Short note on MCMC routines

Chapter 1

Intro

In order to explore possible improvements in estimating parameters via Metropolis type algorithms, we have considered two different generalizations of the classical Metropolis algorithm.

On one side we have implemented an Adaptive Markov Chain Monte Carlo scheme (see [7, 2, 5, 6, 3]) .

On the other hand we have considered a simple code which implements the replica exchange algortithm. This idea goes back to the work of Swendsen and Wang (see [8]) and it has been mostly applied to physical problems (see [4]) and to molecular dynamics (see [1]).

We refer to previous report for some details about theoretical aspects.

The attempt of using the two approaches together should be further investigated.

Chapter 2

Adaptive algos

In the following we discuss the routines we have written in order to perform an Adaptive Markov Chain Monte Carlo algorithm. The main ideas have been borrowed from the work of Laine ([3]) which is based on the paper of Gelman et alii ([2]). We have focused our code on the adaptive part without pay attention to the Delayed Rejection scheme.

2.1 Routines

We briefly outline the routines related to MCMC adaptation.

AdaMetropolis:

It is the main routine. Its input are

- TargetFun: the objective function. It can be defined by invoking a fuction which describe the model underlying the estimation (see options_.mh_modelfun) or diectly.
- xparam: initial paramenter values. They will be also used in the CreateParams.
- vv: Initial (prior) covariance. By default is given by:

  ```
  if isempty(vv)
  vv=(xparam*0.05).^2
  vv(vv==0)=1
  ```

```
vv=diag(vv);
end
```

- mh_bounds: parameters bounds.
- options_: Matlab structure
- varargin: it contains data

It uses CreateParams, CreateOptions and AdaMCMC.

It returns MarkovChain, options_, params

AdaMCMC

It computes the adaptive Metropolis algorithm. Its input are:

data: same use as varargin{2} in Dynare metropolis.m

params: parameters values

options_: see

CreateOptions

It defines the default value for options_ strucure. As the main parameters are concerned we have

- options_.mh_npars: number of parameters;
- options_.mh_replic: number of simulation: default =1000;
- options_.mh_adapint: length of interval for adaptatin: default=10;
- options_.mh_burnintime: burning time period: default = 10;
- options_.mh_incov: initial covariance;
- options_.mh_updatesigma: update error covariance: default =0 no update;
- options_.mh_noadaptind: list of indeces not to be adapted: default=[];
- options_.mh_adascale: scale parameter for adaptation: default =[];
- options_.mh_sigma2: initial value for the error variance default = 1;

- options_.mh_ssfun: will be the target function in Dynare metropolis;
- options_.mh_modelfun: model function, it can be used to define the target function;
- options_.mh_priortype: prior type, default 1 = Gaussian;
- options_.mh_parammu: prior parameter mean: default 0;
- options_.mh_paramsig: prior parameter variance: default Inf.

CreateParameters

It fills the parameters values.

Its inputs are:

- params: initial parameters values;
- mh_bounds: lower and upper parameters bounds;
- options_: Matlab structure with a list of options (see CreateOptions).

gammar

It computes the random deviate from gamma distribution.

Chapter 3

Replica exchange algo

3.1 Routines

RepExMCMC

Computes the replica exchange MCMC. Its inputs are:

- PP: paramters initial values (it depends on the number of replicas we uses);
- options_: same as adaptive algorithm (see)
- varargin: it contains data

CreateOptions

Same as the adaptive case with three extra options.

- options_.mh_nblck: number of replicas: default=2
- options_.mh_exchangeint: length of interval for exchange: default=10
- options_.mh_exchangeind: indeces to be exchanged: default all

3.2 Example

To test the performance of the routines, both for adaptive and replica exchange case, we consider the following simple example.

options_mh.modelfun = @(x,theta) theta(1)*x./(theta(2)+x);

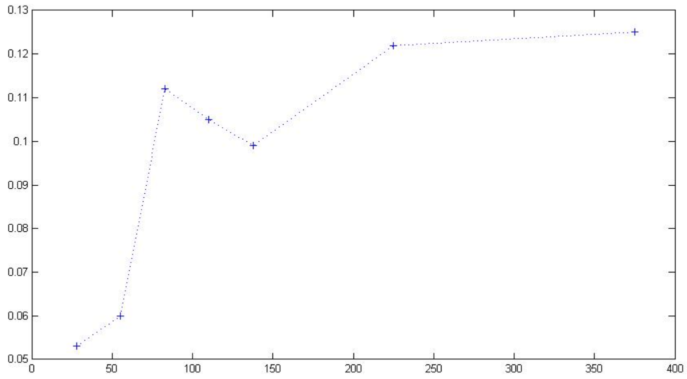
$$f(x,\theta) = \frac{\theta_1 x}{\theta_2 + x}$$

ssfun = residual sum of squares.

The data set is given by:

data.xdata = [28 55 83 110 138 225 375]’;

data.ydata = [0.053 0.060 0.112 0.105 0.099 0.122 0.125]’;



The adaptive scheme is performed on the following setting:

TargetFun=ssfun;

params=[0.1454 49.0530];

ssmin = 8.1677e-004;

mse = ssmin/(n-p) ;

$$vv = \begin{bmatrix} 0.0002 & 0.2501 \\ 0.2501 & 320.9218 \end{bmatrix}$$

mh_bounds=[0 Inf; 0,Inf];

options_.mh_sigma2 = mse;

options_.mh_replic = 500;

options_.mh_npars = 2;

options_.mh_updatesigma =1;

options_.mh_adaptint = 25;

options_.mh_noadaptind =[];

options_.mh_priortype=1.

The replica exchange (on two replicas) starts from:

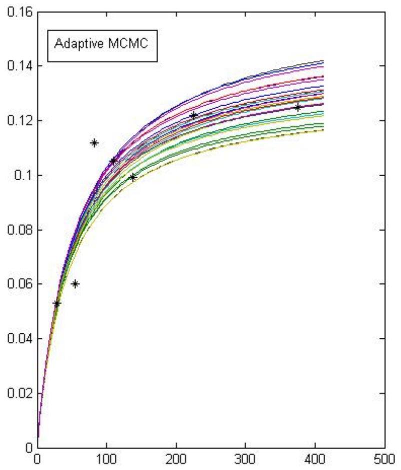
$$PP = \begin{bmatrix} 0.1454 & 0.145 \\ 49.0530 & 52 \end{bmatrix};$$

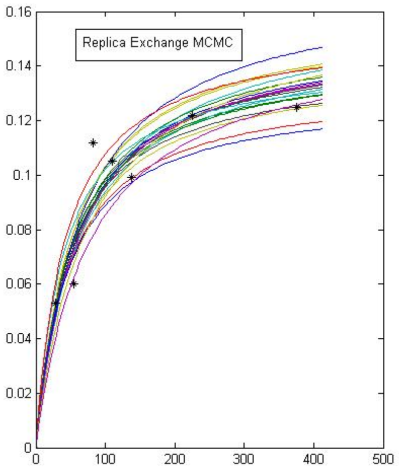
with the following options_

options_.mh_exchangeint=50;

options_.mh_replic=2.

The results are shown in the following





The initial values, length of adaptive interval, length of the exchange interval need to be further invesigated.

Bibliography

[1] M. Eleftheriou, A. Rayshubski, J.W. Pitera, B.G. Fitch, R. Zhou, and R.S. Germain. Parallel implementation of the replica exchange molecular dynamics algorithmon blue gene/l. *IEEE*, 2:1–8, 2006.

[2] A. Gelman, O. Roberts, G., and R. Gilks, W. Efficient metropolis jumping rules. *BAYESIAN STATISTICS*, 5:599–607, 1996.

[3] M. Laine. *ADAPTIVE MCMC METHODS WITH APPLICATIONS IN ENVIRONMENTAL AND GEOPHYSICAL MODELS*. PhD thesis, 2008.

[4] Jon Machta, Mark Newman, and Lincoln Chayes. New algorithm and results for the three-dimensional random field ising model. *Physical Review E*, 62:8782, 2000.

[5] O. Roberts, G. and S. Rosenthal, J. Examples of adaptive mcmc. Technical report, 2008.

[6] Jeffrey S. Rosenthal. Amcmc: An r interface for adaptive mcmc. Working paper, Department of Statistics, University of Toronto, 2007.

[7] C.A. Sims. Adaptive metropolis-hastings sampling, or monte carlo kernel estimation, 1998.

[8] H. Swendsen, R. and S. Wang, J. Replica monte carlo simulation of spin glasses. *Physical Review Letters*, 57:2607–2609, 1986.