OxWaSP Module 1: Adaptive MCMC

Virginia Aglietti Tamar Loach

October 19, 2016

1 Introduction to adaptive MCMC

MCMC algorithms allow sampling from complicated, high-dimensional distributions. Choice of the proposal distribution (from which samples are taken in an attempt to approximate sampling from the target distribution π) determines the ability of the algorithm to explore the parameter space fully and hence draw a good sample. Adaptive MCMC algorithms tackle this challenge by using samples already generated to learn about the target distribution; they push this knowledge back to the choice of proposal distribution iteratively.

This project explores adaptive MCMC algorithms existing in the literature that use covariance estimators to improve convergence to a target distribution supported on a subset of $\mathbb{R}^d$. In this schema we learn about the target distribution π through estimation of its correlation structure from the MCMC samples. We use this correlation structure to improve our estimate of the target. The performance of the algorithm depends heavily on the choice of the proposal distribution and its covariance structure. Different choices for the proposal covariance matrix may lead to different results. On the one hand, if the proposal covariance matrix is too narrow, the parameter space won't be properly explored. On the other hand, if the proposal covariance matrix is too wide, the rejection rate may be very high. In order to obtain a chain that adapts properly and settles down to rapid mixing, we need to select an optimal value for the covariance matrix. Roberts *et al.* (1997) have first shown that, under specific assumptions about the target distribution, the optimal value for the proposed covariance matrix is such that the acceptance rate of the algorithm is 0.234, independently of the d-dimensional target distribution with *iid* components. An optimal acceptance rate $\alpha^* = 0.234$ will be used to compare the performance of the algorithms discussed in this report.

2 The AM algorithm

We first implement an adaptive MCMC algorithm which we will here call AM (Haario et al., 2001). This is a modification of the random walk Metropolis-Hastings algorithm. In the AM algorithm the proposal distribution is updated at time t to be a normal distribution centered on the current point X_{t-1} with

covariance $C_t(X_0, ..., X_{t-1})$ that depends on the the whole history of the chain. The use of historic states means the resulting chain is non-markovian, and reversibility conditions are not satisfied. Haario *et al.* show that the right ergodic properties and correct simulation of the target distribution nonetheless remain. Provided we use a asymptotically symmetric proposal distribution the probability with which to accept candidate points in the chain is:

$$\alpha(X_{t-1}, Y) = \min\left(1, \frac{\pi(Y)}{\pi(X_{t-1})}\right). \tag{1}$$

With C_t given by:

$$C_t = s_d \text{cov}(X_0, ..., X_{t-1}) + s_d \epsilon I_d. \tag{2}$$

Here cov() is the usual empirical covariance matrix:

$$\text{cov}(x_0, ..., x_k) = \frac{1}{k}\left(\Sigma_{i=0}^{k} x_i x_i^T - (k+1)\bar{x}_k \bar{x}_k^T\right), \tag{3}$$

and the parameter $s_d = \frac{2.4^2}{d}$ (Gelman et al., 1996). ϵ is chosen to be very small compared to the subset of $\mathbb{R}^d$ upon which the target function is supported. The AM algorithm is computationally feasible due to recursive updating of the covariance matrix on acquisition of each new sample through the relation:

$$C_{t+1} = \frac{t-1}{t} C_t + \frac{s_d}{t}(t\bar{X}_{t-1}\bar{X}_{t-1}^T - (t+1)\bar{X}_t \bar{X}_t^T + X_t X_t^T + \epsilon I_d). \tag{4}$$

The mean is in turns calculated recursively by:

$$\bar{X}_{t+1} = \frac{t\bar{X}_t + X_{t+1}}{t+1}. \tag{5}$$

Because of the instability of the covariance matrix, to implement the adaptivity we first run the algorithm with no change to the covariance of the proposal distribution. The adaptation starts at a user defined point in time, and until this time the covariance of the proposal is chosen to represent our best knowledge of the target distribution. We use a Gaussian distribution with no correlation structure thorughout this report.

3 An example - testing the AM algorithm

We now numerically test the AM algorithm. We have used two different target distributions: a correlated Gaussian distribution $N(0, \Sigma)$ and a banana-shaped distribution (Roberts and Rosenthal, 2009) given by:

$$f_B(x_1, ..., x_d) \propto \exp\left[-x_1^2/200 - \frac{1}{2}\left(x_2 + Bx_1^2 - 100B\right)^2 - \frac{1}{2}\left(x_3^2 + x_4^2 + ... + x_d^2\right)\right]. \tag{6}$$

$B > 0$ is the so called bananicty constant (set to 0.1 throughout) and d is the dimension. We have chosen the correlated Gaussian distribution as targeting this demonstrates how the use of empirical covariance improves convergence - we learn the target's covariance as we move through steps of the MCMC. The banana-shaped distribution is an additional example with an irregular shape. We use this to test the ability of the markov chain to fully explore the state space with and without adaption. We first run our implementation of the usual random-walk Metropolis-Hasting algorithm, and the AM adaptation modification of this, each time targetting $N(0, \Sigma)$. We have chosen Σ to be generated from eigenvalues chosen unifromly at random on [1,10].

The crucial function in the R package that we have created is called `mcmc()`. This function takes as arguments the following parameters:

```
d = 8                  #  Dimension of the parameter space
n_iter = 20000         #  Total number of iterations
x1 = rep(5,d)          #  Vector of inital values
t_adapt = 2000         #  When to start adapting
adapt = "AM"           #  Algorithm to run
target = pi_norm_corr  #  Target distribution function
```

There are several target distribution functions built in to our package for testing the algorithm. Notice that `x1` represents the starting point of the chain and is user specified. We show here the call to the function `mcmc()` using as target distribution a correlated Gaussian distribution:

```
X = mcmc(target = target,
         n_iter = n_iter,
         x_1 = x1,
         adapt=adapt,
         t_adapt = t_adapt
         )
```

4 Comparing AM and random-walk Metropolis-Hastings

Figure 1 shows the first component and the second component of the Markov chain at each iteration. The plot demostrates poor mixing of the MH algorithm when targeting a positively correlated multivariate Gaussian distribution (Figure 2).

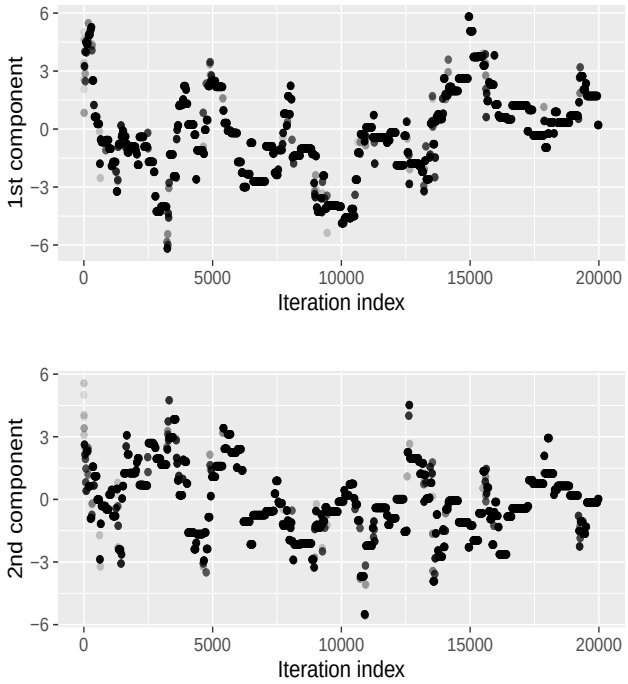

Figure 1: The first (top) and second (bottom) components of the Markov chain resulting from Metropolis-Hastings (MH) targetting a correlated 8-dimensional Gaussian distribution.

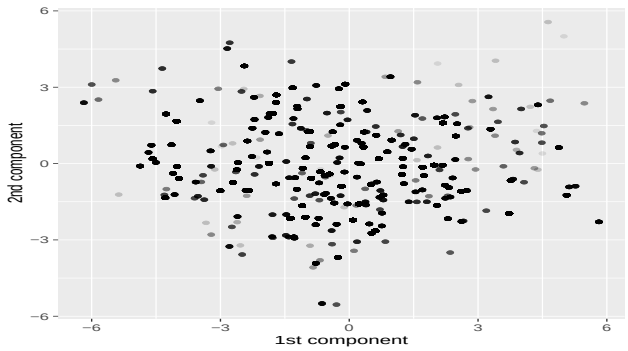

Figure 2: The relationship between the first two components of the Markov chain generated using Metropolis-Hasting targeting the correlated Gaussian distribution.

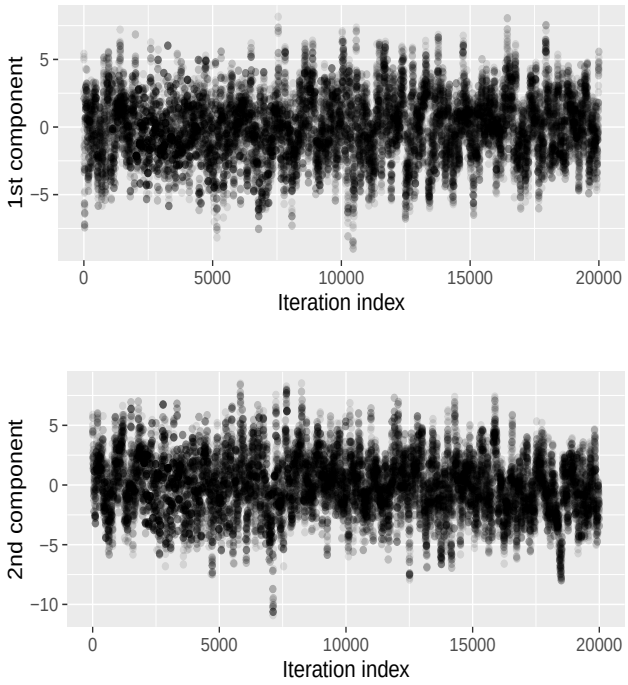

Figure 3: The first (top) and second (bottom) components of the Markov chain resulting from the AM algorithm targetting a correlated 8-dimensional Gaussian distribution.

Figure 3 shows the same results obtained from implementing the AM algorithm. The AM algorithm seems to perform better in terms of parameter space exploration and appears to settle down to a rapid mixing. In addition it seems to better capture the correlation existing among the variables (Figure 4).

We now repeat this analysis using a multidimensional banana-shape distribution as target distribution. Indeed, the AM algorithm is expected to work particularly well on irregularly shaped target densities in which the density contours form roughly elliptical contours. Figure 5 proves rapid mixing of the AM algorithm with respect to the MH algorithm.

5 Comparing AM and AM2

We now explore a slight modification to the adaptation scheme which we will here call AM2 (Roberts and Rosenthal, 2009). This algorithm uses stochastic stabilisation rather than the numerical stabilisation of AM. Roberts and Rosenthal use a mixture of Gaussians as the proposal distribution: with proportion

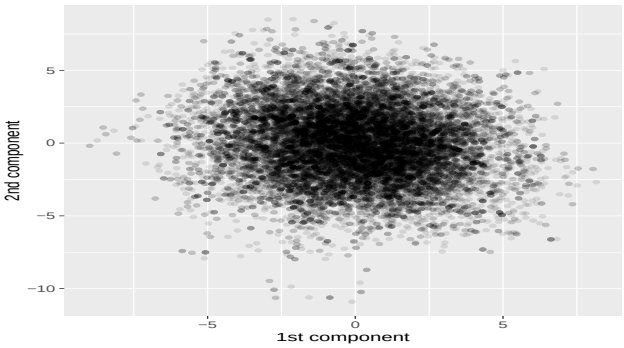

Figure 4: The relationship between the first two components of the Markov chain resulting from the adaptive metropolis algorithm (AM) targeting a correlated 8-dimensional Gaussian distribution.

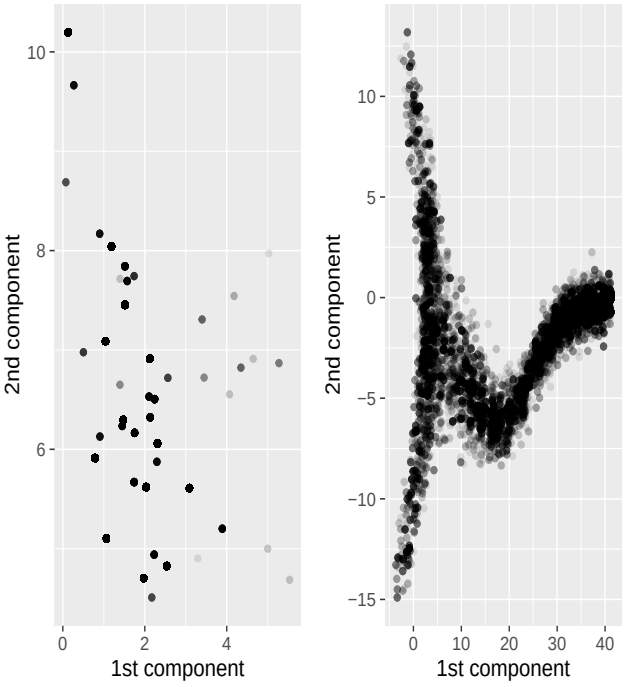

Figure 5: The first and second components of the samples resulting from MH algorithm (left) and AM algorithm (right) targeting an 8-dimensional banana shaped distribution.

β a normal uncorrelated distribution is mixed with a correlated normal distribution. The proposal becomes:

$$Q_n(x, \cdot) = (1 - \beta)N(x, s_d\Sigma_n) + \beta(N(x, (0.1^2)I_d/d) \tag{7}$$

Our implementation (AM2) is otherwise the same as AM1. We choose to leave the choice of the time at which the adaptation is introduced to the user; Roberts and Rosenthal specify the adaptation time to be when there have been more than $2d$ iterations but we find better performance with a longer period of standard random walk Metropolis-Hastings first. The results for our implementation of AM2 are as follows.

6 Less naive covariance estimation

The usual empirical covariance matrix estimator used so far in the adaptive AM and AM2 algorithms is optimal in the classical setting with large samples and fixed low dimensions. This estimator performs poorly in the high dimensional setting and in particular when the dimension of the parameter space is larger than the number of observations. In the recent literature, several alternative covariance matrix estimation techniques have been proposed. In this paper we will focus on two different regularization techniques: the shrinkage estimator and a Cholesky-based method.

Define the empirical covariance matrix as in (3) and a target identity matrix as $D = \text{diag}(d)$ where d represents the dimension of the problem. The idea of shrinkage estimation of the covariance matrix is to take a weighted average of the empirical covariance matrix and D given a parameter λ representing the shrinkage intensity - see Schafer *et al.* (Schafer et al., 2005) for its computation.The covariance matrix used in the algorithm can be defined as:

$$\hat{C} = (1 - \lambda)\hat{\Sigma} + \lambda * \hat{D}. \tag{8}$$

Notice that $0 < \lambda < 1$. When $\lambda = 0$, no shrinkage is applied to the sample covariance matrix and the empirical covariance matrix is used. In contrast $\lambda = 1$ is associated with complete shrinkage of all pairwise covariances. In this case, we are thus ignoring the existence of any covariances among the random variables considered. It is possible to show that, under general conditions, there exists a shrinkage intensity for which the resulting shrinkage estimator contains less estimation error than the original empirical estimator (James and Stein, 1961).

As an alternative to both the empirical and the shrinkage estimator of the covariance matrix, we have used a regularization techniques based on the Cholesky decomposition of the covariance matrix (Rothman et al., 2010). Consider covariance matrix given by Σ. Define $\Sigma = LDL^T$ to be the modified Cholesky decomposition of the covariance matrix where D is diagonal and L is lower triangular with ones on the diagonal. The Cholesky-based method is based on the

idea of bounding the Cholesky factor L. This means introducing sparsity in the Cholesky factor L estimating only the first k sub diagonal and setting the rest to zero. The bounding parameter k must be less than min(n-1,p). Notice that a similar approach for bounding the inverse of the covariance matrix has been proposed by (Bickel and Levina, 2008). However, Rothman *et al.* have shown that the Cholesky based regularization method can be applied directly to the covariance matrix itself to obtain a sparse estimator with guaranteed positive definiteness (Rothman et al., 2010).

In order to reduce the computational cost of the algorithm, the shrinkage estimator of the covariance structure has been computed using a recursion formula similar to (4). On the contrary, the Cholesky-based covariance estimator has been evaluated at each step. Whilst this negatively impacts on the running-time of the algorithm, it doesn't affect its performance in terms of convergence rate α.

The following plot shows α for all the discussed algorithms. We can see how the acceptance rate is far away from the optimum for the MH, AM and AM2 algorithms. Only when we introduce less naive covariance matrix estimators do we achieve better acceptance rates (Figure 7 and Figure 6).

7 Conclusions

We have shown that the AM adaptive algorithm and the AM2 adaptive algorithm perform better than the simple random-walk Metropolis-Hastings algorithm. This is true for irregularly shaped distributions which are more challenging to sample from.

References

Peter J Bickel and Elizaveta Levina. Regularized estimation of large covariance matrices. *The Annals of Statistics*, pages 199–227, 2008.

A Gelman, GO Roberts, and WR Gilks. Efficient metropolis jumping rules. 1996.

Heikki Haario, Eero Saksman, and Johanna Tamminen. An adaptive metropolis algorithm. *Bernoulli*, 7(2):223–242, 04 2001. URL `http://projecteuclid.org/euclid.bj/1080222083`.

William James and Charles Stein. Estimation with quadratic loss. In *Proceedings of the fourth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 361–379, 1961.

Gareth O. Roberts and Jeffrey S. Rosenthal. Examples of adaptive mcmc. *Journal of Computational and Graphical Statistics*, 18(2):349367, 2009. doi: 10.1198/jcgs.2009.06134. URL `http://dx.doi.org/10.1198/jcgs.2009.06134`.

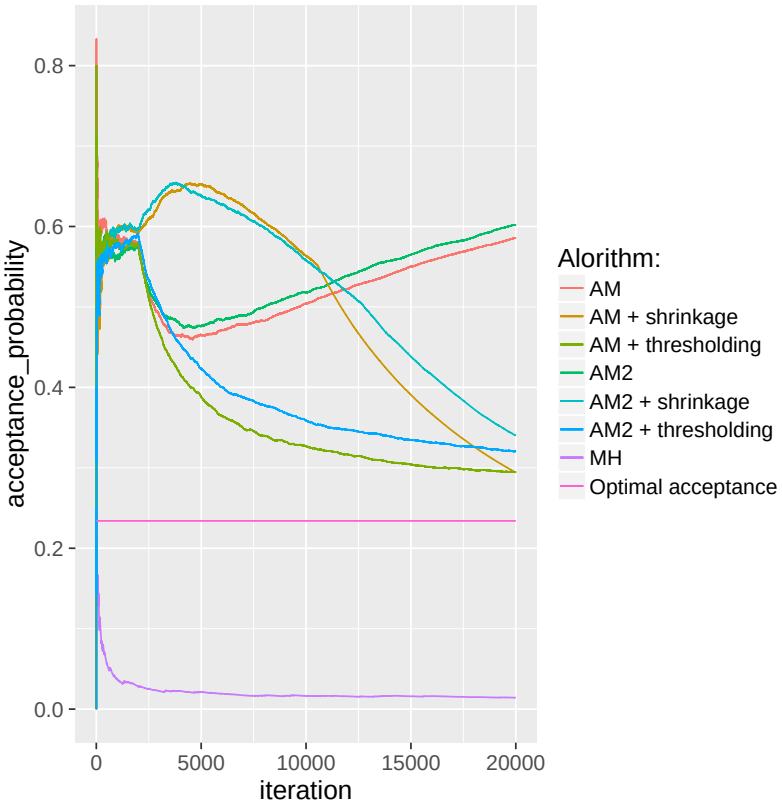

Figure 6: The acceptance probability for the implemented algorithms with a 8-dimensional correlated Gaussian target distribution. The optimal acceptance probability is also shown.

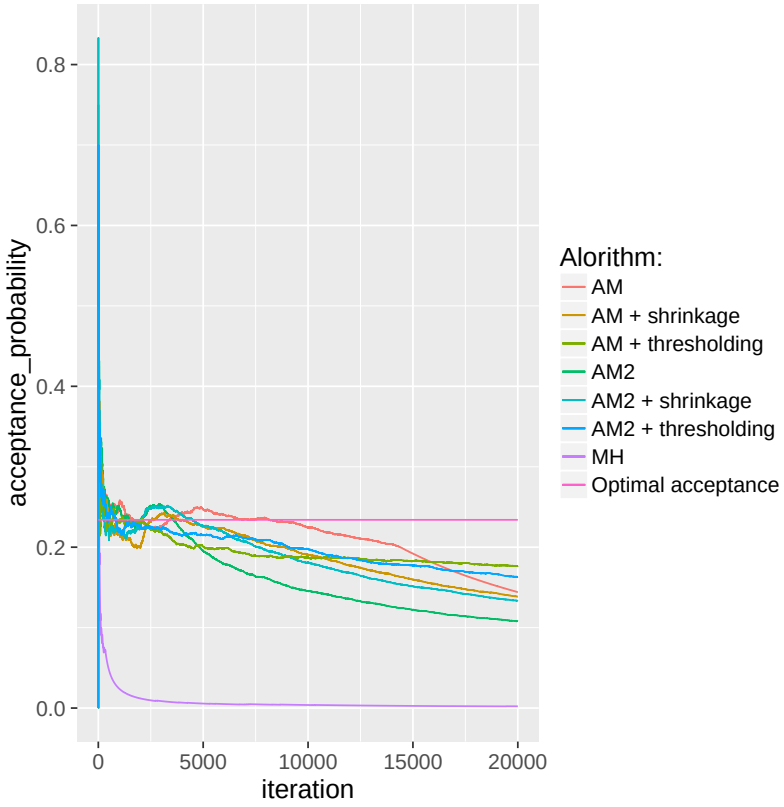

Figure 7: The acceptance probability for the implemented algorithms with a multidimensional banana shaped target distribution.

Adam J Rothman, Elizaveta Levina, and Ji Zhu. A new approach to cholesky-based covariance regularization in high dimensions. *Biometrika*, page asq022, 2010.

Juliane Schafer, Korbinian Strimmer, et al. A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics. *Statistical applications in genetics and molecular biology*, 4(1):32, 2005.