TUNING OF MARKOV CHAIN MONTE CARLO ALGORITHMS USING COPULAS

Radu V. Craiu[1]

Algoritmii de tipul Metropolis-Hastings se constituie într-una dintre cele mai folosite clase de algoritmi Monte Carlo cu lanţuri Markov (MCMC). Eficienta implementare a acestor algoritmi depinde în mare măsură de abilitatea utilizatorului de a crea o bună distributie de propunere. In această lucrare propunem o metodă de construcţie a distribuţiei de propunere bazată pe modele cu copule. Dacă distribuţia de interes π are suportul într-un spaţiu de dimensiune $d > 2$, construim o distribuţie de propunere bazată pe aproximarea distribuţiilor marginale bi-dimensionale ale lui π obţinute dintr-o selecţie iniţială. Eficacitatea metodei este ilustrată prin simulări.

The Metropolis-Hastings class of algorithms is probably the most widely used in the Markov chain Monte Carlo (MCMC) universe. The efficiency of such algorithms hinges on the statistician's ability to consider a good proposal distribution. We consider here an approach in which the tuning of the proposal distribution is performed using approximations build via copulas. In multivariate settings where the target distribution π has support of dimension $d > 2$ we consider an proposal build upon approximations of the bivariate marginals which are estimated from the available samples. We use simulations to show the gain in efficiency produced by the method.

Keywords: Composite likelihood, Copula models, Frank copula, Markov Chain Monte Carlo, Metropolis-Hastings algorithm.

MSC2000: 53C 05.

1. Introduction of Markov chain Monte Carlo Methods

The development of Markov chain Monte Carlo (MCMC) methods within the statistical community has been spectacular in the last two decades. Generated by the work of [16] and [12] the methodology has enhanced enormously the applicability and impact of Bayesian inference in realistic statistical analyses. The main original purpose of the Monte Carlo method is approximating integrals of the type

$$I = \int f(x)\pi(x)dx \tag{1}$$

[1] Associate Professor, University of Toronto, Toronto, Canada, e-mail: `craiu@utstat.toronto.edu`

where π is a density of interest and f is an integrable function with respect to the measure induced by π. Specifically, when computing analytically (1) is impossible, the Monte Carlo paradigm recommends generating M independent and identically distributed (iid) samples $\{x_1, \dots, x_M\}$ from the density π and approximating (1) with

$$\hat{I}_M = \frac{1}{M} \sum_{m=1}^{M} f(x_i) \tag{2}$$

since law of large numbers guarantees the almost sure convergence of $\hat{I}_M$ to I.

Sometimes, obtaining iid samples from π is an impossible task. The fundamental idea shift behind MCMC is to construct an ergodic Markov chain whose stationary distribution is exactly π and use the realizations obtained along the path of the chain as the Monte Carlo sample. (Throughout the paper, we will use a slight abuse of notation, using π to denote both the distribution and the density of interest.)

Theoretically, if the Markov chain is aperiodic, irreducible and positive Harris recurrent with invariant distribution π then the ergodic theorem implies that with probability one $\hat{I}_M \to I$ as $M \to \infty$; for definitions see [17]. The popularity of MCMC is explainable by the ease with which such a chain can be constructed when π is known up to a normalizing constant, a situation that occurs very often in Bayesian statistical inference.

In practice, after an initialization period, also known as burn-in period, the realizations of the chain are collected as dependent samples following distribution π. One of the most used MCMC methods is the Metropolis-Hastings (MH) algorithm that builds the Markov chain using an accept-reject strategy for each proposed new state of the chain. Each of the potential new states are sampled from a *proposal distribution* q which must be easy to sample from and is allowed to depend on the current state of the chain. More precisely, if the chain is initialized at $X_0 = x_0$ then at any time $t \geq 1$ the MH algorithm performs the following steps:

(1) Generate a *proposal* $y \sim q(\cdot|x_{t-1})$,
(2) Compute the acceptance ratio $r = \min\left\{1, \frac{\pi(y)q(x_{t-1}|y)}{\pi(x_{t-1})q(y|x_{t-1})}\right\}$,
(3) With probability r, set $X_t = y$ and otherwise $X_t = x_{t-1}$.

Popular variants of the MH algorithm are the *random walk Metropolis* (RWM) in which $y = x + \epsilon$ and ϵ is generated from a spherically symmetric distribution, e.g., $\epsilon \sim N(0, \Sigma)$, and the *independent Metropolis* (IM) in which $q(y|x) = q(y)$, i.e. q does not depend on the current state of the chain, x. Generally, the RWM is used in situations in which we have little idea about the shape of the target distribution and therefore we need to "totter" through the sample space. The opposite situation is one in which we have a pretty good idea about the target π and we are able to produce a credible approximation q which can be used as the proposal in the IM algorithm.

The speed and modes of convergence of the MH chains have been studied by [15], [20],[19] and [13], among others. Despite understanding theoretically quite well the general convergence properties of the MH algorithms, in practical implementations the user is left with the difficult task of determining and tuning an appropriate proposal distribution. For instance, in the case of a RWM one has to choose carefully the variance Σ so that a good balance is achieved between the acceptance rate and the chain's autocorrelation function. Similarly, in the case of an IM algorithm, one needs to find a distribution q that satisfies two non-trivial conditions: 1) it approximates reasonably well the target π and 2) it can be easily sampled from. In practice, the process of determining a good proposal requires a back-and-forth strategy in which one starts with an initial proposal and subsequently makes a number of modifications while trying to assess their influence on the performance of the algorithm. This "tune-up" requires re-starting the simulation process a number of times and can be time-consuming and is often frustrating. The difficulties are amplified when the target π has support in a high dimensional space. It is thus useful to develop more automatic or generic ways to update the proposal distribution for a wide spectrum of MCMC applications where MH is needed.

A recent promising direction is offered by the class of adaptive MCMC (AMCMC) algorithms in which the proposal distribution is changed *on the go* at any time t using the information contained in the samples obtained up to time t. Such an approach does not require re-starting of the chain, can be fully automatic but requires careful theoretical analysis since, by using the past realizations of the chain (and not only the current state), the process loses its Markovian property and asymptotic ergodicity must be proven on a case-by-case basis. For more details regarding the theoretical analysis and implementability of AMCMC we refer the reader to [10], [18],[21],[2],[8], [3], [4] and references therein.

In the present paper we propose an alternative approach in which we construct a proposal for π based on a number of two-dimensional marginal distributions built using copula models. In Section 2 we detail the construction of the approximation and the performance of the method is tested with simulations in Section 3. Conclusions and plans for further work are in the last section of the paper.

2. **Copula-based tuning for MCMC**

In this section we assume that of interest is sampling from a distribution π that has support in $\mathbf{R}^d$ and is known up to a normalizing constant. We also assume that from an initial stage of the simulation we have available a sample of size M, $\{x_1, \ldots, x_M\}$, that has been obtained with a generic MCMC algorithm designed to sample from π. For instance, this could be a RWM using a Gaussian proposal with a variance chosen so that the acceptance rate is at least 10%.

Assuming that the target has support of dimension $d = 6$ we can write

$$\pi(x) = \pi_{12}(x_1, x_2|x_3, \ldots, x_6)\pi_{34}(x_3, x_4|x_5, x_6)\pi_{56}(x_5, x_6).$$

For an arbitrary dimension d,

$$\pi(x) = \pi_{d,d-1}(x_d, x_{d-1}|x_{d-2}, \ldots, x_1) \ldots \pi_{i,i-1}(x_i x_{i-1}|x_{i-2}, \ldots, x_1) \ldots$$

The last term in the above product is either $\pi_1(x_1)$ or $\pi_{12} = \pi(x_1, x_2)$ depending on whether d is odd or even. The approximation proposed here is two-fold: on one hand we approximate each conditional bivariate distribution using a bivariate distribution (ignoring the conditions), i.e. $\pi_{i;i-1}(x_i, x_{i-1}|x_{i-2}, \ldots, x_1) \approx h_i(x_i, x_{i-1})$ and on the second hand we set

$$h_i(x, y) \approx n(x|\mu_{1i}, \sigma_{1i})n(y|\mu_{2i}, \sigma_{2i})c_{\theta_i}(\Phi((x - \mu_{1i})\sigma_{1i}^{-1}), \Phi((x - \mu_{2i})\sigma_{2i}^{-1})), \forall i, \tag{3}$$

where $n(\cdot|\mu, \sigma)$ is the density of a Gaussian with mean μ and variance σ^2 and $c_\theta(x, y)$ is the density of a parametric copula density function characterized by the parameter θ.

Essentially, model (3) considers bivariate models in which the marginals are Gaussians and the dependence structure is fitted via a copula model. The choice of the copula family to be used in the approximation is very important as has been discussed by [7]. Throughout the paper we consider the Frank's copula due to its flexibility at modeling both negative and positive dependencies. An adaptive MCMC algorithm due to [10] considers the approximation via a multivariate Gaussian distribution whose parameters are continuously updated based on all the samples available. In large dimensions, learning the entire $d \times d$ covariance matrix can be a lengthy process and by approximating the d-dimensional joint distribution using bivariate joint distributions we aim to accelerate the learning process without a great loss of efficiency.

Due to lack of closed form solutions for the copula parameter estimators, the copula-based approach can be computationally intensive if one continuously adapts the proposal distribution. Instead, we consider a simpler strategy, in which the samples obtained during an *initialization* period are used for tuning the copula model parameters (including the parameters of the marginal normals) only *once*.

Similar to [11] one can also approximate the marginal bivariate distributions using bivariate Gaussians with parameters estimated recursively from the available samples. The recursive updating of the parameters can be performed simultaneously while running the chain since parameters can be estimated in closed form, thus minimizing the computational load. This approach is not explored here.

Remark 2.1. The construction used in (3) is in the same vein of composite likelihood approximations based on bivariate marginals or conditional distributions as discussed by [5] and [22]. However, here we seek an accurate approximation of the whole conditional bivariate distribution rather than just around its mode.

Remark 2.2. It should be noted that recent developments in copula methodology allow more complicated models than (3). For instance one could allow the copula parameter to vary with one or more of the variables we condition on, using estimation approaches similar to [6] or [1]

3. Simulation study

We consider a scenario in which the initial M samples from $\pi = N_d(\mu, \Sigma)$ are obtained using a Metropolis-within-Gibbs with a systematic scan in which the chains updates one coordinate at a time. For a description of the Metropolis-within-Gibbs sampler see [9] and [20]. The proposal for each coordinate is a RWM with a conservatively chosen proposal distribution (to ensure a reasonable acceptance rate). By integrating the information contained in the M samples we also want to increase the efficiency of the algorithm. It is known that updating blocks of coordinates in a Gibbs sampler increases the efficiency of the algorithm [14]. Therefore, we use the information about π contained in the M samples and construct a Metropolis-within-Gibbs algorithm which uses an independent proposal to simultaneously update two coordinates at a time. The proposal distribution for each pair of variables is obtained using the copula-based approximation method proposed in Section 2. We consider the case when $d = 6$ and use $M = 3000$. In **Scenario I** we assume $\mu = 0$, $\Sigma = aa^T + \mathbf{I}_d$ where $a \sim N_d(0, \mathbf{I}_d)$ is a random d-dimensional random vector and $\mathbf{I}_d$ is the d-dimensional identity matrix. We estimate the parameters μ_{1i}, μ_{2i}, σ_{1i}, σ_{2i} and θ_i , for $i = 1, 2, 3$ corresponding to the joint distributions of (x_1, x_2), (x_3, x_4) and (x_5, x_6). As a benchmark, we also consider obtaining the estimators $\hat{\mu}$ and $\hat{\Sigma}$ using the available sample of size M and using as the proposal $N_d(\hat{\mu}, \hat{\Sigma})$. Obviously, the latter approach is optimal when π is Gaussian but may be inefficient otherwise. In **Scenario II** we keep the same Gaussian marginals but the dependence structure is build upon a Clayton copula. More precisely, we use Clayton copulas with parameters $4, 6, 8$ to define the dependence between X_1, X_2, X_3, X_4 and X_5, X_6, respectively.

The algorithms replicate 100 times the simulation of 10,000 samples (the 3000 samples generated using the RWM are also different across replicates) and their performance is based on the MSE error as shown in Table 1. One can see that while in **Scenario I** the copula-based tuning yields results that are slightly inferior to the benchmark approach, under **Scenario II** the proposed method performs better than the method using the normal approximation with reduction in MSE by more than 50% for some of the coordinates.

Table 1. *Mean Squared Error (MSE) estimates for the mean parameter of the 6-dimensional distribution using the normal approximation and the copula-based model.*

Scenario I	Normal proposal					
	0.022	0.016	0.017	0.019	0.018	0.019
	Copula-based tuning					
	0.035	0.035	0.031	0.040	0.034	0.035
Scenario II	Normal proposal					
	0.013	0.016	0.048	0.099	0.131	0.185
	Copula-based tuning					
	0.005	0.010	0.028	0.048	0.125	0.126

In Figure 1 we use one of the replicated studies to illustrate another effect of the copula-based model. Comparing the autocorrelation plots presented in the first row for samples obtained using the RWM algorithm with the ones obtained for samples after the copula-based tuning (bottom row plots), we can see that the autocorrelations for all six parameters have decreased significantly after the implementation of the method.

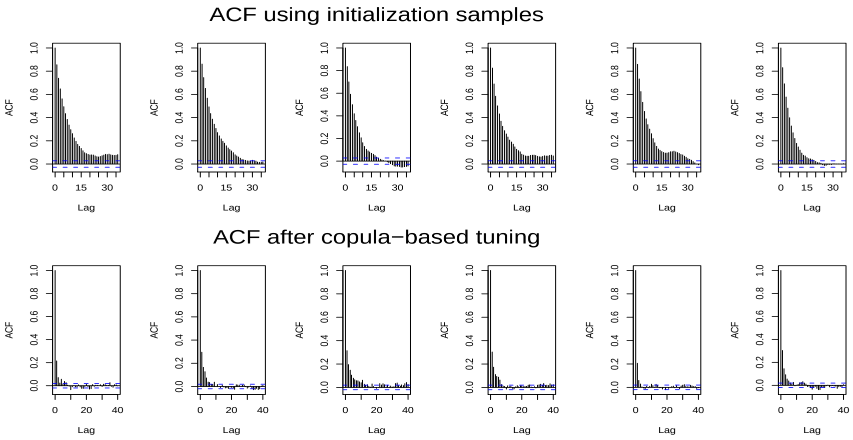

Figure 1. *Autocorrelation plots for all the variables constructed from the samples obtained in the first stage with RWM (top row) and for the samples obtained after using the copula-based approximation of the proposal (bottom row). The reduced correlation implies better mixing of the chain.*

4. **Conclusions and Future work**

We present a general framework for using copulas for tuning of MCMC algorithms. Initial results show that the method can be useful in approximating bivariate marginal/conditional distributions of the target. However, much remains to be done. In practice one would need a realistic copula selection procedure which may allow for different copulas being used across different marginals. The ideas presented here are related to adaptive MCMC and it would be desirable to be able to update the copula parameters as the simulation proceeds. However, this raises considerable the computational burden since no iterative or sequential methods to update the copula parameters are known. Further progress on this issues will be reported elsewhere.

REFERENCES

[1] *E. Acar, R. V. Craiu, and F. Yao.* Dependence calibration in conditional copulas: A nonparametric approach. Biometrics, to appear., 2010.

[2] *C. Andrieu and É. Moulines.* On the ergodicity properties of some adaptive mcmc algorithms. Ann. Appl. Probab., 16(3):1462–1505, 2006.

[3] *C. Andrieu and J. Thoms.* A tutorial on adaptive MCMC. Statist. Comput., 18:343–373, 2008.

[4] *Y. Bai, R. V. Craiu, and A. Di Narzo.* A mixture-based approach to regional adaptation for mcmc. J. Comput. Graph. Statist., to appear, 2010.

[5] *D. Cox and N. Reid.* A note on pseudolikelihood constructed from marginal densities. Biometrika, 91(3):729–737, 2004.

[6] *M. Craiu.* Parametric estimation of conditional copulas. Univ. Politehnica Scientific Bull, 71(3):3–8, 2009.

[7] *R. V. Craiu and M. Craiu.* On the choice of parametric families of copulas. Advances and Applications in Statistics, 10(1):25–40, 2008.

[8] *R. V. Craiu, J. S. Rosenthal, and C. Yang.* Learn from thy neighbor: Parallel-chain adaptive and regional MCMC. Journal of the American Statistical Association, 104:1454–1466, 2009.

[9] *A. E. Gelfand.* Gibbs sampling. J. Amer. Statist. Assoc., 95(452):1300–1304, 2000.

[10] *H. Haario, E. Saksman, and J. Tamminen.* An adaptive Metropolis algorithm. Bernoulli, 7:223–242, 2001.

[11] H. Haario, E. Saksman, and J. Tamminen. Componentwise adaptation for high dimensional MCMC. Computational Statistics, 20:265–273, 2005.

[12] *W. K. Hastings.* Monte Carlo sampling methods using Markov chains and their applications. Biometrika, 57:97–109, 1970.

[13] *S. F. Jarner and E. Hansen.* Geometric ergodicity of metropolis algorithms. Stochastic Processes and their Applications, 85:341–361, 2000.

[14] *J. S. Liu.* Monte Carlo Strategies in Scientific Computing. Springer, 2001.

[15] *K. L. Mengersen and R. L. Tweedie.* Rates of convergence of the hastings and metropolis algorithms. The Annals of Statistics, 24(1):101–121, 1996.

[16] *N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller.* Equations of state calculations by fast computing machines. J. Chem. Ph., 21:1087–1092, 1953.

[17] *S. P. Meyn and R. L. Tweedie* Markov chains and Stochastic Stability. London, Springer-Verlag, 1993.

[18] *G. Roberts and J. Rosenthal.* Coupling and ergodicity of adaptive mcmc. Journal of applied probability, 5(42):458–475, 2007.

[19] *G. O. Roberts and J. S. Rosenthal.* Optimal scaling for various Metropolis-Hastings algorithms. Statist. Sci., 16(4):351–367, 2001.

[20] *G. O. Roberts and J. S. Rosenthal.* Harris recurrence of metropolis-within-gibbs and trans-dimensional markov chains. Ann. Appl. Probab., 16(4):2123–2139, 2006.

[21] *G. O. Roberts and J. S. Rosenthal.* Examples of adaptive MCMC. J. Comput. Graph. Statist., 18(349-367), 2009.

[22] *C. Varin, R. Nancy, and D. Firth.* An overview of composite likelihood methods. Statistica Sinica, to appear., 2010.