Downloaded from UvA-DARE, the institutional repository of the University of Amsterdam (UvA)
http://hdl.handle.net/11245/2.96211

File ID	uvapub:96211
Filename	474fulltext.pdf
Version	unknown

SOURCE (OR PART OF THE FOLLOWING SOURCE):

Type	pre-print - working paper
Title	Stationary preserving and efficiency increasing probability mass transfers made possible
Author	A. Mira
Faculty	FEB: Amsterdam School of Economics Research Institute (ASE-RI)
Year	2003

FULL BIBLIOGRAPHIC DETAILS:
http://hdl.handle.net/11245/1.346860

Copyright

It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use.

UvA-DARE is a service provided by the library of the University of Amsterdam (http://dare.uva.nl)
(pagedate: 2014-08-19)

UvA ECONOMETRICS

Discussion Paper: 2003/06

Stationary preserving and efficiency increasing probability mass transfers made possible

Antonietta Mira and Pieter Omtzigt

www.fee.uva.nl/ke/UvA-Econometrics

Department of Quantitative Economics
Faculty of Economics and Econometrics
Universiteit van Amsterdam
Roetersstraat 11
1018 WB AMSTERDAM
The Netherlands

UvA UNIVERSITEIT VAN AMSTERDAM

Stationary preserving and efficiency increasing probability mass transfers made possible

Antonietta Mira, *Pieter Omtzigt †

October 28, 2003

Abstract

We develop an efficient computational algorithm that produces efficient Markov chain Monte Carlo (MCMC) transition matrices. The first level of efficiency is measured in terms of the number of operations needed to produce the resulting matrix. The second level of efficiency is evaluated in terms of the asymptotic variance of the resulting MCMC estimates. The idea is then extended from finite to general state spaces.

Keywords: Asymptotic efficiency of MCMC estimates, Metropolis-Hastings, Stationarity preserving and efficiency increasing probability mass transfers.

1 Motivation

The idea of stationarity preserving and efficiency increasing probability mass transfers (SPT) developed in [2] is quite appealing from a theoretical point of view in that it produces an efficient Markov chain Monte Carlo (MCMC) transition matrix. Efficiency is measured by a first-degree approximation to the asymptotic variance of the resulting MCMC estimates.

Unfortunately, as presented by Mira in [2], the creation of the first-degree optimal matrix via SPT in not optimized from a computational point of

*University of Insubria, Department of Economics, Via Ravasi 2, 21100 Varese, Italy

†Universiteit van Amsterdam, Faculty of Economics and Econometrics, Roetersstraat 11, 1018 WB Amsterdam

view. As a result, the exact evaluation of the normalizing constant of the distribution of interest π and of the mean of a function f relative to π via brute force calculation, might become less computationally intensive (or at least competitive from a computational point of view) than the construction of the first-degree optimal transition matrix itself.

The aim of this paper is to fill in this gap and to provide a recipe to construct the first-degree optimal matrix in order n steps, n being the dimension of the state space of interest.

The manuscript is organized as follows. In Section 2 the idea of SPT is reviewed and the drawbacks behind its implementation are highlighted. In Section 3 we give the recipe that makes SPT of actual practical interest. In Section 4 an approximation of the first-degree optimal matrix is provided via an algorithm of order $\sqrt{n}$. Each row of the resulting approximate matrix can be computed separately and can be used as a clever proposal distribution in a standard Metropolis-Hastings algorithm ([4]).

Finally, the idea behind the construction of the first-degree optimal matrix is extended from finite to general state spaces in Section 5.

2 Stationarity preserving and efficiency increasing probability mass transfers: SPT

Let $\pi(x), x \in \mathcal{X}$ be a discrete valued distribution of interest, known up to a normalizing constant $c = \sum_{i=1}^{n} \pi(x_i)$. In some application, such as image analysis and signal processing, the large value of n prohibits brute force evaluation of c. The researcher is often interested in computing expectations of various functionals of π: $\mu(f) = \frac{1}{c}\sum_{i}^{n} f(x_i)\pi(x_i)$. Again we are faced with the problem of evaluating a sum which requires order n operations.

To avoid the evaluation of these sums, Monte Carlo (MC) or Markov chain Monte Carlo simulation can be used. MC simulation assumes that we can generate independent and identically distributed (iid) random variables, $\mathbf{x} = x_1, x_2, \cdots x_N$, from π. In MCMC an ergodic Markov chain having π as its unique stationary distribution is set up and run for N steps to produce random variables that have π as their limiting distributions. In either case (MC or MCMC), under week regularity conditions, the low of large numbers

and the central limit theorem ensure that the estimate obtained as

$$\hat{\mu}(f, P, N) = \frac{1}{N} \sum_{i=1}^{N} f(x_i)$$

is asymptotically unbiased, consistent and normaly distributed. The asymptotic variance of the estimator is

$$v(f, P) = f\Pi[2l_P^{-1} - I - A]f' = (f, [2l_P^{-1} - I - A]f), \tag{1}$$

where $A = \mathbf{1}'\pi$ is the limiting matrix, and $l_P = (I - P + A)$ is the Laplacian. The asymptotic variance can be computed (or estimated) and used to compare alternative MC or MCMC schemes in terms of their asymptotic efficiency (note that MC simulation can be regarded as an MCMC algorithm with A as the transition matrix).

Given a transition matrix P stationary with respect to π, $\pi P = \pi$, a Markov chain is simulated by choosing a starting value $X_0 = x_0$ from some initial distribution and then generating X_1 from the row of P corresponding to the value that X_0 has assumed and so on. We will refer to this row as the x_0-row of P: it is the conditional distribution of the position of the chain at time $t + 1$ given that at time t the chain has value x_0.

Starting from a transition matrix P that has π as its stationary distribution and a function of interest f monotone on the state space (this assumption requires at most order n operations to rearrange that state space), SPT transfers produce a new matrix, 1-opt-P, that is first-degree optimal in the sense that it produces estimates for $\mu(f)$ that have the smallest asymptotic variance up to a first-degree approximation.

A single SPT transfer performed on P works as follows: given integers $1 \leq i < j \leq n$ and $1 \leq k < l \leq n$ and a quantity $h > 0$, increase $p_{i,l}$ and $p_{j,k}$ by h and $h\pi_i/\pi_j$ respectively and decrease $p_{i,k}$ and $p_{j,l}$ by h and $h\pi_i/\pi_j$ respectively. Thus a single SPT is completely defined by the 4 indexes, i, j, k, l, and the amount by which to increase/decrease the appropriate entries of the matrix on which the transfer is performed. We will thus identify a SPT by $P(i, j, k, l, h)$. The quantity h must be chosen so that, after the mass transfer, in the resulting matrix all the entries are non-negative and less than one. If P is derived from Q via a sequence of stationarity-preserving and efficiency increasing transfers then (Mira, 2000), $f\Pi(P-Q)f' \leq 0$. Since

$$l_P^{-1} = I + \sum_{i=1}^{\infty} (P - A)^i$$

and $\lim_{i\to\infty}(P-A)^i$ equals, by construction, the $n \times n$ zero matrix, a first-degree approximation to $v(f,P)$ is given by

$$v(f,P) \approx v(f,A) + 2f\Pi(P-A)f'$$

where $v(f,A)$ is the theoretical MC variance of $\hat{\mu}(f,A,N)$ and $f\Pi(P-A)f' = (f,Pf)$ is the first-degree covariance if π is the distribution of the initial state of the Markov chain. Notice that, the smaller the eigenvalues of P are in absolute value, the better this first-degree approximation to the asymptotic variance is.

As a result of this first-degree approximation we have that if P is derived from Q via a sequence of SPT then P produces MCMC estimates that have a smaller asymptotic variance up to a first-degree approximation.

As showed in Mira 2001, the 1-opt-P matrix is *unique*. Notice that if the 1-opt-P matrix is *not irreducible* it cannot be used for MCMC purposes. Reducibility happens if any of the ratios π_i/π_j is a rational number. To circumvent this problem one can "switch on" some of the zero entries of a reducible extreme matrix by making them positive and thus connecting the parts of the state space that were separated in the original extreme matrix. Finally, as proved in the next Theorem, the 1-opt-P matrix is always *reversible* even if the matrix P on which we have performed the SPT transfers was, originally, only stationary with respect to π.

Theorem 1. *The first degree optimal transition matrix is always reversible.*

Proof. Finding the first degree optimal P matrix is equivalent to solve a constrained minimization problem. The function to be minimized, relative to P, is the lag one covariance, (f,Pf), and the minimization is taken over the set K, defined by the stationarity constraints, $\pi P = \pi$ and the stochasticity constraints i.e. the fact that the rows of P have to sum to one. We are thus minimizing a liner function over a convex set and the solution is unique. Let P^* indicate the self-adjoint of P i.e. the unique operator such that $(f,Pf) = (P^*f,f), \quad \forall f \in L_0^2(\pi)$ where $L_0^2(\pi)$ is the set of square integrable function with respect to π having zero mean. Let g be the specific function we have in mind and Q the first degree optimal matrix we have found by $\min_{P\in K}(g,Pg)$. Since $(g,Qg) = (Q^*g,g)$ and since the minimum is unique it follows that $Q = Q^*$ i.e. the first degree optimal matrix is reversible. □

A brute force algorithm takes n^2 operation to produce the 1-opt-P matrix since it has to cycle twice over all entries. In the next Section a more clever

algorithm is proposed that outputs the 1-opt-P matrix in order n steps. We note here that, if we stop the SPT at an intermediate stage (before the extreme 1-opt-P matrix is reached) because we have run out of time, we still obtain an improvement over the original π-stationary matrix (this intermediate matrix might not be reversible).

All the algorithms presented require an initial step in which the state space is re-ordered to make f monotone. This is an easy task is f is specified as an analytic function (which is the case in most applications) such as $f(x) = x$ (no reordering required) or $f(x) = x^2$ (states $\pm x$ need to be placed next to each other, note that more than one reordering is available in this case, depending on $-x$ being placed before or after $+x$, this redundancy is irrelevant for further analysis).

3 SPT made possible

The input of the brute force procedure outlined in the previous section are π and some P matrix in stationarity with respect to π. Since the identity matrix I is stationary with respect to any distribution we can always take $P = I$ to initialize the SPT mechanism, thus, in the sequel, we present an algorithm that only takes π as input and outputs the 1-opt-P matrix.

The idea is to start filling an $n \times n$ transition matrix from the South-West corner with the highest probability mass compatible with the following requirements: *stochastic matrix requirement* (the row sum has to be one and the entries have to be between zero and one), and *stationarity requirement* ($\pi P = \pi$). The stochastic matrix requirement imposes a constraint on the row sum while the stationarity requirement imposes a constraint on the weighted (by the π's entries) column sum. Each time we fill in an entry in the matrix we keep track of how much freedom is left in terms of the mass that we can place on the corresponding row and column. Three cases can occur.

- If no freedom is left on the row sum in the next iteration we have to move up one row, otherwise we move along the same row toward East.
- If no freedom is left on the column sum in the next iteration we have to move right one column, otherwise we move along the same column up-words (toward North).

- If no freedom is left in either the column and the row sum in the next iteration we have to move both right one column and up one row, this is equivalent to moving diagonally toward the North-East corner of the matrix.

The procedure is iterated until the North-East corner is reached and thus requires at most $n + (n - 1)$ steps since, at most, there are $n + (n - 1)$ non-zero entries in the final 1-opt-P matrix. In other words the longest path that takes us from the South-West to the North-East corner has $n + (n - 1)$ steps.

4 A clever proposal for Metropolis-Hastings algorithms

It would be of great help if we could produce a single row of the 1-opt-P matrix without having the fill in the entire matrix. Thanks to the Markov property, this is in fact sufficient to simulate a Markov chain having the required stationary distribution: if the chain is in position x at time t then the x-row of the 1-opt-P matrix is needed to sample the position of the chain at time $t + 1$. This is helpful because points in the state space that have very small probability under π are likely not visited by a simulated chain in a finite time simulation. If many such points exists we would then save a lot of computational time by evaluating only the rows of the 1-opt-P matrix associated with high π-probability regions. The required rows would be evaluated upon request (i.e. the first time the chain finds itself in a specific state) and stored.

Unfortunately the iterative procedure outline in Section 3 does not allow this. In order to fill in a row we need to know the content of all the rows below it. A partial solution to this problem is to rearrange the state space so that π is increasing (order n operations), fill in the rows with highest π probability up to a threshold (for example fill in only $\sqrt{n}$ rows or rows that get, under π, a probability greater than some threshold) and then rearrange the states (and the rows and columns of P accordingly) to make f monotone. The resulting matrix has some blank rows. If one of the blank rows is required the next position in the chain is sampled via a standard Metropolis-Hastings procedure (or by using the corresponding row of the original P matrix, if available).

As an alternative we could produce and approximation to the row of

the 1-opt-P matrix requested for the next move and use it as the proposal distribution in a standard Metropolis-Hastings algorithm. In other words, to determine the position of the chain at time $t+1$ we would generate a candidate move from the approximation of the row of interest and accept the move with the probability that preserves detailed balance (which, in turn, implies stationarity):

$$\alpha(x,y) = \min\left[1, \frac{\pi(y)q(y,x)}{\pi(x)q(x,y)}\right]$$

where $q(x,y)$ is the proposal distribution (in this setting the x-row of the approximating matrix).

The following remain open questions: How can we construct the approx-opt-P i.e. the matrix whose rows are "good" approximations of the rows in the 1-opt-P matrix? And can we produce such approximation on a row-by-row basis? One possible way to find an answer to the above questions would be to evaluate π on a (random or equally spaced) grid of $\sqrt{n}$ points and use these values to construct the approximation we need.

5 General state spaces

The natural question that occurs is: how does the construction of the first degree optimal matrix extend to *general state spaces*? The intuition tells us that there might exist a function, $\phi : \mathcal{X} \to \mathcal{X}$, that associates, to every point in the state space x, the point which has the highest negative correlation with x. To find this function we have to solve the following differential equation. Let $P(x_n \leq x) = \int_{-\infty}^{x} P_n(dx)$. Then we require that, in stationarity,

$$\int_{-\infty}^{x} P_{n+1}(dx) = \int_{\phi^{-1}(x)}^{\infty} P_n(dx).$$

The function $\phi(x) = F^{-1}[1 - F(x)]$ gives a solution to the above equation where F is the distribution function associated with π. This ϕ is related to the minimum of the Fréchet class i.e. the class of all the bivariate joint distributions with given marginals. If the marginals are both equal to π then the joint distribution that achieves the highest negative correlation among the marginals is

$$C(x,y) = \max\{F(x) + F(y) - 1, 0\}, \qquad \forall (x,y) \in \mathcal{X}^2.$$

This means that, for any other joint distribution H belonging to the same Fréchet class, we have

$$C(x,y) \leq H(x,y), \qquad \forall (x,y) \in \mathcal{X}^2.$$

The distribution function C assigns probability one to the points in the plane that lay on the graph of $\phi(x)$.

Just to give an example, if $\pi(x) = e^{-x}$ then $\phi(x) = -\log(1-e^{-x})$. On discrete state spaces if we define $F^{-1}(x) = \min\{y : F(y) > x\}$, then, for any i, the function $\phi(x_i)$ gives the set of points in the state space that have some chance of being reached from x_i in one step.

Since ϕ is a self-inverse function i.e. $\phi[\phi(x)] = x$, it is useless to apply ϕ twice in a MCMC simulation. In light of this, one possible strategy to improve MCMC performance, in terms of first degree efficiency, for every function f monotone on the state space, would be to run a regular MCMC sampler and then, once stationarity is reached, at every point in time, t, associate to x_t the corresponding $\phi(x_t)$ (this can be done by post-processing the chain and after having run convergence diagnostics) and then estimate μ by

$$\hat{\mu}_B = \frac{1}{2N}[\sum_{i=1}^{N} x_i + \sum_{i=1}^{N} \phi(x_i)]. \tag{2}$$

We call this strategy the *branching sampler*.

An alternative to the above construction is to build an *hybrid sampler* i.e. insert, within a regular MCMC algorithm, first degree optimal steps with some probability p.

Typically neither F nor F^{-1} is available when we resort to MCMC simulation to study a distribution function π or to approximate integrals with respect to π. Various options are available to solve this problem. A pilot run can be performed and the output used to approximate the distribution function of π and, consequently, to find an approximation to $\phi(x)$ for any given x. Or, as suggested by Chauveau and Vandekerkhove [1], we could run parallel chains, all having π as their limiting distribution, and then, as the simulation goes on, use some of those auxiliary chains to approximate $\phi(x)$ and then disregard the chain used, so that the Markovian property is retained.

Once an MCMC sample from π has been obtained, $x_1, \cdots, x_K$ we approximate $\phi(x)$ by ordering the sample increasingly $x_{(1)} \leq x_{(2)} \leq \cdots \leq x_{(K)}$,

finding the position of x within the ordered sample, say $x_{(i)} \leq x < x_{(i+1)}$, and then approximating $\phi(x)$ by $x_{(K-i)}$.

So far we have considered updating x all at once. Suppose now that x is a vector with coordinates or group of coordinates that we want to update simultaneously, giving the following partition of the original vector: $x = (x^1, \cdots, x^d)$. When updating coordinate x^j we would consider the full conditional distribution function of x^j given the rest, in place of F above.

What we propose here has the *over-relaxation* suggested by Neal [3] as a special case. Neal considers the setting of a Gibbs sampler where we update one (real-valued) coordinate at a time. Suppose we are updating coordinate j i.e. we need to sample a value from $\pi(x^j|x_{-j})$, the corresponding full conditional. Neal suggests sampling K values from the full conditional, ordering them, together with the current value of x^j. Without loss of generality assume that x^j has position j in the ordering, we would then update that coordinate with the value $x_{(K-j)}$ where $x_{(j)}$ denotes the j-th order statistics. This is exactly what we would suggest, in this special setting, if we were building an approximation of the full conditional distribution function at each step of the simulation. An alternative would be to use pilot runs of length between K and $K \times N$ to build approximations to the full conditional distribution functions once and for all at the beginning of the simulation, thus avoiding the extra cost needed to sample the K values at each point in simulation time.

6 Conclusions

We propose an efficient way to construct transition matrices for MCMC samplers where efficiency has a double meaning: computational efficiency and statistical efficiency. The idea is extended from finite to general state spaces and connections with existing ways of improving the statistical efficiency of MCMC samplers are discussed.

References

[1] D. Chauveau and P. Vandekerkhove. Improving convergence of the hastings-metropolis algorithm with a learning proposal. *Scandinavian Journal of Statistics*, To appear.

[2] A. Mira. Ordering and improving the performance of Monte Carlo Markov chains. *Statistical Science*, 16:340–350, 2002.

[3] R. M. Neal. Suppressing random walks in Markov chain Monte Carlo using ordered overrelaxation. In M. I. Jordan, editor, *Learning in Graphical Models*, pages 205–225. Dordrecht: Kluwer Ac. Pub., 1998.

[4] L. Tierney. Markov chains for exploring posterior distributions. *Annals of Statistics*, 22:1701–1762, 1994.