

Comparison of Wc and Wb
model Wc log-marginal  Wc model probs Wb log-marginal  Wb model probs
slx         -747.3945          0.0000       -742.2608          0.0000
sdm         -723.0018          0.0002       -714.8122          0.6326
sdem        -722.9218          0.0002       -715.3568          0.3670
```

The results point to an SDM specification based on the border miles in common weight matrix having a probability of 0.6326. There is very little support for the weight matrix based on equally weighted contiguous states, since the model probabilities are 0.0002 for these models. The model probabilities in the last set of printed results should sum to one (ignoring possible rounding error in the truncated printed output).

Note that effects estimates from the SDEM versus SDM models are shown below based on the weight matrix W_b. The effects estimates do not lead to large differences in the conclusions we would draw regarding the impact of changes in the explanatory variables on the unemployment claims. This is consistent with the posterior model probabilities assigned to these two models (0.6326 vs. 0.3670) based on the weight matrix W_b. Of course, these results are consistent with the point made regarding log-marginal likelihoods for different model specifications. Estimates and inferences from different models will not be very different in situations where spatial dependence is low. In this situation, SLX, SDM and SDEM models all produce the same model fit to the sample data, and therefore the same scalar summary effects estimates.

```
Heterocedastic model estimates
             SDEM model estimates                         SDM model estimates
Direct     Coefficient        t-stat      t-prob       Coefficient        t-stat      t-prob
jobposts     -0.167982     -3.722425    0.000202         -0.156187     -3.865206    0.000114
athome        0.659354      4.077779    0.000047          0.656518      4.386910    0.000012

Indirect   Coefficient        t-stat      t-prob       Coefficient        t-stat      t-prob
jobposts     -0.076794     -0.933839    0.350479         -0.069234     -0.782596    0.433940
athome       -0.959907     -4.129705    0.000038         -1.037713     -4.477645    0.000008

Total      Coefficient        t-stat      t-prob       Coefficient        t-stat      t-prob
jobposts     -0.244777     -2.471448    0.013524         -0.225421     -2.150995    0.031574
athome       -0.300552     -1.530038    0.126136         -0.381195     -1.942272    0.052219
```

5.3 Comparing weight matrices for a given model specification

Another illustration of using the *lmarginal_static_panel()* function to compare spatial weight matrices involved a comparison of 10 different numbers of nearest neighbors. We illustrate a case where we have decided on the spatial regression model specification, for example, and SDM model and then proceed to compare a series of 10 different SDM models each based on a weight matrix constructed using a different number of equally weighted nearest neighbors.

The code stores the 10 different weight matrices in a *structure* variable named *Wmatrix(j).model*, where j ranges from 1 to 10 nearest neighbors. All weight matrices are based on the *same* latitude-longitude coordinates which were generated as random normal deviates.

A $y-$variable was generated based on a weight matrix that has 5 nearest neighbors. Log-marginal likelihoods are calculated for all 10 models by the *for iter=1:10* loop. The function *lmarginal_static_panel()* returns log-marginal likelihoods for the SLX, SDM and SDEM models, but we save only the returned value for the SDM model. (This was the true model used to generate the $y-$vector.)

```
% model_comparison_chapter5p3.m file
clear all;
sd = 221010;
rng(sd);

N = 400;
xc = randn(N,1);
yc = randn(N,1);

Wmatrix(1).model = make_neighborsw(xc,yc,1);

for j=2:10
        Wmatrix(j).model = make_neighborsw(xc,yc,j);
end

T = 10;
k = 2;
beta = [1
        2];
theta = [1.5
         -0.5];
bparms = [beta
          theta];
x = randn(N*T,k);
rho = 0.4;
sige = 1;
evec = randn(N*T,1)*sqrt(sige);

Wtrue = Wmatrix(5).model;
xbeta = [x  kron(eye(T),Wtrue)*x]*bparms;

% add fixed effects to the DGP
tts = (1:N)*(1/N);
SFE = kron(ones(T,1),tts');
ttt = (1:T)*(1/T);
TFE = kron(ttt',ones(N,1));

y = (speye(N*T) - rho*kron(eye(T),Wtrue))\(xbeta + SFE + TFE + evec);

xmat = [x kron(eye(T),Wtrue)*x];

lmarginal_save = [];
rnames = strvcat('# of neighbors');
```

```
for iter = 1:10
    W = Wmatrix(iter).model;
model = 3; % fixed effects for both regions and time periods
[ywith,xwith,meanny,meannx,meanty,meantx]=demean(y,xmat,N,T,model);

info.lflag = 0; % exact log-determinant
result = lmarginal_static_panel(ywith,xwith,W,N,T,info);

lmarginal_save = [lmarginal_save
                  result.logm_sdm];
rnames = strvcat(rnames,num2str(iter));
end

probs = model_probs(lmarginal_save);

in.rnames = rnames;
in.cnames = strvcat('log-marginal','prob');
in.width = 10000;
in.fmt = '%10.4f';
fprintf(1,'log-marginals for varying W-matrices \n');
mprint([lmarginal_save probs],in);
```

After storing log-marginal likelihoods for all 10 SDM models, we call a function *model_probs()* to calculates probabilities from a vector of log-marginal likelihoods input to the function. What we see is that the correct 5 nearest neighbors has a model probability of 1, with all other models having zero probabilities.

```
log-marginals for varying W-matrices
N = 400, rho = 0.4
# of neighbors log-marginal               prob
1              -7049.5492                 0.0000
2              -6980.0610                 0.0000
3              -6948.2477                 0.0000
4              -6920.6005                 0.0000
5              -6886.0737                 1.0000
6              -6913.8026                 0.0000
7              -6923.9645                 0.0000
8              -6934.5711                 0.0000
9              -6949.7847                 0.0000
10             -6963.0255                 0.0000
log-marginals for varying W-matrices
N = 200, rho = 0.2
# of neighbors log-marginal               prob
1              -3472.3730                 0.0000
2              -3466.0821                 0.0000
3              -3459.9610                 0.0000
4              -3454.8303                 0.0005
5              -3447.3366                 0.9766
6              -3452.2006                 0.0075
7              -3452.7329                 0.0044
8              -3452.0855                 0.0085
9              -3453.4688                 0.0021
10             -3455.2923                 0.0003
```

The results reported in the bottom part of the results above, are for a case where the sample size was decreased to $N = 200$ and a value of $\rho = 0.2$ was used. We would expect this experiment to produce poorer results. We do see some degradation in the model probabilities, but would still come the the correct conclusion that a 5 nearest neighbors is the correct weight matrix.

5.4 Comparing weight matrices and models

We can compare the SLX, SDM, SDEM model specifications for varying weight matrices *and* models. Once we have a *matrix* of log-marginal likelihoods we can create a vector of these and call the *model_probs()* function to calculate model probabilities across both weight matrices and model specifications.

The model probabilities are normalized to include the 3 model specifications times 10 different weight matrices, so our vector of 30 log-marginal likelihoods will have probabilities that sum to one.

The program below illustrates this by saving the 1×3 vectors of log-marginals from the *result.lmarginal* field returned by the *lmarginal_static_panel()* function in a 10×3 matrix named *lmarginal_save*.

```
% model_comparison_chapter5p4.m file
clear all;
sd = 221010;
rng(sd);

N = 400;
xc = randn(N,1);
yc = randn(N,1);
Wmatrix(1).model = make_neighborsw(xc,yc,1);

for j=2:10
        Wmatrix(j).model = make_neighborsw(xc,yc,j);
end

T = 10;
k = 2;
beta = [1
        2];
theta = [1.5
         -0.5];
bparms = [beta
          theta];
x = randn(N*T,k);
rho = 0.7;
sige = 1;
evec = randn(N*T,1)*sqrt(sige);

Wtrue = Wmatrix(5).model;
xbeta = [x  kron(eye(T),Wtrue)*x]*bparms;

% add fixed effects to the DGP
tts = (1:N)*(1/N);
SFE = kron(ones(T,1),tts');
```

```
ttt = (1:T)*(1/T);
TFE = kron(ttt',ones(N,1));

y = (speye(N*T) - rho*kron(eye(T),Wtrue))\(xbeta + SFE + TFE + evec);

xmat = [x kron(eye(T),Wtrue)*x];

lmarginal_save = zeros(10,3);
rnames = strvcat('# of neighbors');

for iter = 1:10
    W = Wmatrix(iter).model;
model = 3; % fixed effects for both regions and time periods
[ywith,xwith,meanny,meannx,meanty,meantx]=demean(y,xmat,N,T,model);
info.lflag = 0; % exact log-determinant
result = lmarginal_static_panel(ywith,xwith,W,N,T,info);

lmarginal_save(iter,:) = result.lmarginal';
rnames = strvcat(rnames,num2str(iter));
end

probs = model_probs(vec(lmarginal_save));

probs3 = reshape(probs,10,3);

in.rnames = rnames;
in.cnames = strvcat('SLX','SDM','SDEM');
in.width = 10000;
in.fmt = '%10.4f';
fprintf(1,'log-marginals for varying W-matrices AND models \n');
mprint(probs3,in);

log-marginals for varying W-matrices AND models
# of neighbors        SLX        SDM       SDEM
1                  0.0000     0.0000     0.0000
2                  0.0000     0.0000     0.0000
3                  0.0000     0.0000     0.0000
4                  0.0000     0.0000     0.0000
5                  0.0000     1.0000     0.0000
6                  0.0000     0.0000     0.0000
7                  0.0000     0.0000     0.0000
8                  0.0000     0.0000     0.0000
9                  0.0000     0.0000     0.0000
10                 0.0000     0.0000     0.0000
```

The matrix is then vectorized and used in the *model_probs()* function to find the 30 different model probabilities. I use the MATLAB *reshape()* function to produce a 10×3 matrix for printing the results. This facilitates comparing *both* models and weight matrices in the proper sequence.

The results point to the correct model *and* weight matrix that was used to generate the sample data.

5.5 Comparing SLX weight matrices and models

I mentioned that if the true model is the SLX, then we will not have a probability of 1 associated with this model, because in the case where $\rho = 0$, both the SDM and SDEM models collapse to the SLX specification. The following program demonstrates this by generating an SLX model as the true model, with a weight matrix based on 5 nearest neighbors.

```
% model_comparison_chapter5p5.m file
clear all;
sd = 221010;
rng(sd);
N = 200;
xc = randn(N,1);
yc = randn(N,1);

Wmatrix(1).model = make_neighborsw(xc,yc,1);
for j=2:10
        Wmatrix(j).model = make_neighborsw(xc,yc,j);
end

T = 10;
k = 2;
beta = [-1
        1];
theta = [0.5
         -0.5];
bparms = [beta
          theta];
x = randn(N*T,k);
sige = 1;
evec = randn(N*T,1)*sqrt(sige);

Wtrue = Wmatrix(5).model;
% SLX model x-matrix times beta
xbeta = [x  kron(eye(T),Wtrue)*x]*bparms;

% add fixed effects to the DGP
tts = (1:N)*(1/N);
SFE = kron(ones(T,1),tts');
ttt = (1:T)*(1/T);
TFE = kron(ttt',ones(N,1));
% slx model
y = (xbeta + SFE + TFE + evec);

lmarginal_save = zeros(10,3);
rnames = strvcat('# of neighbors');
model = 3; % fixed effects for both regions and time periods

for iter = 1:10
    W = Wmatrix(iter).model;
    xmat = [x kron(eye(T),W)*x];

[ywith,xwith,meanny,meannx,meanty,meantx]=demean(y,xmat,N,T,model);

info.lflag = 0; % exact log-determinant
```

```
result = lmarginal_static_panel(ywith,xwith,W,N,T,info);
lmarginal_save(iter,:) = result.lmarginal';
rnames = strvcat(rnames,num2str(iter));
end

probs = model_probs(vec(lmarginal_save));
probs3 = reshape(probs,10,3);

in.rnames = rnames;
in.cnames = strvcat('SLX','SDM','SDEM');
in.width = 10000;
in.fmt = '%10.4f';
fprintf(1,'log-marginals for varying W-matrices AND models \n');
mprint(probs3,in);
```

```
log-marginals for varying W-matrices AND models
# of neighbors        SLX        SDM       SDEM
1                  0.0000     0.0000     0.0000
2                  0.0000     0.0000     0.0000
3                  0.0000     0.0000     0.0000
4                  0.0000     0.0000     0.0000
5                  0.3435     0.3014     0.3551
6                  0.0000     0.0000     0.0000
7                  0.0000     0.0000     0.0000
8                  0.0000     0.0000     0.0000
9                  0.0000     0.0000     0.0000
10                 0.0000     0.0000     0.0000
```

The results shown above do point to the SLX model with a 5 nearest neighbors weight matrix, since all three model probabilities are close to 0.33. However, unlike the case of the SDM model, we do not see probabilities close to 1.0 for the SLX model with 5 neighbors. Instead, we see nearly equal model probabilities, indicating that it is difficult to distinguish between an SLX, SDM and SDEM when $\rho = 0$, which we know to be theoretically true.

We would in this case select the SLX model specification, but we could check to see if $\rho = 0$ by estimating the SDM and SDEM specifications.

To examine the limiting behavior, I produced results based on a very small noise variance $\sigma^2 = 0.001$ and a sample size of 10,000 (in the file: *model_comparison_chatper5p5b.m*), which produced the results shown below. Since we have a large sample size, I used the log-determinant approximation to produce these results, which took 5.7 seconds. (Using the exact log-determinant took — seconds.)

```
log-marginals for varying W-matrices AND models
# of neighbors        SLX        SDM       SDEM
1                  0.0000     0.0000     0.0000
2                  0.0000     0.0000     0.0000
3                  0.0000     0.0000     0.0000
4                  0.0000     0.0000     0.0000
5                  0.4970     0.2515     0.2516
6                  0.0000     0.0000     0.0000
7                  0.0000     0.0000     0.0000
8                  0.0000     0.0000     0.0000
9                  0.0000     0.0000     0.0000
10                 0.0000     0.0000     0.0000
```

We see strong evidence in favor of the true SLX model with a model probability of near 0.5 for the case of a 5 neighbors weight matrix. The remaining 0.5 probability is distributed equally across the SDM and SDEM models, which of course in this case are very similar to each other and also similar to the SLX model. Note that despite the fact that $\rho = 0$ in the DGP, some non-zero values of ρ may have some support during integration, leading to the mixed results that indicate some support for the SDM and SDEM specifications.

5.6 Chapter summary

We have presented a series of convex combination of weight matrices models that extend the flexibility of the spatial regression models to allow for more than simply spatial connectivity or dependence between observations. We can for example construct weight matrices that reflect dependence based on commodity or migration flows between regions which are very different from spatial connectivity.

Once we open the door to use of multiple weight matrices, questions regarding the number of matrices and which specific matrices to incorporate in our models arise. The next chapter tackles comparison of models based on differing number of weight matrices.

Unlike the case in ordinary least-squares regression where addition of irrelevant explanatory variables produces no bias in estimates associated with relevant explanatory variables, the convex combination of weights spatial regression models can suffer from the introduction of irrelevant weight matrices. To see this, consider that irrelevant weight matrices will lead to posterior estimates for the convex combination parameters that we have labeled $\gamma_m, m = 1, \ldots, M$ that are non-zero, having distributions that pile-up at the lower and upper bounds of the 0,1 parameter space for these parameters. A large number of such irrelevant weight matrices will then subtract weight from the γ parameters associated with the relevant weights.

5.7 Chapter references

Barry, R.P. and R.K. Pace (1999). A Monte Carlo Estimator of the Log Determinant of Large Sparse Matrices, *Linear Algebra and its Applications*, 289, 41-54.

Debarsy, N. and LeSage, J.P. 2021. Bayesian model averaging for spatial autoregressive models based on convex combinations of different types of connectivity matrices, *Journal of Business & Economic Statistics*

Grieser, W, LeSage, J.P., Zekhnini, M. (2021) Industry Networks and the Geography of Firm Behavior, working paper, Available at SSRN: https://ssrn.com/abstract=3555390 or http://dx.doi.org/10.2139/ssrn.3555390

Grieser, W, Hadlock, C., LeSage, J.P., Zekhnini, M. (2021) Network Effects in Corporate Financial Policies, working paper.

Koop, G. (2003). *Bayesian Econometrics*, West Sussex, England: John Wiley & Sons.

LeSage James P. (2014). Spatial econometric panel data model specification: A Bayesian approach, *Spatial Statistics* Volume 9, Issue C, 122-145.

LeSage, J. P. and R. K. Pace. 2014. The biggest myth in spatial econometrics, *Econometrics*, Volume 2, Issue 4, pp. 217-249. doi:10.3390/econometrics2040217. http://www.mdpi.com/2225-1146/2/4/217

LeSage, James P. (2015). Software for Bayesian cross section and panel spatial model comparison, *Journal of Geographical Systems*, 17, 297-310.

LeSage, J.P. (2020). Fast MCMC estimation of multiple W-matrix spatial regression models and Metropolis Hastings-Monte Carlo log marginal likelihoods *Journal of Geographical Systems*, 22:1, 47-75.

LeSage, J. P. and R. K. Pace (2014). Interpreting Spatial Econometric Models, *Handbook of Regional Science*, M. M. Fischer and Peter Nijkamp (Eds.) Springer, Berlin 2014, pp. 1535-1552.

Pace, R.K. and R.P. Barry (1997). Quick computation of spatial autoregressive estimators, *Geographical Analysis*, 29, 232-246.

Zellner, A. (1971). *An Introduction to Bayesian Inference in Econometrics.* New York: John Wiley & Sons.

Chapter 6

Model comparison for convex combination models

6.1 Comparing convex combination spatial regression models

We can also use the log-marginal likelihood to compare convex combination of weights models, but in these models the parameters cannot be integrated out using a simple univariate numerical integration routine. This is because analytical integration of the parameters β, σ^2 leaves us with the joint posterior distribution of the parameters ρ, Γ which must be integrated out. Multivariate numerical integration is computationally intensive. LeSage (2020) and Debarsy and LeSage (2021) tackle the problem of integrating over the parameters $\rho, \gamma_1, \ldots, \gamma_M$ using Metropolis-Hastings guided Monte Carlo integration.

Monte Carlo integration evaluates the expression to be integrated using random draws of the parameter values. A drawback to this approach is inefficiency because many of the random draws for the parameters are not in areas of high density of the function being integrated. LeSage (2020) suggests that during estimation of the model parameters, the Metropolis-Hastings sampling procedure used to produce draws of the dependence parameters can be used for Monte Carlo integration. This should work well because Metropolis-Hastings sampling of the parameters ρ, Γ should steer these sampled parameter values to areas of high density of the joint posterior. This allows us to produce an efficient Monte Carlo integration of the log-marginal likelihood.

A drawback to this approach is that we need to actually estimate the models being compared to produce our *estimated* log-marginal likelihood. This is different from the situation discussed and illustrated for the conventional static panel data models where univariate numerical integration could be carried out without estimating the models being compared. This also means that one *cannot legitimately* compare the log-marginal likelihoods from static models produced by numerical integration to those from the convex combination models produced via Metropolis-Hastings guided Monte Carlo integration.

The convex combination of connectivity matrices model specification raises the question of which matrices should be used and which should be ignored. For example, in the case of L candidate connectivity matrices, there are $M = 2^L - L - 1$ possible ways to employ *two or more* of the L matrices in alternative model specifications. When $L = 5$, we have $M = 26$ possible models involving two or more matrices, and for $L = 10$, $M = 1,013$. To account for this model uncertainty, we rely on Metropolis-Hastings guided Monte Carlo integration during MCMC estimation of the models to produce log-marginal likelihoods and associated posterior model probabilities for the set of M possible models, which allows for Bayesian model averaged estimates. Given the speed of estimation for single convex combinations of weights models and the availability of multi-core computer architecture, it is possible to estimate models based on all possible combinations of two or more connectivity matrices, even in cases of 10 matrices.

This chapter discusses Bayesian model averaging for the case of the convex combination of weight matrices static panel data model specifications. We tackle the problem of model uncertainty inherent in models involving numerous connectivity matrices by relying on a Bayesian model averaging procedure.

6.2 The SAR convex combination BMA model

Given the non-linear relationship between the underlying parameters β, Γ, ρ and the scalar summary measures of direct and indirect effects which are the focus of inference in these models, model averaged estimates should be constructed by applying model probabilities to the scalar summary estimates of the direct and indirect effects from each model.

As an illustration, we present estimation results for all models ($M = 26$) involving two or more connectivity matrices using a set of five candidate $W-$matrices and a sample of $N = 400, T = 5$ or $N \times T = 2000$ observations. This takes only 37 seconds on a Dell XPS 15-inch 9570 Laptop with 6 cores and an Intel Core i9-8950HK CPU, with 32 GB of RAM memory. I am using MATLAB Version: 9.10.0.1602886 (R2021a). Note that you can use the MATLAB command *ver* to see if you have the Parallel Computing Toolbox installed. If not, you should set the input prior.parallel = 0 input option to the function *sar_conv_panel_bma_g.m* which will use a simple *for* loop command in place of the *parfor* loop. The default value is parallel = 1, so you *must* use the option *prior.parallel = 0* to turn this off if you do not have the Parallel Computing Toolbox installed..

For this program, it took 29 seconds to produce results using parallel processing versus 87 seconds using standard for loops to estimate each model separately.

The program code is shown below where we generate 5 weight matrices, but set the true $\gamma_5 = 0$, so that only 4 weights are used in the DGP that generates y. The 5 weight matrices are based on varying numbers of nearest neighbors, with different random latitude-longitude vectors used to produce each weight matrix. This means that the weight matrices could be viewed as reflecting different types of network structures and should not be viewed as merely testing for the appropriate number of nearest neighbors to use. LeSage and Pace (2014) point out that weights based on differing numbers of nearest neighbors are highly correlated, if one considers the correlation between $W_1u, W_2u, \ldots, W_Mu$, so this would not be a good way to proceed.

The table below shows that these matrices are *not* highly correlated.

```
 1.0000    0.0036    0.0075   -0.0470    0.0168
 0.0036    1.0000   -0.0129    0.0705   -0.0329
 0.0075   -0.0129    1.0000   -0.0105    0.0953
-0.0470    0.0705   -0.0105    1.0000    0.0453
 0.0168   -0.0329    0.0953    0.0453    1.0000
```

The program will rely on the Metropolis-Hastings guided Monte Carlo integration procedure to produce log-marginal likelihoods for each of the 26 models. It will print out the results of these calculations showing estimated values for the parameters γ as well as log-marginal likelihood estimates, and ρ estimates for each model. Of the 26 different models, model #21 is the one used in the DGP to produce the vector y. Of course, all of the $X-$variables remain the same when estimating the 26 different models. It is not valid to compare models with different explanatory variables using the log-marginal likelihood estimates from these models because they do not rely on priors assigned to the parameters β, σ^2.

```
% sar BMA program for sar_conv_panel_bma_gd.m
clear all;
sd = 221010;
rng(sd);

% estimate all possible models
% with two or more W-matrices
nweights = 5;

% np = 26
n = 400;
t = 5;
xc = randn(n,1);  % generate 5 W-matrices
yc = randn(n,1);
W1 = make_neighborsw(xc,yc,5); % 5 nearest neighbors W-matrix

xc = randn(n,1);
yc = randn(n,1);
W2 = make_neighborsw(xc,yc,8); % 8 nearest neighbors W-matrix

xc = randn(n,1);
yc = randn(n,1);
W3 = make_neighborsw(xc,yc,10); % 10 nearest neighbors W-matrix

xc = randn(n,1);
yc = randn(n,1);
W4 = make_neighborsw(xc,yc,6); % 6 nearest neighbors W-matrix

xc = randn(n,1);
yc = randn(n,1);
W5 = make_neighborsw(xc,yc,4); % 4 nearest neighbors W-matrix

gamma1 = 0.3; % assign gamma weights
gamma2 = 0.3;
gamma3 = 0.2;
gamma4 = 0.2;
gamma5 = 0.0; % W5 is not really in the model
gtrue = [gamma1
         gamma2
         gamma3
         gamma4
         gamma5];
%
Wc = gamma1*W1 + gamma2*W2 + gamma3*W3 + gamma4*W4 + gamma5*W5;
u = randn(n,1);

corrcoef([W1*u W2*u W3*u W4*u W5*u])
```

```
k=4; % 4 explanatory variables
x = [randn(n*t,k)];
beta = [1
        -1
        -0.5
        1.5];
btrue = beta;
sige = 1;
strue = sige;
rho = 0.7;
ptrue = rho;
% generate True model
% add fixed effects to the DGP
tts = (1:n)*(1/n);
SFE = kron(ones(t,1),tts');
ttt = (1:t)*(1/t);
TFE = kron(ttt',ones(n,1));

y = (speye(n*t) - rho*kron(eye(t),Wc))\(x*beta + SFE + TFE + randn(n*t,1)*sqrt(sige));

ndraw = 30000;
nomit = 10000;
prior.thin = 5; % retains only 4000 draws from 30,000
                % by skipping every 5
prior.model = 3; % fixed effects

Wmatrices = [W1 W2 W3 W4 W5];

% Estimation of Bayesian model averaging estimates using five matrices
result = sar_conv_panel_bma_g(y,x,Wmatrices,n,t, ndraw, nomit, prior);
vnames = strvcat('y','x1','x2','x3','x4');
prt_panel_bma(result, vnames);

prior.parallel = 0; % turn off parallel processing
result = sar_conv_panel_bma_g(y,x,Wmatrices,n,t, ndraw, nomit, prior);
prt_panel_bma(result, vnames);

% results printed by the sar_conv_panel_bma_g() function
% file: sar_convg_panel_bma_gd

Starting parallel pool (parpool) using the 'local' profile ...
Connected to the parallel pool (number of workers: 6).
Models           logm     Prob     rho      W1      W2      W3      W4      W5
Model 1     -3334.294    0.000   0.400   0.580   0.420   0.000   0.000   0.000
Model 2     -3350.817    0.000   0.364   0.633   0.000   0.367   0.000   0.000
Model 3     -3343.130    0.000   0.366   0.634   0.000   0.000   0.366   0.000
Model 4     -3364.280    0.000   0.257   0.901   0.000   0.000   0.000   0.099
Model 5     -3407.411    0.000   0.302   0.000   0.551   0.449   0.000   0.000
Model 6     -3400.439    0.000   0.305   0.000   0.559   0.000   0.441   0.000
Model 7     -3420.696    0.000   0.193   0.000   0.865   0.000   0.000   0.135
Model 8     -3414.925    0.000   0.275   0.000   0.000   0.514   0.486   0.000
Model 9     -3434.002    0.000   0.169   0.000   0.000   0.819   0.000   0.181
Model 10    -3428.947    0.000   0.161   0.000   0.000   0.000   0.821   0.179
Model 11    -3323.569    0.000   0.522   0.439   0.316   0.245   0.000   0.000
Model 12    -3313.926    0.000   0.537   0.432   0.317   0.000   0.251   0.000
```

```
Model 13    -3336.807   0.000   0.418   0.550   0.397   0.000   0.000   0.054
Model 14    -3331.249   0.000   0.497   0.463   0.000   0.267   0.270   0.000
Model 15    -3352.846   0.000   0.389   0.586   0.000   0.344   0.000   0.069
Model 16    -3345.278   0.000   0.388   0.594   0.000   0.000   0.341   0.066
Model 17    -3388.758   0.000   0.439   0.000   0.383   0.310   0.307   0.000
Model 18    -3409.446   0.000   0.326   0.000   0.499   0.416   0.000   0.085
Model 19    -3402.612   0.000   0.327   0.000   0.512   0.000   0.407   0.081
Model 20    -3416.581   0.000   0.303   0.000   0.000   0.460   0.437   0.103
Model 21    -3302.692   0.922   0.658   0.348   0.253   0.194   0.205   0.000
Model 22    -3325.929   0.000   0.542   0.420   0.303   0.235   0.000   0.042
Model 23    -3316.350   0.000   0.554   0.416   0.302   0.000   0.243   0.039
Model 24    -3333.318   0.000   0.519   0.440   0.000   0.255   0.255   0.050
Model 25    -3390.735   0.000   0.462   0.000   0.358   0.292   0.289   0.061
Model 26    -3305.161   0.078   0.676   0.337   0.243   0.189   0.199   0.033
BMA         -3302.885   1.000   0.660   0.347   0.252   0.194   0.205   0.003
highest     -3302.692   0.922   0.658   0.348   0.253   0.194   0.205   0.000
```

The printed results above show that model #21 receives the highest posterior model probability of 0.922, with model #26 being the only other model that receives non-zero probability equal to 0.078. Note that model #26 is one that relied on all 5 weight matrices, whereas model #21 was based on the true weights of $W1, W2, W3, W4$. Model #26 produces an estimate $\hat{\gamma}_5 = 0.033$, which detracts from the estimated $\gamma_i, i = 1, \ldots, 4$ values for this incorrect model. The model averaged estimates are constructed by using the posterior model probabilities to form a linear combination of the parameters from models #21 and #26, are are shown in the row labeled BMA. The highest probability model is shown below the model averaged estimates for ease of comparison in the results printed by the function. We see that $\hat{\gamma}_5 = 0.0024$ is downweighted by the model averaging procedure, which would allow the other $\gamma_i = 1, \ldots, 4$ parameters to come closer to the true values. Of course, in this example, since the true model receives almost all of the posterior model probability, the model averaged estimates are not very different from those of model #21.

The call to the printing function produces a printout of the model averaged estimates for the parameters β, ρ, Γ as well as model averaged estimates for the direct, indirect and total effects estimates. These must be calculated by applying the model probabilities to the effects estimates from all the models. This is because they are non-linear functions of the underlying parameters β, ρ, (see LeSage and Pace (2009, Chapter 5). Note that you need to call a special function named *prt_panel_bma()*, **not** the usual *prt_panel()* function.

```
Bayesian Model Average of SAR convex panel W models
Dependent Variable  =              y
BMA Log-marginal    = -3302.8848
Nobs, T, Nvars      =    400,     5,     4
# weight matrices   =      5
ndraws,nomit        =  30000, 10000
total time          =   29.7820
thinning for draws  =      5
min and max rho     =   -1.0000,   1.0000
***************************************************************
      MCMC diagnostics ndraws = 4000
Variable         Mean     MC error        tau     Geweke
x1             0.9602   0.00031805   0.909862   0.998519
x2            -0.9386   0.00032089   1.098095   0.998504
```

```
x3            -0.5324   0.00027172   0.887941   0.996799
x4             1.5246   0.00024351   0.910044   0.999907
rho            0.6597   0.00088000   2.370429   0.996973
gamma1         0.3469   0.00086871   4.544389   0.992425
gamma2         0.2523   0.00079841   3.640595   0.998074
gamma3         0.1936   0.00050939   4.019119   0.987590
gamma4         0.2048   0.00061409   1.874944   0.998329
gamma5         0.0026   0.00004331   2.578434   0.978089

*************************************************************
      Posterior Estimates
Variable       lower 0.01       lower 0.05          median       upper 0.95       upper 0.99
x1               0.908293         0.921199        0.959423         1.002934         1.014602
x2              -0.992031        -0.977853       -0.938696        -0.898901        -0.885991
x3              -0.587649        -0.572023       -0.532334        -0.492739        -0.476977
x4               1.473872         1.484745        1.524707         1.563374         1.575763
rho              0.561660         0.586547        0.659938         0.734560         0.754014
gamma1           0.289254         0.300060        0.346915         0.394131         0.410587
gamma2           0.184427         0.201000        0.252238         0.302776         0.323691
gamma3           0.116318         0.134532        0.193973         0.249531         0.266679
gamma4           0.138910         0.155986        0.204532         0.252267         0.268513
gamma5           0.000030         0.000152        0.002402         0.005789         0.007084

Direct         lower 0.01       lower 0.05          median       upper 0.95       upper 0.99
x1               0.925821         0.938606        0.977972         1.021638         1.034422
x2              -1.011439        -0.996568       -0.957034        -0.915999        -0.903731
x3              -0.598884        -0.583297       -0.542814        -0.501948        -0.487369
x4               1.500957         1.512914        1.554193         1.593557         1.605477

Indirect       lower 0.01       lower 0.05          median       upper 0.95       upper 0.99
x1               1.230628         1.355490        1.848133         2.646928         2.956422
x2              -2.881563        -2.583970       -1.806800        -1.316233        -1.210320
x3              -1.644103        -1.466783       -1.022836        -0.747078        -0.685916
x4               1.967549         2.169728        2.934098         4.195444         4.693646

Total          lower 0.01       lower 0.05          median       upper 0.95       upper 0.99
x1               2.200192         2.323225        2.825973         3.648605         3.951802
x2              -3.842684        -3.554333       -2.765301        -2.271758        -2.157343
x3              -2.204655        -2.021385       -1.565308        -1.279665        -1.214096
x4               3.505393         3.709850        4.487071         5.735890         6.253441
```

Another point is that running this program more than once with the Parallel Computing Toolbox (even if the random number generator seed is fixed) seems to produce slightly different results. This does not happen if you do not use parallel computing, so I this is an issue with the way the random number seed is applied by the Parallel Computing Toolbox.[1] Of course, if you see dramatically different results from multiple runs of the program, you have an MCMC convergence problem. You might increase the number of MCMC draws and see if the problem persists. It is good practice with MCMC estimation to see if estimates based on a larger number of draws produce different estimation results. Three runs of this program produced model probabilities of 0.919, 0.921, and 0.922 for model #21, so results are not very different. These results are likely inline with the Monte Carlo error associated with the Metropolis-Hastings Monte Carlo approximation used to

[1] This issue has received some attention in the MATLAB forums, so it is not exclusive to me.

estimate the log-marginal likelihood. Results from two runs of the program with the input option *prior.parallel = 0* are shown below, where we see identical results.

```
% 1st run using standard for-loop
Models          logm    Prob     rho      W1      W2      W3      W4      W5
Model 1     -3334.283   0.000   0.400   0.579   0.421   0.000   0.000   0.000
Model 2     -3350.834   0.000   0.363   0.633   0.000   0.367   0.000   0.000
Model 3     -3343.157   0.000   0.366   0.634   0.000   0.000   0.366   0.000
Model 4     -3364.286   0.000   0.256   0.902   0.000   0.000   0.000   0.098
Model 5     -3407.440   0.000   0.301   0.000   0.552   0.448   0.000   0.000
Model 6     -3400.448   0.000   0.305   0.000   0.558   0.000   0.442   0.000
Model 7     -3420.681   0.000   0.194   0.000   0.864   0.000   0.000   0.136
Model 8     -3414.948   0.000   0.275   0.000   0.000   0.515   0.485   0.000
Model 9     -3434.024   0.000   0.169   0.000   0.000   0.821   0.000   0.179
Model 10    -3428.927   0.000   0.161   0.000   0.000   0.000   0.817   0.183
Model 11    -3323.548   0.000   0.523   0.438   0.317   0.245   0.000   0.000
Model 12    -3313.928   0.000   0.537   0.431   0.317   0.000   0.252   0.000
Model 13    -3336.836   0.000   0.418   0.549   0.398   0.000   0.000   0.053
Model 14    -3331.228   0.000   0.497   0.462   0.000   0.268   0.270   0.000
Model 15    -3352.826   0.000   0.387   0.590   0.000   0.343   0.000   0.067
Model 16    -3345.320   0.000   0.388   0.595   0.000   0.000   0.340   0.065
Model 17    -3388.715   0.000   0.439   0.000   0.382   0.312   0.307   0.000
Model 18    -3409.407   0.000   0.324   0.000   0.505   0.411   0.000   0.083
Model 19    -3402.629   0.000   0.326   0.000   0.516   0.000   0.405   0.079
Model 20    -3416.563   0.000   0.303   0.000   0.000   0.458   0.439   0.103
Model 21    -3302.642   0.921   0.659   0.348   0.252   0.194   0.206   0.000
Model 22    -3325.964   0.000   0.542   0.420   0.301   0.236   0.000   0.042
Model 23    -3316.407   0.000   0.552   0.418   0.302   0.000   0.241   0.039
Model 24    -3333.314   0.000   0.520   0.441   0.000   0.253   0.256   0.050
Model 25    -3390.745   0.000   0.460   0.000   0.359   0.293   0.288   0.059
Model 26    -3305.099   0.079   0.678   0.336   0.243   0.188   0.200   0.033
BMA         -3302.836   1.000   0.660   0.347   0.251   0.194   0.205   0.003
highest     -3302.642   0.921   0.659   0.348   0.252   0.194   0.206   0.000
```

```
% 2nd run using standard for-loop
Models          logm    Prob     rho      W1      W2      W3      W4      W5
Model 1     -3334.283   0.000   0.400   0.579   0.421   0.000   0.000   0.000
Model 2     -3350.834   0.000   0.363   0.633   0.000   0.367   0.000   0.000
Model 3     -3343.157   0.000   0.366   0.634   0.000   0.000   0.366   0.000
Model 4     -3364.286   0.000   0.256   0.902   0.000   0.000   0.000   0.098
Model 5     -3407.440   0.000   0.301   0.000   0.552   0.448   0.000   0.000
Model 6     -3400.448   0.000   0.305   0.000   0.558   0.000   0.442   0.000
Model 7     -3420.681   0.000   0.194   0.000   0.864   0.000   0.000   0.136
Model 8     -3414.948   0.000   0.275   0.000   0.000   0.515   0.485   0.000
Model 9     -3434.024   0.000   0.169   0.000   0.000   0.821   0.000   0.179
Model 10    -3428.927   0.000   0.161   0.000   0.000   0.000   0.817   0.183
Model 11    -3323.548   0.000   0.523   0.438   0.317   0.245   0.000   0.000
Model 12    -3313.928   0.000   0.537   0.431   0.317   0.000   0.252   0.000
Model 13    -3336.836   0.000   0.418   0.549   0.398   0.000   0.000   0.053
Model 14    -3331.228   0.000   0.497   0.462   0.000   0.268   0.270   0.000
Model 15    -3352.826   0.000   0.387   0.590   0.000   0.343   0.000   0.067
Model 16    -3345.320   0.000   0.388   0.595   0.000   0.000   0.340   0.065
Model 17    -3388.715   0.000   0.439   0.000   0.382   0.312   0.307   0.000
Model 18    -3409.407   0.000   0.324   0.000   0.505   0.411   0.000   0.083
Model 19    -3402.629   0.000   0.326   0.000   0.516   0.000   0.405   0.079
Model 20    -3416.563   0.000   0.303   0.000   0.000   0.458   0.439   0.103
```

```
Model 21    -3302.642   0.921   0.659   0.348   0.252   0.194   0.206   0.000
Model 22    -3325.964   0.000   0.542   0.420   0.301   0.236   0.000   0.042
Model 23    -3316.407   0.000   0.552   0.418   0.302   0.000   0.241   0.039
Model 24    -3333.314   0.000   0.520   0.441   0.000   0.253   0.256   0.050
Model 25    -3390.745   0.000   0.460   0.000   0.359   0.293   0.288   0.059
Model 26    -3305.099   0.079   0.678   0.336   0.243   0.188   0.200   0.033
BMA         -3302.836   1.000   0.660   0.347   0.251   0.194   0.205   0.003
highest     -3302.642   0.921   0.659   0.348   0.252   0.194   0.206   0.000
```

6.3 The SDM convex combination BMA model

There is a problem with calculating the log-marginal likelihood for SDM models that rely on a specification that includes spatial lags of the explanatory variables based on all $W-$matrices, e.g., $W_1X, W_2X, \ldots, W_MX$. The problem is that when considering all possible models involving two or more W_m, the number of explanatory variables in the matrix $\left(X \quad \sum_{m=1}^{M} W_m X \right)$ will change when we compare a model based on W_1, W_2 and (say) a model based on W_1, W_2, W_3. Allowing the size of the parameter space to change when calculating log-marginal likelihoods will encounter a problem that has been labeled the *Lindley Paradox*. Koop (2003) discusses this issue, indicating that unless we assign proper Bayesian prior distributions on the model parameters in this type of situation, the log-marginal likelihoods will always produce higher model probabilities for a more parsimonious model, one with fewer explanatory variables.

To be clear about this, the SDM model specification is shown in (6.1), where we see that each matrix $W_i, i = 1, \ldots, m$ contributes explanatory variables to the model.

$$\begin{aligned} y &= \rho W_c(\Gamma) y + X\beta + \sum_{m=1}^{M} W_m X \theta_m + \varepsilon, \quad \varepsilon \sim N(0, \sigma^2 I_n), \qquad (6.1) \\ W_c(\Gamma) &= \sum_{m=1}^{M} \gamma_m W_m, \quad 0 \leq \gamma_m \leq 1, \ \sum_{m=1}^{M} \gamma_m = 1, \\ \Gamma &= (\gamma_1, \ldots, \gamma_M)' \end{aligned}$$

What will this mean for model comparison of the SDM convex combination model as specified based on using all weight matrices to create spatial lags of the explanatory variables? To illustrate the possible problem, results are shown below for the case of three W_1, W_2, W_3 with the the true values of $\gamma_1 = 0.5, \gamma_2 = 0.3, \gamma_3 = 0.2$, so all three weights appear in the model that generated the $y-$vector.

```
% file: sdm_conv_panel_bma_gd.m
% sdm BMA program for sdm_conv_panel_bma_gd.m
clear all;
sd = 221010;
rng(sd);

% estimate all possible models
% with two or more W-matrices
% nweights = 3, so we have 4 models with 2 or more W-matrices

n = 800;
```

```
t = 5;
xc = randn(n,1);  % generate 3 W-matrices
yc = randn(n,1);
W1 = make_neighborsw(xc,yc,5); % 5 nearest neighbors W-matrix
xc = randn(n,1);
yc = randn(n,1);
W2 = make_neighborsw(xc,yc,8); % 8 nearest neighbors W-matrix
xc = randn(n,1);
yc = randn(n,1);
W3 = make_neighborsw(xc,yc,12); % 12 nearest neighbors W-matrix

gamma1 = 0.5; % assign gamma weights
gamma2 = 0.3;
gamma3 = 0.2;

Wc = gamma1*W1 + gamma2*W2 + gamma3*W3;

k=4; % 4 explanatory variables
x = [randn(n*t,k)];
beta = [1
        1
        1
        1];
theta1 = 0.5*beta;
theta2 = -0.75*beta;
theta3 =  beta;

bvec = [beta
        theta1
        theta2
        theta3];
sige = 1;
rho = 0.7;

% generate True model
% add fixed effects to the DGP
tts = (1:n)*(1/n);
SFE = kron(ones(t,1),tts');
ttt = (1:t)*(1/t);
TFE = kron(ttt',ones(n,1));

    Wx = [x kron(speye(t),W1)*x kron(speye(t),W2)*x kron(speye(t),W3)*x];
    Wxb = Wx*bvec;

y = (speye(n*t) - rho*kron(eye(t),Wc))\(Wxb + SFE + TFE + randn(n*t,1)*sqrt(sige));

ndraw = 50000;
nomit = 10000;
prior.thin = 5; % retains only 8000 draws from 40,000
                % by skipping every 5
prior.model = 3; % fixed effects

Wmatrices = [W1 W2 W3];

% Estimation of Bayesian model averaging estimates using three matrices
```

```
result = sdm_conv_panel_bma_g(y,x,Wmatrices,n,t, ndraw, nomit, prior);
```

Models	logm	Prob	rho	W1	W2	W3
Model 1	-7730.514	0.000	0.556	0.638	0.362	0.000
Model 2	-7116.520	0.000	0.514	0.694	0.000	0.306
Model 3	-8187.989	0.000	0.333	0.000	0.601	0.399
Model 4	-6837.422	1.000	0.676	0.512	0.287	0.201
BMA	-6837.422	1.000	0.676	0.512	0.287	0.201
highest	-6837.422	1.000	0.676	0.512	0.287	0.201

Note that the function *sdm_conv_panel_bma_g()* prints out the results shown. From the results, we see that they point to the true model that contains all W_1, W_2, W_3, and the estimated parameters $\rho, \gamma_1, \gamma_2, \gamma_3$ are reasonably close to the true values used in the DGP.

We can push this example further by examining a situation where there are many more weight matrices to see if the results point to a more parsimonious model. Results are shown below for the case of 5 weight matrices and a model with only two explanatory variables in the matrix X.

The results again point to the correct model (#26) based on all 5 weight matrices, and the estimated $\rho = 0.653$ is reasonably close the the true value of 0.7, as are the estimated $\gamma_i, i = 1, \dots, 5$ which were all set to values of 0.20 in the DGP.

Models	logm	Prob	rho	W1	W2	W3	W4	W5
Model 1	-7186.616	0.000	0.228	0.576	0.424	0.000	0.000	0.000
Model 2	-7170.987	0.000	0.323	0.410	0.000	0.590	0.000	0.000
Model 3	-7211.843	0.000	0.298	0.423	0.000	0.000	0.577	0.000
Model 4	-6972.640	0.000	0.225	0.583	0.000	0.000	0.000	0.417
Model 5	-7346.368	0.000	0.302	0.000	0.318	0.682	0.000	0.000
Model 6	-7386.810	0.000	0.270	0.000	0.351	0.000	0.649	0.000
Model 7	-7175.809	0.000	0.202	0.000	0.504	0.000	0.000	0.496
Model 8	-7373.961	0.000	0.360	0.000	0.000	0.532	0.468	0.000
Model 9	-7156.033	0.000	0.308	0.000	0.000	0.637	0.000	0.363
Model 10	-7200.472	0.000	0.267	0.000	0.000	0.000	0.608	0.392
Model 11	-7090.384	0.000	0.427	0.312	0.229	0.459	0.000	0.000
Model 12	-7135.128	0.000	0.396	0.320	0.242	0.000	0.438	0.000
Model 13	-6891.927	0.000	0.321	0.413	0.325	0.000	0.000	0.262
Model 14	-7124.353	0.000	0.478	0.268	0.000	0.381	0.350	0.000
Model 15	-6873.470	0.000	0.416	0.322	0.000	0.445	0.000	0.234
Model 16	-6922.756	0.000	0.377	0.337	0.000	0.000	0.426	0.237
Model 17	-7302.625	0.000	0.462	0.000	0.207	0.425	0.368	0.000
Model 18	-7081.909	0.000	0.405	0.000	0.251	0.496	0.000	0.253
Model 19	-7129.896	0.000	0.359	0.000	0.279	0.000	0.456	0.265
Model 20	-7114.834	0.000	0.451	0.000	0.000	0.414	0.348	0.238
Model 21	-7040.724	0.000	0.580	0.223	0.165	0.321	0.290	0.000
Model 22	-6784.719	0.000	0.516	0.263	0.201	0.368	0.000	0.168
Model 23	-6838.716	0.000	0.472	0.271	0.219	0.000	0.343	0.167
Model 24	-6825.512	0.000	0.553	0.238	0.000	0.316	0.280	0.166
Model 25	-7037.794	0.000	0.545	0.000	0.183	0.348	0.291	0.177
Model 26	-6733.362	1.000	0.653	0.203	0.157	0.275	0.240	0.125
BMA	-6733.362	1.000	0.653	0.203	0.157	0.275	0.240	0.125
highest	-6733.362	1.000	0.653	0.203	0.157	0.275	0.240	0.125

Despite these promising results, there are a number of factors at play when comparing SDM convex combination models. A discussion of some of these issues follows.

One important factor that will determine the accuracy of results is the coefficients on the $W_1X, W_2X, \ldots W_mX$ variables. Intuitively, if all of the coefficients associated with these variables are zero, then the SDM model collapses back to the SAR model. In this case, we should have no issues arising from changes in the number of explanatory variables when comparing alternative models, since all models would only be based on the $X-$matrix.

In generating the second example with 5 weight matrices, I set the parameters $\beta = \begin{pmatrix} 1 & 1 \end{pmatrix}'$. Parameters associated with the $W_1X, W_2X, \ldots, W_5X$ were set to: $0.5{*}\beta, -0.5{*}\beta, 0.25\beta 0.25\beta, -0.25\beta, 0.5{*}\beta$, so these were far away from zero values. This means that the good performance of the SDM model results was not because in reality we were working with a SAR model, where there are no issues associated with changing the model dimension.

Of course, results will depend on the standard features of spatial regression models, like the strength of dependence and the correlation/similarity of weight matrices used. The signal-to-noise or model fit will also be important in determining accuracy of model comparisons, and of course the size of the weight matrices will matter.

Results are shown below for a case where the true model involved use of 3 $W-$matrices in the DGP, but 5 weight matrices were used as inputs to the *sdm_conv_panel_bma_g()* function. Here, the true model is model #11, with W_1, W_2, W_3 included and W_4, W_5 excluded. The posterior model probability for model #11 is 1, and the estimated $\gamma_1, \gamma_2, \gamma_3$ are close to the true values of 0.5, 0.25 and 0.25 for $\gamma_1, \gamma_2, \gamma_3$ respectively.

```
Models           logm   Prob    rho     W1     W2      W3     W4      W5
Model 1     -7181.422  0.000  0.526  0.699  0.301   0.000  0.000   0.000
Model 2     -7022.144  0.000  0.552  0.670  0.000   0.330  0.000   0.000
Model 3     -7420.807  0.000  0.388  0.948  0.000   0.000  0.052   0.000
Model 4     -7419.463  0.000  0.378  0.974  0.000   0.000  0.000   0.026
Model 5     -7595.233  0.000  0.341  0.000  0.520   0.480  0.000   0.000
Model 6     -7874.670  0.000  0.195  0.000  0.883   0.000  0.117   0.000
Model 7     -7875.016  0.000  0.184  0.000  0.928   0.000  0.000   0.072
Model 8     -7807.486  0.000  0.184  0.000  0.000   0.888  0.112   0.000
Model 9     -7806.955  0.000  0.180  0.000  0.000   0.907  0.000   0.093
Model 10    -8066.402  0.000  0.001  0.000  0.000   0.000  0.483   0.517
Model 11    -6748.327  1.000  0.712  0.519  0.225   0.255  0.000   0.000
Model 12    -7190.746  0.000  0.542  0.677  0.290   0.000  0.033   0.000
Model 13    -7190.296  0.000  0.533  0.690  0.294   0.000  0.000   0.016
Model 14    -7032.028  0.000  0.568  0.651  0.000   0.321  0.028   0.000
Model 15    -7030.276  0.000  0.563  0.657  0.000   0.324  0.000   0.020
Model 16    -7429.027  0.000  0.396  0.925  0.000   0.000  0.052   0.024
Model 17    -7603.551  0.000  0.357  0.000  0.494   0.452  0.054   0.000
Model 18    -7604.383  0.000  0.351  0.000  0.498   0.461  0.000   0.041
Model 19    -7883.830  0.000  0.202  0.000  0.825   0.000  0.113   0.062
Model 20    -7815.839  0.000  0.194  0.000  0.000   0.815  0.104   0.081
Model 21    -6757.979  0.000  0.723  0.510  0.221   0.249  0.020   0.000
Model 22    -6756.867  0.000  0.719  0.514  0.222   0.251  0.000   0.012
Model 23    -7199.732  0.000  0.549  0.669  0.283   0.000  0.034   0.015
Model 24    -7040.179  0.000  0.576  0.640  0.000   0.314  0.027   0.019
Model 25    -7612.742  0.000  0.366  0.000  0.477   0.430  0.056   0.037
Model 26    -6766.710  0.000  0.731  0.504  0.218   0.245  0.020   0.012
BMA         -6748.329  1.000  0.712  0.519  0.225   0.255  0.000   0.000
highest     -6748.327  1.000  0.712  0.519  0.225   0.255  0.000   0.000
```

The function *prt_panel_bma()* prints out model averaged estimates based on information on the posterior model probabilities in the *results* structure returned by the *sdm_conv_panel_bma_g()* function.

The true effects estimates for this model were calculated as:

```
true effects estimates
variables     direct   indirect      total
x1            1.0323     4.8011     5.8333
x2           -1.0323    -4.8011    -5.8333
```

which compare favorably to the median scalar summary estimates printed out by the program. You should note that indirect effects will tend to be larger for a convex combination SDM model because these effects are *cumulative* in that they sum over impacts falling on all neighbors. In the case of multiple weight matrices there are of course more neighboring observations to cumulative over.

```
Bayesian Model Average of SDM convex panel W models
Dependent Variable  =             y
BMA Log-marginal    = -6748.3288
Nobs, T, Nvars      =    800,     5,     2
# weight matrices   =      5
ndraws,nomit        =  50000, 10000
total time          =  115.8330
thinning for draws  =      5
min and max rho     =   -1.0000,   1.0000
***************************************************************
      MCMC diagnostics ndraws = 8000
Variable         Mean     MC error        tau    Geweke
x1             0.9716   0.00013835   1.156375   0.998262
x2            -0.9774   0.00020397   1.089035   0.999123
W1*x1          0.4507   0.00070127   1.356955   0.998732
W1*x2         -0.5031   0.00049001   1.379928   0.996854
W2*x1         -0.8827   0.00054241   1.184388   0.996486
W2*x2          0.7961   0.00076037   1.188992   0.997342
W3*x1          1.0280   0.00122232   3.201924   0.990074
W3*x2         -0.7922   0.00131251   2.750326   0.991663
W4*x1          0.0000   0.00000009   1.102920   0.989073
W4*x2          0.0000   0.00000009   1.166437   0.932356
W5*x1          0.0000   0.00000000   0.985554   0.991486
W5*x2          0.0000   0.00000000   1.046096   0.836882
rho            0.7116   0.00085734   8.734568   0.995575
gamma1         0.5194   0.00080626  13.288309   0.991033
gamma2         0.2251   0.00053108   4.980542   0.995825
gamma3         0.2555   0.00051724   6.402590   0.985310
gamma4         0.0000   0.00000002   2.854552   0.988811
gamma5         0.0000   0.00000004   3.650927   0.949253
***************************************************************
      Posterior Estimates
Variable       lower 0.01       lower 0.05          median       upper 0.95       upper 0.99
x1               0.928234         0.938645        0.971625         1.004203         1.013569
x2              -1.019475        -1.009734       -0.977443        -0.944616        -0.934916
W1*x1            0.340863         0.367953        0.450942         0.533293         0.558391
W1*x2           -0.614080        -0.586021       -0.502890        -0.420267        -0.398130
W2*x1           -1.005194        -0.973131       -0.883522        -0.790005        -0.757864
W2*x2            0.675772         0.704823        0.795914         0.885963         0.920269
```

```
W3*x1         0.859241     0.898896     1.027325     1.161490     1.203838
W3*x2        -0.974041    -0.926610    -0.792708    -0.659007    -0.617905
W4*x1        -0.000013    -0.000008     0.000008     0.000023     0.000029
W4*x2        -0.000018    -0.000013     0.000003     0.000018     0.000023
W5*x1        -0.000000    -0.000000     0.000000     0.000000     0.000000
W5*x2        -0.000000    -0.000000     0.000000     0.000000     0.000000
rho           0.622174     0.647130     0.711394     0.778013     0.801297
gamma1        0.463626     0.473638     0.519230     0.567675     0.588230
gamma2        0.162602     0.179407     0.225294     0.268991     0.282600
gamma3        0.184265     0.203558     0.255996     0.303241     0.318844
gamma4        0.000000     0.000000     0.000001     0.000003     0.000004
gamma5        0.000000     0.000000     0.000002     0.000008     0.000010

Direct      lower 0.01   lower 0.05       median   upper 0.95   upper 0.99
x1            0.987176     0.998983     1.034609     1.069917     1.079679
x2           -1.083058    -1.070145    -1.035560    -0.999807    -0.991211

Indirect    lower 0.01   lower 0.05       median   upper 0.95   upper 0.99
x1            3.253174     3.485734     4.398994     5.779940     6.387412
x2           -6.112351    -5.409826    -4.079247    -3.197487    -2.960184

Total       lower 0.01   lower 0.05       median   upper 0.95   upper 0.99
x1            4.259431     4.503174     5.433708     6.830013     7.428478
x2           -7.166872    -6.461252    -5.114977    -4.219137    -3.967393
```

6.4 The SDEM convex combination BMA model

The same problem with calculating the log-marginal likelihood for SDM models that rely on a specification that includes spatial lags of the explanatory variables based on all $W-$matrices, e.g., $W_1X, W_2X, \ldots, W_MX$ exists for the SDEM model specification. This is because when considering all possible models involving two or more W_m, the number of explanatory variables in the matrix $\left(X \quad \sum_{m=1}^{M} W_mX \right)$ will change when we compare a model based on W_1, W_2 and (say) a model based on W_1, W_2, W_3. So, the size of the parameter space changes which will possibly create problems when calculating log-marginal likelihoods.

To be clear about this, the SDEM model specification is shown in (6.2), where we see that each matrix $W_i, i = 1, \ldots, m$ contributes explanatory variables to the model.

$$
\begin{aligned}
y &= X\beta + \sum_{m=1}^{M} W_mX\theta_m + u, \\
u &= \rho(I_T \otimes W_c)u + \varepsilon, \quad \varepsilon \sim N(0, \sigma^2 I_n), \\
W_c(\Gamma) &= \sum_{m=1}^{M} \gamma_m W_m, \quad 0 \le \gamma_m \le 1, \quad \sum_{m=1}^{M} \gamma_m = 1, \\
\Gamma &= (\gamma_1, \ldots, \gamma_M)'
\end{aligned}
\tag{6.2}
$$

The program file *sdem_conv_panel_bma_gd.m* generates an SDEM model based on the same three weight matrices and γ parameters used in the *sdm_conv_panel_bma_gd.m* program, where all three weight matrices appear in the model that generated the $u-$vector, $u = (I_{nt} - \rho(I_T \otimes W_c)^{-1}\varepsilon$.

```
\footnotesize
\begin{verbatim}
% file: sdem_conv_panel_bma_gd.m
% file: sdem_conv_panel_bma_gd.m
clear all;
sd = 221010;
rng(sd);
% estimate all possible models
% with two or more W-matrices
% nweights = 3, so we have 4 models with 2 or more W-matrices

n = 800;
t = 5;
xc = randn(n,1);  % generate 3 W-matrices
yc = randn(n,1);
W1 = make_neighborsw(xc,yc,5); % 5 nearest neighbors W-matrix

xc = randn(n,1);
yc = randn(n,1);
W2 = make_neighborsw(xc,yc,8); % 8 nearest neighbors W-matrix

xc = randn(n,1);
yc = randn(n,1);
W3 = make_neighborsw(xc,yc,12); % 12 nearest neighbors W-matrix

m=3;
gamma1 = 0.5; % assign gamma weights
gamma2 = 0.3;
gamma3 = 0.2;
gtrue = [gamma1
         gamma2
         gamma3];
%
Wc = gamma1*W1 + gamma2*W2 + gamma3*W3;

k=4; % 4 explanatory variables
x = [randn(n*t,k)];
beta = [1
        1
        1
        1];
theta1 = 0.5*beta;
theta2 = -0.75*beta;
theta3 =  beta;

bvec = [beta
        theta1
        theta2
        theta3];

% calculate true direct and indirect effects estimates
for ii=1:k
tmp2 = (eye(n)*beta(ii,1) + eye(n)*theta1(ii,1) + eye(n)*theta2(ii,1) + eye(n)*theta3(ii,1));
total_true(ii,1) = mean(sum(tmp2,2));
tmp1 = eye(n)*beta(ii,1); % + eye(n)*theta1(ii,1) + eye(n)*theta2(ii,1) + eye(n)*theta3(ii,1));
```

```
direct_true(ii,1) = mean(diag(tmp1));
indirect_true(ii,1) = total_true(ii,1) - direct_true(ii,1);
end

fprintf(1,'true effects estimates \n');
in.cnames = strvcat('direct','indirect','total');
in.rnames = strvcat('variables','x1','x2','x3','x4');

out = [direct_true indirect_true total_true];
mprint(out,in);

sige = 0.1;
rho = 0.7;
% generate True model
% add fixed effects to the DGP
tts = (1:n)*(1/n);
SFE = kron(ones(t,1),tts');
ttt = (1:t)*(1/t);
TFE = kron(ttt',ones(n,1));

Wmatrices = [kron(eye(t),W1) kron(eye(t),W2) kron(eye(t),W3)];

    Wx = [x kron(speye(t),W1)*x kron(speye(t),W2)*x kron(speye(t),W3)*x];
    Wxb = Wx*bvec;

u = (speye(n*t) - rho*kron(eye(t),Wc))\randn(n*t,1)*sqrt(sige);
y = (Wxb + SFE + TFE + u);

ndraw = 50000;
nomit = 10000;
prior.thin = 5; % retains only 8000 draws from 40,000
                % by skipping every 5
prior.model = 3; % fixed effects

% Estimation of Bayesian model averaging estimates using three matrices
result = sdem_conv_panel_bma_g(y,x,Wmatrices,n,t, ndraw, nomit, prior);

vnames = strvcat('y','x1','x2','x3','x4');
prt_panel_bma(result, vnames);
```

The model comparison results produced are shown below, where we see that the correct model that includes all three weight matrices was assigned a posterior model probability of 0.979. The model averaged estimates are also shown below, where we see reasonably accurate estimates of the parameters $\rho, \gamma_1, \gamma_2, \gamma_2$.

```
Models         logm        Prob         rho          W1          W2          W3
Model 1   -10023.317       0.000       0.458       0.608       0.392       0.000
Model 2    -9997.666       0.000       0.411       0.726       0.000       0.274
Model 3    -9977.985       0.021       0.245       0.000       0.679       0.321
Model 4    -9974.153       0.979       0.616       0.548       0.327       0.125
BMA        -9974.234       1.000       0.608       0.536       0.335       0.129
highest    -9974.153       0.979       0.616       0.548       0.327       0.125

true effects estimates
variables     direct   indirect      total
```

```
x1          1.0000     0.7500     1.7500
x2          1.0000     0.7500     1.7500
x3          1.0000     0.7500     1.7500
x4          1.0000     0.7500     1.7500

Homoscedastic model
Bayesian Model Average of SDEM convex panel W models
Dependent Variable  =              y
BMA Log-marginal    = -9974.2342
Nobs, T, Nvars      =    800,     5,     4
# weight matrices   =      3
ndraws,nomit        =  50000, 10000
total time          =  161.3620
thinning for draws  =      5
min and max rho     =   -0.9999,   0.9999
***************************************************************
      MCMC diagnostics ndraws = 8000
Variable        Mean     MC error        tau     Geweke
x1            0.9853   0.00022504   1.004816   0.997814
x2            1.0333   0.00020417   0.994830   0.998866
x3            1.0088   0.00017534   1.016852   0.999423
x4            0.9740   0.00017400   0.938384   0.999423
W1*x1         0.4652   0.00045495   1.076984   0.995458
W1*x2         0.5152   0.00030760   1.111984   0.998468
W1*x3         0.5594   0.00049780   0.939128   0.997243
W1*x4         0.4836   0.00054751   1.093726   0.992374
W2*x1        -0.7803   0.00066983   0.985454   0.993892
W2*x2        -0.5153   0.00043945   0.939390   0.994817
W2*x3        -0.6192   0.00059901   1.022682   0.998519
W2*x4        -0.7115   0.00049191   1.043275   0.997837
W3*x1         1.0034   0.00055230   1.042820   0.999454
W3*x2         0.9561   0.00080437   1.009224   0.996735
W3*x3         1.0246   0.00045155   0.972878   0.998794
W3*x4         1.1099   0.00071794   0.946314   0.995223
rho           0.6080   0.00111124   3.392614   0.994456
gamma1        0.5361   0.00113377   3.970410   0.995234
gamma2        0.3347   0.00057097   1.970481   0.998939
gamma3        0.1291   0.00122740   2.823362   0.977557

***************************************************************
      Posterior Estimates
Variable       lower 0.01       lower 0.05          median       upper 0.95       upper 0.99
x1               0.942293         0.952161        0.985239         1.018210         1.028388
x2               0.991552         1.000678        1.033235         1.065720         1.076841
x3               0.964561         0.975548        1.008720         1.041563         1.052834
x4               0.931465         0.942597        0.973762         1.005693         1.015140
W1*x1            0.357538         0.384642        0.464632         0.549679         0.573325
W1*x2            0.403827         0.429505        0.514954         0.601093         0.629131
W1*x3            0.448239         0.476471        0.560158         0.640300         0.664612
W1*x4            0.374416         0.401041        0.483241         0.565190         0.592329
W2*x1           -0.916447        -0.881708       -0.780011        -0.681305        -0.649786
W2*x2           -0.642273        -0.611359       -0.515527        -0.418846        -0.388712
W2*x3           -0.745018        -0.714689       -0.619009        -0.523291        -0.492582
W2*x4           -0.835513        -0.804803       -0.712059        -0.616457        -0.585307
W3*x1            0.861013         0.894764        1.003695         1.112237         1.144471
```

W3*x2	0.807308	0.843748	0.956751	1.069055	1.104187
W3*x3	0.874877	0.911599	1.024550	1.139337	1.176726
W3*x4	0.966624	0.996073	1.110079	1.220521	1.254294
rho	0.492539	0.518216	0.607334	0.703954	0.730337
gamma1	0.438323	0.458438	0.534497	0.621278	0.652758
gamma2	0.237669	0.260935	0.335295	0.406716	0.432320
gamma3	0.015489	0.032844	0.130661	0.220747	0.243215

Direct	lower 0.01	lower 0.05	median	upper 0.95	upper 0.99
x1	0.942293	0.952161	0.985239	1.018210	1.028388
x2	0.991552	1.000678	1.033235	1.065720	1.076841
x3	0.964561	0.975548	1.008720	1.041563	1.052834
x4	0.931465	0.942597	0.973762	1.005693	1.015140

Indirect	lower 0.01	lower 0.05	median	upper 0.95	upper 0.99
x1	0.461838	0.519241	0.688557	0.856804	0.915428
x2	0.727509	0.785450	0.954804	1.129741	1.177594
x3	0.741739	0.795038	0.962302	1.140789	1.199065
x4	0.660778	0.709281	0.882102	1.052307	1.099936

Total	lower 0.01	lower 0.05	median	upper 0.95	upper 0.99
x1	1.432077	1.491676	1.674058	1.853248	1.916554
x2	1.745574	1.806735	1.988892	2.173314	2.219503
x3	1.736706	1.791623	1.971849	2.161266	2.219183
x4	1.624207	1.671993	1.855439	2.036215	2.086876

We note that the direct effects in this model are simply the estimates for the coefficients on the matrix X of explanatory variables, while the indirect effects are constructed from the sum coefficients from *all* the explanatory variables matrix $W_c X$. These reflect what LeSage and Pace (2014) label *local spillovers* to immediately neighboring observations.

The lower 0.01, 0.05 and upper 0.95, 0.99 intervals are produced from the retained MCMC draws for these parameters. The posterior distribution for the total effects is constructed from the sum of the MCMC draws for the two sets of parameters, along with the credible intervals. Median estimates are presented because these would be better estimates of the central tendency of the posterior distribution in cases where the distribution is not symmetric. Of course, median estimates were also printed for the case of the SAR and SDM BMA model estimates.

6.5 Chapter summary

Model comparison of SAR, SDM and SDEM model specifications that produce estimates for all possible models involving two or more of the candidate weight matrices can be carried out. The basis for model comparison is the Monte Carlo integration estimate of the log-marginal likelihood produced during MCMC estimation of these models. Functions to explore which of the proposed weight matrices are most consistent with the sample data can exploit multi-processor computing through the use of the MATLAB parallel computing toolbox. If your computer does not have multiple cores, or the parallel computing toolbox is not installed, the function will proceed to loop over all possible models involving two or more weight matrices and produce MCMC estimates for each of the models, along with estimated log-marginal likelihoods.

A parsimonious set of weight matrices is advisable because inclusion of irrelevant weight matrices lead to estimates for the convex combination parameters near the zero boundary of the parameter space for the γ parameters. Many small γ_i parameters will subtract weight from the true γ_i parameters, possibly leading to erroneous inferences regarding the relative importance of the different types of connectivity being explored.

A point to note is that the example code used to demonstrate the convex combination models generated alternative weight matrices based on *unique* latitude-longitude coordinates and differing numbers of nearest neighbors. This was done for convenience, to mimic a situation where different weight matrices reflecting different types of connectivity are a modeling issue of interest. I am not suggesting one should use convex combinations of weight matrices based on different number of nearest neighbors in these models.

In terms of realistic examples of alternative types of weight matrices, Sheng and LeSage (2021) in a model that explores Chinese city-level knowledge production consider weights based on 1) spatial proximity, (W_S) 2) language and cultural similarity (W_L), 3) industry similarity (W_I), and 4) flows of recent college graduates (W_C). These all serve as different ways to view connectivity between a network of cities. Estimates for the parameters γ showed that γ showed γ parameters for: $W_S) = 0.3615, W_L) = 0.0695, W_I = 0.1790, W_C) = 0.3882$, so flows of recent college graduates was the most important aspect of city connections.

Fischer and LeSage (2020) construct weight matrices for a set of countries in a model of trade flows that reflect 1)spatial proximity, 2) membership in trade organizations, 3) historical colonial ties, 4) common currency, and 5) common language. Again, these serve as different ways to view connectivity between countries involved in country-level trade of commodities.

Debarsy and LeSage (2021) consider a cross-sectional hedonic house price regression where they explore alternative definitions of *comparable homes*. They construct alternative matrices W_i based on a set of weights constructed from a set of twenty nearest neighboring homes (within two miles) that had the same number of bedrooms (W_{beds}), same number of full plus half baths (W_{baths}) and same categorical age variable (W_{age}). Estimates for the parameters γ showed γ parameters for: $W_{beds}) = 0.196, W_{baths}) = 0.375, W_{age}) = 0.429$, so house age was the most important aspect of comparable homes.

Finally, users should be aware of potential problems with model comparison for the case of SDM and SDEM model specifications. The problem that arises is that the dimension of the parameter space changes for models based on different sets of $W-$matrices in these models because the set of explanatory variables depends on the number of weight matrices used in each model.

Of course the standard aspects of spatial regression models will exert an influence on how well these model comparison methods work. For example, things like the strength of dependence, the and the correlation/similarity of weight matrices used, the signal-to-noise or model fit, and the size of the weight matrices will all influence performance of these methods.

6.6 Chapter references

Debarsy, N. and LeSage, J.P. 2021. Bayesian model averaging for spatial autoregressive models based on convex combinations of different types of connectivity matrices, *Journal of Business & Economic Statistics*

Fischer, M.M. and J.P. LeSage (2020) Network dependence in multi-indexed data on international trade flows, *Journal of Spatial Econometrics*, Volume 1, 4 (2020).
https://doi.org/10.1007/s43071-020-00005-w

Koop, G. (2003). *Bayesian Econometrics*, West Sussex, England: John Wiley & Sons.

LeSage, J.P. (2020). Fast MCMC estimation of multiple W-matrix spatial regression models and Metropolis Hastings-Monte Carlo log marginal likelihoods *Journal of Geographical Systems*, 22:1, 47-75.

LeSage, J.P. and R.K. Pace (2009) *Introduction to Spatial Econometrics*, Taylor & Francis, CRC Press.

LeSage, J. P. and R. K. Pace (2014). Interpreting Spatial Econometric Models, *Handbook of Regional Science*, M. M. Fischer and Peter Nijkamp (Eds.) Springer, Berlin 2014, pp. 1535-1552.

Sheng, Y. and J.P. LeSage (2021) A spatial regression methodology for exploring the role of regional connectivity in knowledge production: Evidence from Chinese regions, (forthcoming) *Papers in Regional Science*, https://doi.org/10.1111/pirs.12601.

Chapter 7

Using the functions to produce estimates for cross-sectional models

7.1 Cross-sectional spatial regression models

Users may wish to produce estimates for single cross-sectional data models rather than panel data models, especially in the case of the convex combination models that can be efficiently estimated using the toolbox functions.[1]

Using the functions to produce cross-sectional model estimates simply requires that a data set based on one time period (T=1) is used as input to the functions, and that the option for no fixed effects is used (prior.model=0).

While the demonstrations that follow focus on convex combination of weight matrix models, the other functions such as *sar_panel_FE_g*, *sdm_panel_FE_g*, *sem_panel_FE_g*, *sdem_panel_FE_g*, also work this way to produce cross-sectional model estimates.

7.2 The SAR convex combination and BMA cross-sectional models

The file: *sar_conv_cross_section_gd.m* reads latitude-longitude coordinates from an ArcView Shapefile and generates a convex combination of weight matrices model based on 3,111 US counties in the lower 48 States (and the District of Columbia, D.C.). The true DGP uses two weight matrices, one based on contiguity calculated using the *xy2cont()* function from my Spatial Econometrics toolbox that uses Delaunay triangles to determine contiguity relationships and the other based on six nearest neighbors constructed using the *make_neighborsw()* function from my Spatial Econometrics toolbox. A third weight matrix is fed to the estimation function that reflects an inverse distance matrix with cut-off at six nearest neighbors. By this, I mean that non-zero weights based on inverse distance are assigned to the six nearest neighbors and zero weights to all other elements in the matrix. This matrix would be close to the six nearest neighbor weight matrix which equally weights the six neighbors. These two weight matrices should also be close to the contiguity matrix which has an average of around six nearest neighbors.

[1] At this point in time, there is to my knowledge no other software for estimating convex combinations of weight matrix models.

Note that the third weight matrix is not used in the DGP, because the value of $\gamma_3 = 0$ was used to generate the cross-sectional $y-$vector.

```
% sar_conv_cross_section_gd demo file
clear all;
rng(10203444);

% read an Arcview shape file for 3,111 US counties
map_results = shape_read('../demo_data/uscounties_projected');
latt = map_results.data(:,3);
long = map_results.data(:,4);
n = length(latt);
t = 1;
[j,Wcont,j] = xy2cont(latt,long); % Delaunay contiguity W
W6 = make_neighborsw(latt,long,6);% 6 nearest neighbors W
Wdist = distance(latt,long) + eye(n);
% create inverse distance with a 6 neighbor cut-off W
Wcut = (ones(n,n)./Wdist).*W6;
Wdist = normw(Wcut);

rho = 0.6;
k = 2;
x = randn(n*t,k);
beta = ones(k,1);
sige = 1;
evec = randn(n*t,1)*sqrt(sige);

gamma1 = 0.3;
gamma2 = 0.7;
gamma3 = 0.0;

Wc = gamma1*kron(eye(t),Wcont) + gamma2*kron(eye(t),W6) + gamma3*kron(eye(t),Wdist);

y = (speye(n*t) - rho*Wc)\(ones(n,1)*2 + x*beta + evec);

ndraw = 20000;
nomit = 10000;
prior.model = 0;
prior.thin = 5;

Wmatrices = [Wcont W6 Wdist];

result1 = sar_conv_panel_g(y,[ones(n,1) x],Wmatrices,n,t,ndraw,nomit,prior);
vnames = strvcat('y','constant','x1','x2');
prt_panel(result1,vnames);
```

Estimation results are shown below where we see a great deal of skew in the posterior distribution of the estimate for $\hat{\gamma}_2$, with a mean of 0.5676 and a median of 0.6178, as well as a skewed distribution for the $\hat{\gamma}_3$ estimate that exhibits a distribution piled-up at the zero boundary of the (0,1) parameter space.

```
MCMC SAR convex combination W model with no fixed effects
Homoscedastic model
```

```
Bayesian spatial autoregressive convex W model
Dependent Variable  =        y
Log-marginal        = -5622.2305
Log-marginal MCerror=  0.039611
R-squared           =    0.7463
corr-squared        =    0.6715
mean of sige draws  =    1.0007
posterior mode sige =    0.9973
Nobs, Nvars         =   3111,     3
ndraws,nomit        =  20000, 10000
time for effects    =    0.5160
time for sampling   =    2.6930
time for Taylor     =    0.1670
thinning for draws  =        5
min and max rho     =   -1.0000,   1.0000
***************************************************************
      MCMC diagnostics ndraws = 2000
Variable        mode         mean     MC error        tau     Geweke
constant       1.9331       1.9484   0.00182409   1.289902   0.994080
x1             0.9898       0.9903   0.00043771   0.998687   0.999671
x2             0.9792       0.9801   0.00044626   1.163389   0.998519
rho            0.6134       0.6106   0.00038228   1.293866   0.992488
gamma1         0.3794       0.3725   0.00223666   4.024133   0.985859
gamma2         0.6178       0.5676   0.00259521   4.278851   0.986122
gamma3         0.0028       0.0599   0.00132381   2.672744   0.770904

***************************************************************
      Posterior Estimates
Variable      Coefficient  Asymptot t-stat    z-probability
constant         1.948448        24.841571         0.000000
x1               0.990280        55.449442         0.000000
x2               0.980116        54.006478         0.000000
rho              0.610634        40.616887         0.000000
gamma1           0.372546         5.160702         0.000000
gamma2           0.567569         6.960202         0.000000
gamma3           0.059885         1.306025         0.191544

Direct        Coefficient           t-stat           t-prob         lower 05         upper 95
x1               1.076564        54.608866         0.000000         1.039465         1.115406
x2               1.065518        52.715727         0.000000         1.027675         1.105793

Indirect      Coefficient           t-stat           t-prob         lower 05         upper 95
x1               1.470380        15.646313         0.000000         1.297383         1.666272
x2               1.455349        15.452771         0.000000         1.283032         1.654145

Total         Coefficient           t-stat           t-prob         lower 05         upper 95
x1               2.546944        24.328856         0.000000         2.353534         2.762381
x2               2.520867        23.841765         0.000000         2.323256         2.743628
```

We can produce BMA estimates in and effort to see if this produces estimates based on a model involving the two true $W-$matrices, by adding the lines:

```
result2 = sar_conv_panel_bma_g(y,[ones(n,1) x],Wmatrices,n,t,ndraw,nomit,prior);
vnames = strvcat('y','constant','x1','x2');
prt_panel_bma(result2,vnames);
```

which produces the estimation results shown below. Here, we see that the correct model is identified and assigned a posterior model probability of 0.965. This results slightly improved estimates based on the median values for the parameters γ.

```
Models          logm         Prob          rho           W1           W2           W3
Model 1    -5618.752        0.965        0.612        0.383        0.617        0.000
Model 2    -5640.656        0.000        0.592        0.690        0.000        0.310
Model 3    -5631.743        0.000        0.598        0.000        0.906        0.094
Model 4    -5622.077        0.035        0.611        0.368        0.576        0.056
BMA        -5618.868        1.000        0.612        0.382        0.616        0.002
highest    -5618.752        0.965        0.612        0.383        0.617        0.000


Bayesian Model Average of SAR convex panel W models
Dependent Variable  =        y
BMA Log-marginal    = -5618.8676
Nobs, T, Nvars      =   3111,     1,     3
# weight matrices   =      3
ndraws,nomit        =  20000, 10000
total time          =    5.0220
thinning for draws  =      5
min and max rho     =   -1.0000,   1.0000
***************************************************************
      MCMC diagnostics ndraws = 2000
Variable         Mean     MC error        tau    Geweke
constant       1.9397   0.00236492   1.220240   0.989782
x1             0.9901   0.00045662   1.204218   0.997054
x2             0.9796   0.00040330   1.031225   0.999694
rho            0.6120   0.00031928   1.211217   0.996532
gamma1         0.3823   0.00169637   1.509398   0.976083
gamma2         0.6158   0.00171041   1.499187   0.985605
gamma3         0.0019   0.00005720   2.534404   0.912257

***************************************************************
      Posterior Estimates
Variable        lower 0.01       lower 0.05           median       upper 0.95       upper 0.99
constant          1.757028         1.793435         1.941922         2.084837         2.133922
x1                0.943668         0.956613         0.989968         1.024132         1.032266
x2                0.937715         0.947388         0.979659         1.013014         1.023374
rho               0.574540         0.583786         0.611457         0.641031         0.647184
gamma1            0.215297         0.247207         0.383349         0.525205         0.556871
gamma2            0.441130         0.473776         0.615220         0.750325         0.783086
gamma3            0.000011         0.000043         0.001626         0.005502         0.007022

Direct          lower 0.01       lower 0.05           median       upper 0.95       upper 0.99
x1                1.024322         1.038122         1.076049         1.112949         1.123668
x2                1.019116         1.028695         1.064758         1.101495         1.113259

Indirect        lower 0.01       lower 0.05           median       upper 0.95       upper 0.99
x1                1.263941         1.308300         1.474729         1.669011         1.732967
x2                1.253704         1.299563         1.458298         1.643806         1.697759

Total           lower 0.01       lower 0.05           median       upper 0.95       upper 0.99
x1                2.311060         2.365411         2.551156         2.763910         2.820185
x2                2.295104         2.343285         2.524089         2.726483         2.793879
```

7.3 The SDM convex combination and BMA cross-sectional models

The program *sdm_conv_cross_section_gd.m* estimates a convex combination of weights SDM model, where the true DGP reflects a model based on two matrices W_1, W_2, but the estimation function is called with three weight matrices, W_1, W_2, W_3.

```
% sdm_conv_cross_section_gd demo file
% demonstrate using the function to produce
% estimates for a cross-sectional model
clear all;
rng(30203040);

n = 3000;
t = 1;
rho = 0.2;
k = 2;
x = [randn(n*t,k)];
beta = ones(k,1);
theta1 = -0.75*ones(k,1);
theta2 = -0.25*ones(k,1);

bvec = [beta
        theta1
        theta2];

sige = 1;
evec = randn(n*t,1)*sqrt(sige);
latt = rand(n,1);
long = rand(n,1);
W1 = make_neighborsw(latt,long,2);
latt = rand(n,1); % A different set of latt-long
long = rand(n,1); % coordinates
W2 = make_neighborsw(latt,long,6);
latt = rand(n,1); % A different set of latt-long
long = rand(n,1); % coordinates
W3 = make_neighborsw(latt,long,12);

gamma1 = 0.2;
gamma2 = 0.8;

    Wx = [x kron(speye(t),W1)*x kron(speye(t),W2)*x];
    Wxb = Wx*bvec;

% calculate true direct and indirect effects estimates
Wc = gamma1*W1 + gamma2*W2;
direct_true = zeros(k,1);
indirect_true = zeros(k,1);
total_true = zeros(k,1);

B = (speye(n) - rho*Wc)\(speye(n));

for ii=1:k
tmp2 = B*(eye(n)*beta(ii,1) + W1*theta1(ii,1) + W2*theta2(ii,1));
```

```
total_true(ii,1) = mean(sum(tmp2,2));
tmp1 = B*(eye(n)*beta(ii,1) + W1*theta1(ii,1) + W2*theta2(ii,1));
direct_true(ii,1) = mean(diag(tmp1));
indirect_true(ii,1) = total_true(ii,1) - direct_true(ii,1);
end

fprintf(1,'true effects estimates \n');
in.cnames = strvcat('direct','indirect','total');
in.rnames = strvcat('variables','x1','x2');

out = [direct_true indirect_true total_true];
mprint(out,in);

y = (speye(n*t) - rho*Wc)\(ones(n,1)*2 + Wxb + evec);

prior.model = 0;
prior.plt_flag = 0;
ndraw = 20000;
nomit = 10000;

Wmatrices = [W1 W2 W3];

result1 = sdm_conv_panel_g(y,[ones(n,1) x],Wmatrices,n,t,ndraw,nomit,prior);
vnames = strvcat('y','constant','x1','x2');
prt_panel(result1,vnames);
```

Results are shown below, where we see relatively poor estimates for the parameters γ_1, γ_2, which should not be surprising given the value of $\rho = 0.2$ used in the DGP. It might at first seem somewhat surprising that the effects estimates which are relatively close to their true values (also shown in the results) given the poor estimates for the parameters $\rho, \gamma_1, \gamma_2, \gamma_3$. However, given the relatively low value of the true dependence parameter $\rho = 0.2$, and the estimated value $\hat{\rho} = 0.2482$, the effects estimates are largely determined by the coefficients on the X, W_1X, W_2X, W_3X variables in the model. In other words, this model is very similar to an SLX specification.

```
true effects estimates
variables     direct   indirect      total
x1            0.9874    -0.9874    -0.0000
x2            0.9874    -0.9874    -0.0000

MCMC SDM convex combination W model with no fixed effects
Homoscedastic model
Bayesian spatial Durbin convex W model
Dependent Variable  =         y
Log-marginal        = -5335.9298
Log-marginal MCerror=  0.065554
R-squared           =    0.7147
corr-squared        =    0.7105
mean of sige draws  =    0.9966
posterior mode sige =    0.9924
Nobs, Nvars         =   3000,     3
ndraws,nomit        =  20000, 10000
time for effects    =    7.0490
time for sampling   =    5.9830
time for Taylor     =    0.4108
```

```
thinning for draws  =         1
min and max rho     =   -1.0000,   1.0000
*************************************************************
      MCMC diagnostics ndraws = 10000
Variable            mode         mean     MC error         tau     Geweke
constant          1.8624       1.8574   0.01651375  434.294715   0.913443
x1                0.9920       0.9918   0.00026725    1.191781   0.996815
x2                1.0046       1.0048   0.00037733    1.242339   0.997506
W1*x1            -0.7794      -0.7792   0.00089370    2.531784   0.989761
W1*x2            -0.7566      -0.7564   0.00092480    2.587249   0.988743
W2*x1            -0.2475      -0.2443   0.00231010    2.885154   0.886454
W2*x2            -0.2385      -0.2355   0.00230205    3.191452   0.909513
W3*x1             0.0546       0.0479   0.00356454    3.302248   0.532680
W3*x2            -0.0897      -0.0950   0.00314574    2.905943   0.641913
rho               0.2482       0.2498   0.00671526  444.586556   0.699424
gamma1            0.1851       0.1835   0.00171032   13.772461   0.966141
gamma2            0.6410       0.6284   0.00808468  114.281798   0.882245
gamma3            0.1739       0.1881   0.00878559  132.609891   0.506388

*************************************************************
      Posterior Estimates
Variable         Coefficient  Asymptot t-stat    z-probability
constant            1.857375        19.936566         0.000000
x1                  0.991812        53.859400         0.000000
x2                  1.004778        54.491715         0.000000
W1*x1              -0.779239       -28.094364         0.000000
W1*x2              -0.756449       -27.215144         0.000000
W2*x1              -0.244315        -4.967944         0.000001
W2*x2              -0.235519        -4.873242         0.000001
W3*x1               0.047945         0.681287         0.495690
W3*x2              -0.094985        -1.408240         0.159060
rho                 0.249773         6.740645         0.000000
gamma1              0.183504         3.454304         0.000552
gamma2              0.628394         7.693433         0.000000
gamma3              0.188102         2.160102         0.030765

Direct           Coefficient           t-stat           t-prob         lower 05         upper 95
x1                  0.977757        52.658323         0.000000         0.941174         1.013852
x2                  0.988646        53.338964         0.000000         0.952272         1.025626
Indirect         Coefficient           t-stat           t-prob         lower 05         upper 95
x1                 -0.958369        -8.032291         0.000000        -1.201212        -0.731836
x2                 -1.100502        -9.514728         0.000000        -1.327357        -0.881924
Total            Coefficient           t-stat           t-prob         lower 05         upper 95
x1                  0.019388         0.156483         0.875663        -0.230158         0.256592
x2                 -0.111856        -0.934677         0.350030        -0.346791         0.114498
```

As in the case of the SAR model, we can add a few lines to produce BMA estimates.

```
result2 = sdm_conv_panel_bma_g(y,[ones(n,1) x],Wmatrices,n,t,ndraw,nomit,prior);
vnames = strvcat('y','constant','x1','x2');
prt_panel_bma(result2,vnames);
```

The results from this set of estimates are shown below, where we see much improved estimates for a model based on only W_1, W_2 matrices, ρ, γ_1, γ_2, which receives a posterior model probability of 0.998.

```
Models          logm         Prob          rho           W1           W2           W3
Model 1    -5329.430        0.998        0.206        0.226        0.774        0.000
Model 2    -5349.521        0.000        0.085        0.541        0.000        0.459
Model 3    -5983.225        0.000        0.187        0.000        0.840        0.160
Model 4    -5335.867        0.002        0.244        0.186        0.638        0.175
BMA        -5329.440        1.000        0.206        0.226        0.773        0.000
highest    -5329.430        0.998        0.206        0.226        0.774        0.000

Homoscedastic model
Bayesian Model Average of SDM convex panel W models
Dependent Variable  =        y
BMA Log-marginal    = -5329.4400
Nobs, T, Nvars      =   3000,     1,     3
# weight matrices   =      3
ndraws,nomit        =  20000, 10000
total time          =    8.4780
thinning for draws  =      2
min and max rho     =   -1.0000,   1.0000
***************************************************************
      MCMC diagnostics ndraws = 5000
Variable            Mean     MC error        tau     Geweke
constant          1.9664   0.00491096  40.318775   0.996454
x1                0.9913   0.00038998   1.161556   0.999859
x2                1.0058   0.00035795   1.180285   0.999659
W1*x1            -0.7795   0.00065056   1.585565   0.999849
W1*x2            -0.7581   0.00068298   1.932482   0.999936
W2*x1            -0.2485   0.00123513   2.202255   0.996464
W2*x2            -0.2395   0.00162006   2.263689   0.994270
W3*x1             0.0001   0.00000321   1.811206   0.914751
W3*x2            -0.0001   0.00000282   1.799071   0.977510
rho               0.2059   0.00193668  42.666305   0.988097
gamma1            0.2264   0.00226960   5.636609   0.995330
gamma2            0.7733   0.00226995   5.638253   0.998667
gamma3            0.0003   0.00000824  20.638171   0.896720
***************************************************************
      Posterior Estimates
Variable           lower 0.01       lower 0.05           median       upper 0.95       upper 0.99
constant             1.818488         1.847909         1.966308         2.080917         2.116320
x1                   0.945653         0.955195         0.991310         1.026794         1.038008
x2                   0.956873         0.969641         1.005815         1.042176         1.052883
W1*x1               -0.852162        -0.834403        -0.779274        -0.726466        -0.709779
W1*x2               -0.827438        -0.811957        -0.758008        -0.703988        -0.686089
W2*x1               -0.371629        -0.342312        -0.248010        -0.157757        -0.129490
W2*x2               -0.361433        -0.334528        -0.238578        -0.146906        -0.111082
W3*x1               -0.000189        -0.000125         0.000086         0.000294         0.000368
W3*x2               -0.000411        -0.000355        -0.000147         0.000057         0.000117
rho                  0.149856         0.161590         0.206267         0.252015         0.261938
gamma1               0.056869         0.103827         0.226048         0.342163         0.379878
gamma2               0.619763         0.657678         0.773776         0.895896         0.942733
gamma3               0.000007         0.000037         0.000282         0.000549         0.000625

Direct             lower 0.01       lower 0.05           median       upper 0.95       upper 0.99
x1                   0.927136         0.936152         0.972173         1.008049         1.019158
x2                   0.939057         0.951339         0.987463         1.023560         1.034000
Indirect           lower 0.01       lower 0.05           median       upper 0.95       upper 0.99
```

```
x1            -1.203983     -1.156876     -1.017934     -0.886184     -0.848869
x2            -1.154024     -1.112556     -0.975760     -0.852821     -0.808540
Total         lower 0.01    lower 0.05       median    upper 0.95    upper 0.99
x1            -0.245506     -0.193537     -0.046692      0.094402      0.134537
x2            -0.176206     -0.134113      0.011738      0.145534      0.185107
```

7.4 The SDEM convex combination and BMA cross-sectional models

We can take the same approach with the SDEM model, estimating cross-sectional versions of the convex combination model as well as BMA model estimates. The program below illustrates this.

```
% sdem BMA program for sdem_conv_cross_section_gd.m
clear all;
sd = 221010;
rng(sd);
% estimate all possible models
% with two or more W-matrices
% nweights = 3, so we have 4 models with 2 or more W-matrices
n = 3000;
t = 1;
m=3;

xc = randn(n,1);  % generate 5 W-matrices
yc = randn(n,1);
W1 = make_neighborsw(xc,yc,5); % 5 nearest neighbors W-matrix
xc = randn(n,1);
yc = randn(n,1);
W2 = make_neighborsw(xc,yc,8); % 8 nearest neighbors W-matrix
xc = randn(n,1);
yc = randn(n,1);
W3 = make_neighborsw(xc,yc,12); % 12 nearest neighbors W-matrix

gamma1 = 0.5; % assign gamma weights
gamma2 = 0.3;
gamma3 = 0.2;
gtrue = [gamma1
         gamma2
         gamma3];
%
Wc = gamma1*W1 + gamma2*W2 + gamma3*W3;

k=2; % 4 explanatory variables
x = [randn(n*t,k)];
beta = [1
        1];
theta1 = 0.5*beta;
theta2 = 1*beta;
theta3 = -1*beta;
bvec = [beta
        theta1
        theta2
        theta3];
```

```
sige = 1;
rho = 0.6;
% calculate true direct and indirect effects estimates
for ii=1:k
tmp2 = (eye(n)*beta(ii,1) + eye(n)*theta1(ii,1) + eye(n)*theta2(ii,1) + eye(n)*theta3(ii,1));
total_true(ii,1) = mean(sum(tmp2,2));
tmp1 = eye(n)*beta(ii,1); % + eye(n)*theta1(ii,1) + eye(n)*theta2(ii,1) + eye(n)*theta3(ii,1));
direct_true(ii,1) = mean(diag(tmp1));
indirect_true(ii,1) = total_true(ii,1) - direct_true(ii,1);
end

fprintf(1,'true effects estimates \n');
in.cnames = strvcat('direct','indirect','total');
in.rnames = strvcat('variables','x1','x2');
out = [direct_true indirect_true total_true];
mprint(out,in);

    Wx = [x kron(speye(t),W1)*x kron(speye(t),W2)*x kron(speye(t),W3)*x];
    Wxb = Wx*bvec;

u = (speye(n*t) - rho*kron(eye(t),Wc))\randn(n*t,1)*sqrt(sige);
y = (ones(n,1)*2.0 + Wxb + u);

ndraw = 20000;
nomit = 10000;
prior.thin = 5; % retains only 2000 draws from 10,000
                % by skipping every 5
prior.model = 0; % no fixed effects

Wmatrices = [W1 W2 W3];
result = sdem_conv_panel_g(y,[ones(n,1) x],Wmatrices,n,t, ndraw, nomit, prior);
vnames = strvcat('y','constant','x1','x2');
prt_panel(result, vnames);
% Estimation of Bayesian model averaging estimates using three matrices
result = sdem_conv_panel_bma_g(y,[ones(n,1) x],Wmatrices,n,t, ndraw, nomit, prior);
vnames = strvcat('y','constant','x1','x2');
prt_panel_bma(result, vnames);
```

The results based on estimation of a convex combination of weights model where the true three $W-$matrices used in the DGP were input to the function are shown below, where we see reasonably accurate estimates close to the true values used in the DGP.

In contrast, the BMA estimates assign a posterior model probability of 1.0 to the wrong model, one based on only the matrices W_2, W_3. This of course produces inaccurate estimates of the parameters $\rho, \gamma_1, \gamma_2, \gamma_3$ as well as the effects estimates. In this case the dependence parameter $\rho = 0.6$, so the effects estimates are dependent on more than the coefficients associated with the spatial lags of the $X-$variables.

```
true effects estimates
variables     direct   indirect      total
x1            1.0000     0.5000     1.5000
x2            1.0000     0.5000     1.5000

Homoscedastic model
Bayesian spatial Durbin error convex W model
```

```
Dependent Variable  =         y
Log-marginal      = -7445.2802
Log-marginal MCerror=  0.116020
R-squared          =    0.7112
Rbar-squared       =    0.7343
mean of sige draws  =    0.9712
Nobs, Nvars        =   3000,     3
ndraws,nomit       =  20000, 10000
total time in secs =   39.9470
time for sampling  =   39.0310
time for Taylor    =    0.9160
min and max lambda =   -0.9999,   0.9999
***************************************************************
      MCMC diagnostics ndraws = 2000
Variable            mean     MC error        tau     Geweke
constant          1.9934   0.00150954   1.032081   0.996334
x1                1.0225   0.00045870   0.964969   0.998248
x2                0.9787   0.00042646   1.119898   0.998319
W1*x1             0.4879   0.00078021   0.871439   0.996827
W1*x2             0.5113   0.00133095   1.149980   0.999483
W2*x1             1.0333   0.00111344   0.957232   0.999819
W2*x2             0.9509   0.00144369   0.950376   0.993863
W3*x1            -1.0323   0.00097965   0.994400   0.994737
W3*x2            -0.9649   0.00121375   0.927276   0.997211
rho               0.6819   0.00274678   3.118095   0.988509
gamma1            0.4639   0.00203274   4.032013   0.976764
gamma2            0.2933   0.00104449   2.467300   0.992146
gamma3            0.2428   0.00189861   2.412504   0.965619

***************************************************************
      Posterior Estimates
Variable         Coefficient  Asymptot t-stat    z-probability
constant            1.993421        33.917911         0.000000
x1                  1.022491        56.038712         0.000000
x2                  0.978653        53.682788         0.000000
W1*x1               0.487917        10.848841         0.000000
W1*x2               0.511323        10.995969         0.000000
W2*x1               1.033260        19.292047         0.000000
W2*x2               0.950861        17.311233         0.000000
W3*x1              -1.032293       -15.962755         0.000000
W3*x2              -0.964917       -14.724689         0.000000
rho                 0.681930        11.492120         0.000000
gamma1              0.463943        11.092780         0.000000
gamma2              0.293258         6.874957         0.000000
gamma3              0.242799         4.938592         0.000001

Direct           Coefficient           t-stat           t-prob         lower 05         upper 95
x1                  1.022491        56.038712         0.000000         0.985574         1.057591
x2                  0.978653        53.682788         0.000000         0.942956         1.013901
Indirect         Coefficient           t-stat           t-prob         lower 05         upper 95
x1                  0.488884         5.064511         0.000000         0.301415         0.675888
x2                  0.497268         5.064531         0.000000         0.299708         0.692178
Total            Coefficient           t-stat           t-prob         lower 05         upper 95
x1                  1.511374        14.812347         0.000000         1.314706         1.706869
x2                  1.475921        14.100572         0.000000         1.269920         1.686485
```

```
Models          logm         Prob          rho           W1           W2           W3
Model 1    -7461.032        0.000        0.468        0.612        0.388        0.000
Model 2    -7469.300        0.000        0.370        0.714        0.000        0.286
Model 3    -7436.292        1.000        0.313        0.000        0.550        0.450
Model 4    -7445.399        0.000        0.684        0.462        0.294        0.243
BMA        -7436.293        1.000        0.313        0.000        0.550        0.450
highest    -7436.292        1.000        0.313        0.000        0.550        0.450

Homoscedastic model
Bayesian Model Average of SDEM convex panel W models
Dependent Variable  =         y
BMA Log-marginal    = -7436.2928
Nobs, T, Nvars      =   3000,     1,     3
# weight matrices   =      3
ndraws,nomit        =  20000, 10000
total time          =   28.8960
thinning for draws  =      5
min and max rho     =   -0.9999,   0.9999
***************************************************************
      MCMC diagnostics ndraws = 2000
Variable            Mean     MC error        tau     Geweke
constant          1.9863   0.00052136   0.918918   0.999039
x1                1.0271   0.00039588   1.088777   0.998452
x2                0.9736   0.00046977   1.014482   0.998314
W1*x1             1.0392   0.00101793   1.108424   0.997416
W1*x2             0.9108   0.00180323   0.896122   0.992192
W2*x1            -1.0357   0.00132094   0.939376   0.996284
W2*x2            -0.9417   0.00200353   0.978509   0.999950
W3*x1            -0.0001   0.00000018   0.954244   0.992756
W3*x2            -0.0001   0.00000013   1.012949   0.993963
rho               0.3135   0.00095639   1.440658   0.983289
gamma1            0.0001   0.00000015   2.631086   0.999018
gamma2            0.5503   0.00197368   1.388264   0.976676
gamma3            0.4497   0.00197369   1.388264   0.971792
***************************************************************
      Posterior Estimates
Variable          lower 0.01       lower 0.05           median       upper 0.95       upper 0.99
constant            1.916642         1.932144         1.985589         2.039936         2.059201
x1                  0.981706         0.990455         1.027250         1.064685         1.078291
x2                  0.927675         0.936454         0.973069         1.011841         1.022865
W1*x1               0.882831         0.924350         1.039692         1.154403         1.190353
W1*x2               0.755578         0.794217         0.909973         1.031568         1.059909
W2*x1              -1.214938        -1.180269        -1.033404        -0.899821        -0.852588
W2*x2              -1.124961        -1.081899        -0.942483        -0.806260        -0.768191
W3*x1              -0.000132        -0.000129        -0.000115        -0.000101        -0.000096
W3*x2              -0.000126        -0.000122        -0.000107        -0.000093        -0.000089
rho                 0.173386         0.209316         0.313837         0.420711         0.457156
gamma1              0.000041         0.000043         0.000051         0.000061         0.000064
gamma2              0.334879         0.378011         0.544206         0.740906         0.828112
gamma3              0.171833         0.259039         0.455740         0.621940         0.665067

Direct            lower 0.01       lower 0.05           median       upper 0.95       upper 0.99
x1                  0.981706         0.990455         1.027250         1.064685         1.078291
x2                  0.927675         0.936454         0.973069         1.011841         1.022865
```

```
Indirect        lower 0.01    lower 0.05      median    upper 0.95    upper 0.99
x1               -0.243731     -0.179971    0.005699      0.177660      0.242617
x2               -0.261795     -0.213218   -0.030876      0.147407      0.216588
Total           lower 0.01    lower 0.05      median    upper 0.95    upper 0.99
x1                0.774660      0.836118    1.032646      1.214378      1.278957
x2                0.702305      0.755944    0.942497      1.132585      1.195718
```

7.5 Model comparison for cross-sectional models

There is a function *lmarginal_cross_section()* that can be used to carry out model comparison of cross-sectional models. The following program demonstrates use of this function, which produces log-marginal likelihoods and posterior model probabilities for the SLX, SDM and SDEM models. Recall that LeSage (2014, 2015) argues that these are the only models that need be considered. The program generates $y-$vectors for SLX, SDM and SDEM models and calls the function *lmarginal_cross_section()* to calculate log-marginal likelihoods and posterior model probabilities for all three models.

```
% file: cross_section_demo.m
% simulate SLX, SDM, SDEM models then calculate log-marginals
clear all;
load schools.dat;
% col 1 = school district ID
% col 2 = longitude centroid for the district
% col 3 = latitude centroid for the district

long = schools(:,2);
latt = schools(:,3);
W = make_neighborsw(latt,long,6);
[n,junk] = size(W);
N = n;
rng(86573);
sigx = 1;
x = randn(N,1)*sqrt(sigx);
x1 = x;
x = randn(N,1)*sqrt(sigx);
x2 = x;
x = randn(N,1)*sqrt(sigx);
x3 = x;
% generate y
xo = [x1 x2 x3];
tmp = [ -0.5 1 0.5];
beta = tmp';
tmp = [-1 0.5 1]; % make sure these don't equal -rho*beta
gamm = tmp';
sige = 1;
alpha = 10;
% ============================================
xmat = [ones(N,1) xo W*xo]; % model includes W*x-variables
    rho = 0.5;
    lam = 0.4;
F = speye(N) - rho*W;
G = speye(N) - lam*W;
```

```
xmat = [ones(N,1) xo W*xo]; % model includes W*x-variables

beta_gamma = [alpha
    beta
    gamm];

eterm = randn(N,1)*sqrt(sige);
tmp = [xmat*beta_gamma];
y_slx = tmp + eterm;

result1 = lmarginal_cross_section(y_slx,xo,W);
fprintf(1,'true model is SLX \n');
fprintf(1,'time taken is: %16.4f seconds \n',result1.time);
in.cnames = strvcat('log-marginal','model probs');
in.rnames = strvcat('model','slx','sdm','sdem');
in.width = 10000;
in.fmt = '%10.4f';
out = [result1.lmarginal result1.probs];
mprint(out,in);

y_sdm = F\(tmp + eterm);

result2 = lmarginal_cross_section(y_sdm,xo,W);
fprintf(1,'true model is SDM \n');
fprintf(1,'time taken is: %16.4f seconds \n',result2.time);
fprintf('rho = %10.4f  \n',0.5);
in.cnames = strvcat('log-marginal','model probs');
in.rnames = strvcat('model','slx','sdm','sdem');
in.width = 10000;
in.fmt = '%10.4f';
out = [result2.lmarginal result2.probs];
mprint(out,in);

y_sdem = tmp + G\eterm;

result3 = lmarginal_cross_section(y_sdem,xo,W);
fprintf(1,'true model is SDEM \n');
fprintf(1,'time taken is: %16.4f seconds \n',result3.time);
fprintf('lambda = %10.4f  \n',0.4);
in.cnames = strvcat('log-marginal','model probs');
in.rnames = strvcat('model','slx','sdm','sdem');
in.width = 10000;
in.fmt = '%10.4f';
out = [result3.lmarginal result3.probs];
mprint(out,in);
```

The results are shown below, where we see that the correct models were identified and those having the highest posterior model probability. Recall that in the case of the SLX model we are not going to see a model probability close to one as in the case of the SDM and SDEM model specifications. The number of observations is 605, which reflects the number of school districts in the state of Ohio (at some point in time, since these change over time).

```
true model is SLX
time taken is:           0.0950 seconds
model log-marginal  model probs
```

```
slx     -1088.6072       0.4447
sdm     -1089.1336       0.2627
sdem    -1089.0255       0.2927

true model is SDM
time taken is:           0.0630 seconds
rho =     0.5000
model log-marginal  model probs
slx     -1177.4566       0.0000
sdm     -1104.4638       0.9998
sdem    -1113.2265       0.0002

true model is SDEM
time taken is:           0.0630 seconds
lambda =     0.4000
model log-marginal  model probs
slx     -1124.6364       0.0000
sdm     -1103.0695       0.0057
sdem    -1097.9087       0.9943
```

There are input options, one is *info.lflag = 0,1;*, where a value of 0 uses a sparse Cholesky approach to calculate the exact log-determinant term that appears in the log-marginal likelihood, and a value of 1 uses the (faster) Barry and Pace (1999) Monte Carlo approximation. Another is *info.eig = 0,1;*, where a value of 0 calculates the exact minimum eigenvalue and a value of 1 sets this to a value of -1. The default values for both of these are set to 1 to produce the fastest results. The results based on setting info.eig = 1, and info.lflag = 1 are shown below, where we come to the same conclusions regarding the models most consistent with the sample data. Note that the time taken is dramatically different, 5.6400/0.0950 = 59 times as long when using the slower options. This is not a problem for small sample sizes, but these differences would grow for larger sample sizes. A positive point is that the approximations should be produce smaller differences in the log-marginal likelihoods for larger samples, since they will be more accurate.

```
true model is SLX
time taken is:           5.6400 seconds
model log-marginal  model probs
slx     -1088.6072       0.3686
sdm     -1088.8197       0.2980
sdem    -1088.7077       0.3334

true model is SDM
time taken is:           5.2300 seconds
rho =     0.5000
model log-marginal  model probs
slx     -1177.4566       0.0000
sdm     -1101.6389       0.9996
sdem    -1109.5475       0.0004

true model is SDEM
time taken is:           5.1860 seconds
lambda =     0.4000
model log-marginal  model probs
slx     -1124.6364       0.0000
sdm     -1101.6033       0.0034
sdem    -1095.9084       0.9966
```

7.5.1 Comparing weight matrices for cross-sectional models

We can also use the *log_marginal_cross_section()* function to compare both model specifications and weight matrices. The following file examines the SLX, SDM and SDEM model specifications while looping over weight matrices constructed based on 4 to 16 nearest neighbors.

```
% file: cross_section_demo3.m, demonstrates comparison of weight matrices
clear all;
rng(86573);
load uscounties.data;
% a matrix now exist named uscounties
% the matrix contains 11 columns of county-level data
% col 1  FIPS
% col 2  LATITUDE
% col 3  LONGITUDE
% col 4  POP1990
% col 5  1987_PCI (per capita income)
% col 6  1988_PCI
% col 7  1989_PCI
% col 8  1990_PCI
% col 9 1991_PCI
% col 10 1992_PCI
% col 11 1993_PCI
[n,k] = size(uscounties); % find the size of the matrix
pci1987 = uscounties(:,5);  % extract the 5th column from the data matrix
pci1993 = uscounties(:,11); % creates an n x 1 column vector
pop1990 = uscounties(:,4);
% calculate growth rate of per capita income over the 1987 to 1993 period
pci_growth = log(pci1993) - log(pci1987);
% make these annualized growth rates
pci_growth = pci_growth/7;
% do a growth regression
% which involves regressing the growth rate on the (logged) initial level
xmatrix = [log(pci1987) log(pop1990)];
% run SDM model
latt = uscounties(:,2); % extract latt-long coordinates
long = uscounties(:,3);
W(1).matrix = zeros(n,n);
neigh = [];
for ii=4:16
    W(ii-3).matrix = make_neighborsw(latt,long,ii);
    neigh = [neigh
             ii];
end

lmarginal = [];
nweights = size(neigh,1);
for i=1:nweights
res(i).result = lmarginal_cross_section(pci_growth,xmatrix,W(i).matrix);
lmarginal = [lmarginal
             res(i).result.lmarginal]; % a 3 x 1 vector for each loop i
end

probs = model_probs(lmarginal);
probs_matrix = reshape(probs,nweights,3);
```

```
in.fmt = '%16.4f';
in.cnames = strvcat('slx','sdm','sdem');
in.rnames = strvcat('#neighbors',num2str(neigh));
mprint(probs_matrix,in);

% run the best model
% 1st we need to figure out which model is best
% and how many neighbors to use
[nprob,nmax] = max(probs_matrix,[],1);
% nmax is an index of the highest SLX, SDM, SDEM model probs

[prob,mmax] = max(nprob);
% mmax is an index to the best model SLX,SDM,SDEM

if mmax == 1
    model = 'slx';
elseif mmax == 2
    model = 'sdm';
elseif mmax == 3
    model = 'sdem';
end

neighbors_index = nmax(mmax);

num_neighbors = neigh(nmax(mmax));
% # of neighbors for weight matrix with highest prob

W = make_neighborsw(latt,long,num_neighbors);
ndraw = 2500;
nomit = 500;
prior.novi_flag = 1;
prior.model = 0;
T = 1;

switch model

    case{'slx'}

result = slx_panel_FE_g(pci_growth,[ones(n,1) xmatrix],W,T,ndraw,nomit,prior);

    case{'sdm'}

result = sdm_panel_FE_g(pci_growth,[ones(n,1) xmatrix],W,T,ndraw,nomit,prior);

    case{'sdem'}

result = sdem_panel_FE_g(pci_growth,[ones(n,1) xmatrix],W,T,ndraw,nomit,prior);

    otherwise
        % do nothing
       disp('Unknown model');

end
% print out estimation results
```

```
vnames = strvcat('income growth','constant','log(pci0)','log(pop0)');
prt_panel(result,vnames);
```

Results from running the program are shown below, where we see that only SDEM models had any noticeable posterior probability support. An SDEM model based on 10 nearest neighbors was the best model. Note that by constructing a *single* vector of log-marginal likelihoods that were used in the call to the function *model_probs()*, we are comparing SLX, SDM, SDEM model specifications based on 13 different weight matrices, for a total of $3 \times 13 = 39$ different models. Therefore, 39×1 is the dimension of the vector labeled *lmarginal* in the program.

The program takes advantage of the MATLAB *switch()* function to execute a call to the appropriate estimation function based on the model comparison probabilities, which in our case is the SDEM specification.

```
#neighbors              slx              sdm              sdem
 4                   0.0000           0.0000           0.0001
 5                   0.0000           0.0000           0.0000
 6                   0.0000           0.0000           0.0000
 7                   0.0000           0.0000           0.0068
 8                   0.0000           0.0000           0.0000
 9                   0.0000           0.0000           0.0007
10                   0.0000           0.0000           0.9399
11                   0.0000           0.0000           0.0000
12                   0.0000           0.0000           0.0000
13                   0.0000           0.0000           0.0032
14                   0.0000           0.0000           0.0000
15                   0.0000           0.0000           0.0000
16                   0.0000           0.0000           0.0491

Homoscedastic model
MCMC SDEM model with no fixed effects
Dependent Variable =    income growth
R-squared           =    0.2639
corr-squared        =    0.2672
sigma^2             =    0.0002
Nobs,Nvar,#FE       =   3111,     3,     0
ndraw,nomit         =   2500,   500
rvalue              =      0
min and max rho     =   -1.0000,   1.0000
total time in secs  =    4.0010
time for eigs       =    0.0670
time for MCMC draws =    3.8890
Pace and Barry, 1999 MC lndet approximation used
order for MC appr  =     50
iter  for MC appr  =     30
***************************************************************
Variable             Coefficient  Asymptot t-stat    z-probability
constant                0.428901        13.815419         0.000000
log(pci0)              -0.047247       -25.321679         0.000000
log(pop0)               0.001887         6.831178         0.000000
W*log(pci0)             0.009678         2.214313         0.026807
W*log(pop0)             0.003815         5.166071         0.000000
rho                     0.724763        39.317353         0.000000

Direct               Coefficient           t-stat           t-prob         lower 05         upper 95
```

```
log(pci0)          -0.047247    -25.321679    0.000000    -0.050833    -0.043573
log(pop0)           0.001887      6.831178    0.000000     0.001353     0.002446
Indirect        Coefficient        t-stat      t-prob     lower 05     upper 95
log(pci0)           0.009678      2.214313    0.026880     0.000903     0.017780
log(pop0)           0.003815      5.166071    0.000000     0.002400     0.005249
Total           Coefficient        t-stat      t-prob     lower 05     upper 95
log(pci0)          -0.037569     -8.610049    0.000000    -0.046153    -0.029148
log(pop0)           0.005701      7.574325    0.000000     0.004234     0.007173
```

7.6 Chapter summary

This chapter demonstrated that user can produce estimates for cross-sectional models using the *Panel_g* Toolbox functions. This is probably most useful in the context of the new convex combinations of spatial weight matrix models, for which code is now available for both cross-sectional as well as fixed effects panel data models.

7.7 Chapter references

Barry, R. and R.K. Pace (1999) A Monte Carlo estimator of the log-determinant of large sparse matrices, *Linear Algebra and its Applications*, 289, 41-51.

LeSage James P. (2014). Spatial econometric panel data model specification: A Bayesian approach, *Spatial Statistics* Volume 9, Issue C, 122-145.

LeSage, James P. (2015). Software for Bayesian cross section and panel spatial model comparison, *Journal of Geographical Systems*, 17, 297-310.