Adaptive MCMC for Component-wise Metropolis Hastings

Radu Craiu

Department of Statistical Sciences
University of Toronto

Université du Québec à Montréal, March 2019

Outline

Markov Chain Monte Carlo

- A search for Markov chain Monte Carlo (or MCMC) articles on Google Scholar yields over 100,000 hits.
- A general web search on Google yields 1.7 million hits. Why so popular?
- MCMC algorithms are used to solve problems in many scientific fields, including physics (where many MCMC algorithms originated), chemistry and computer science.
- The widespread popularity of MCMC samplers is largely due to their impact on solving statistical computation problems related to Bayesian inference.

Challenges ahead

- Thanks to MCMC developments, for 30+ years Bayesian statisticians were *computationally liberated* when thinking about a statistical model.
- The ubiquity of massive data samples and the higher inferential expectations require a drastically augmented level of model complexity.
- Together, these two aspects pose very serious challenges to computational statistics.

A good friend: The Metropolis-Hastings algorithm

- The Metropolis-Hastings sampler is one of the most used algorithms in MCMC. It operates as follows:
  - Given the current state of the MC, θ, a "proposed sample" ξ is drawn from a proposal density $q(\xi|\theta)$.
  - Accept ξ with probability $\min\left\{1, \frac{\pi(\xi)q(\theta|\xi)}{\pi(\theta)q(\xi|\theta)}\right\}$.
  - If ξ is accepted, the next state is ξ, otherwise it is (still) θ.
- The *random walk Metropolis (RWM)* is obtained if $q(\xi|\theta) = q(\theta|\xi)$
- Note that $\pi(\theta)$ needs to be computed at each iteration. (hence $L(\theta|\mathcal{D})$ must also be computable)

Difficult Computation Scenarios

When data $\mathcal{D}$ is massive:

- Divide data into batches, $\mathcal{D}_1 \cup \ldots \mathcal{D}_K$, distribute the sampling from the K sub-posteriors $\pi_j(\theta) \propto [L_k(\theta|\mathcal{D}_j)]^a [p_j(\theta)]^b$ among K processing units
- Design recombination strategies (related to the choice of a, b) for the samples to recover info for the full posterior distribution (Scott et al, IJMSEM 2016; Entezari et al., CJS 2018)
- Irreversible MCMC - the Zig-Zag sampler/deterministic MCMC (Neal 2005; Bierkens et al, AOS 2019)

Difficult Computation Scenarios

When $L(\theta|\mathcal{D})$ is not computable:

- Pseudo-marginal approach (Andrieu & Roberts, AOS 2009) requires only an unbiased estimator of L, $\tilde{L}$ so MH the accept probability is

$$\min\left\{1, \frac{\tilde{\pi}(\xi)q(\theta|\xi)}{\tilde{\pi}(\theta)q(\xi|\theta)}\right\},$$

  where $\tilde{\pi}(\theta) \propto p(\theta)\tilde{L}(\theta|\mathcal{D})$.

- If one can simulate from $p(\mathcal{D}|\theta)$ for all θ then the Approximate Bayesian Computation (Marin et al., Comp & Stat. 2012) or Bayesian Synthetic Likelihood (Price et al, JCGS 2018) methods can be used.

Many/All of the above require "good" transition kernels and that is where adaptive MCMC can play an important role.

The Component-wise MH

- We want to sample from $\pi(x)$ where $x \in \mathcal{X} \subset \mathbf{R}^d$
- When d is large and/or π has irregular features we update x one component at a time using a MH rule.
- MH acceptance probability for the kth component involves $\pi(\cdot|x_{[-k]})$, $1 \leq k \leq d$.
- Ideally we would like to know the "right" proposal for $\pi(\cdot|x_{[-k]})$ as $x_{[-k]}$ varies.

Multiple-try Metropolis

- MTM is a generalization of the Metropolis-Hastings sampler (Liu, Liang and Wong, JASA 2000; Casarin, Craiu and Leisen, Stats & Comp 2013).
- m candidate moves (instead of one) are competing for selection.
- The proposals can be generated from *different* distributions.
- The selection of proposals can be modified to favour longer jumps.
- Selected candidate is accepted/rejected based on a MH-type ratio.

Multiple-try Metropolis

- Candidate $y_j \sim q_j(\cdot|x_t)$, $1 \leq j \leq m$.
- Select one of the candidate one with probability

$$w_j(y_j|x_t) \propto \pi(y_j) q_j(x_t|y_j) \lambda_j(x_t, y_j)$$

- Accept y with probability

$$\min \left\{ 1, \frac{\sum_{j=1}^{J} w_j(y_j|x_t)}{\sum_{j=1}^{J} w_j(x_j^*|y)} \right\}$$

  where y is the selected candidate and $x_j^* \sim q_j(\cdot|y)$
- If q_j are symmetric and $\lambda_j(x, y) = \frac{\|x-y\|^\alpha}{q_j(x|y)}$ then

$$w_j(y_j|x_t) \propto \pi(y_j) \times \|x_t - y_j\|^\alpha$$

Component-wise Multiple-try Metropolis

- For each coordinate we use an MTM transition kernel.
- For updating the kth coordinate, the jth proposal is generated as $y_j^{(k)} = (x_{t,1}, \ldots, x_{t,k-1}, z_j^{(k)}, x_{t,k+1}, \ldots, x_{t,d})$ where $z_j^{(k)} \sim N(x_{t,k}; \sigma_j^{(k)})$.
- The aim: a simple and intuitive method for finding appropriate values for $\sigma_j^{(k)}$, $k = 1, \ldots, d$, $j = 1, \ldots, m$.

Illustration for CMTM update

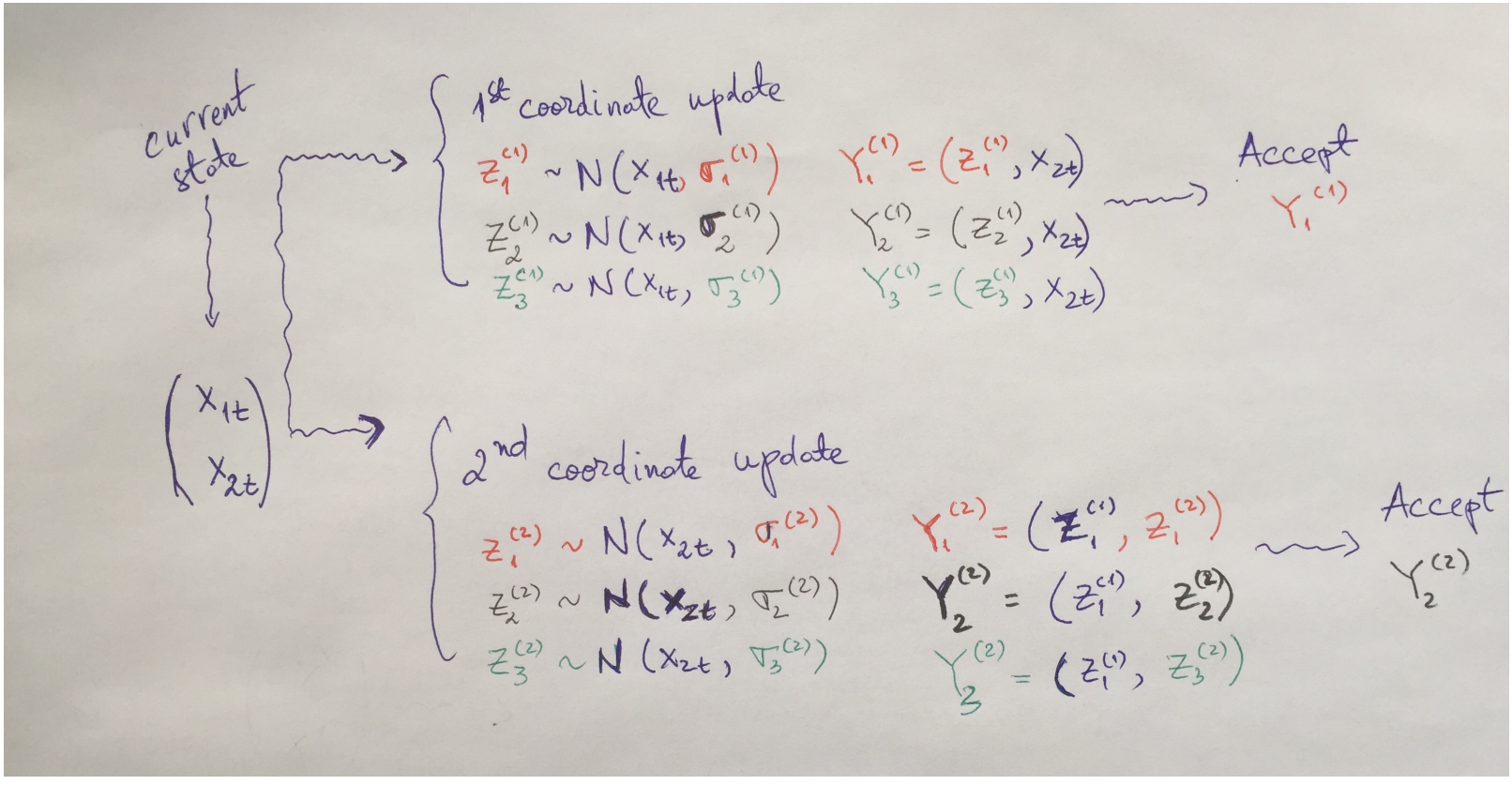


Illustration of CMTM update when $d = 2$ and $m = 3$.

Road ahead

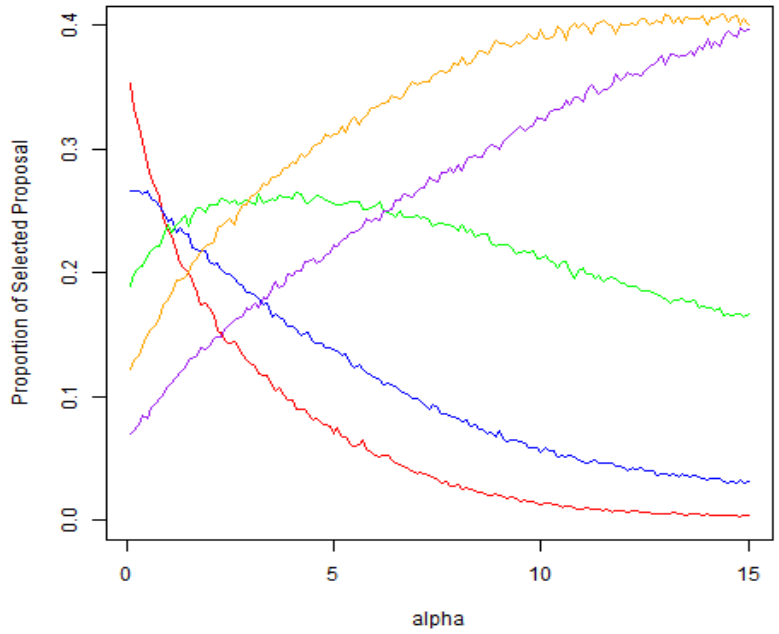
- Choice of α
- Does CMTM do what we expect it to do? Role of selection vs. acceptance.
- Can we improve CMTM using adaptive ideas?

Choice of α

As α increases, the jump distance will become more important for selection.

$0.5N(\mu_1, \Sigma_1) \quad + \quad 0.5N(\mu_2, \Sigma_2)$
where

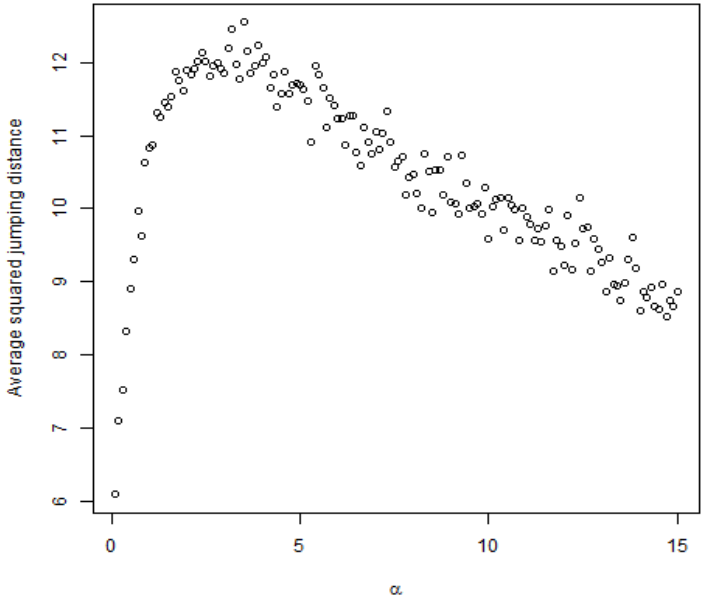
$$\begin{cases} \mu_1 &= (5,0)^T \\ \mu_2 &= (15,0)^T \\ \Sigma_1 &= diag(6.25, 6.25) \\ \Sigma_2 &= diag(6.25, 0.25) \end{cases}$$



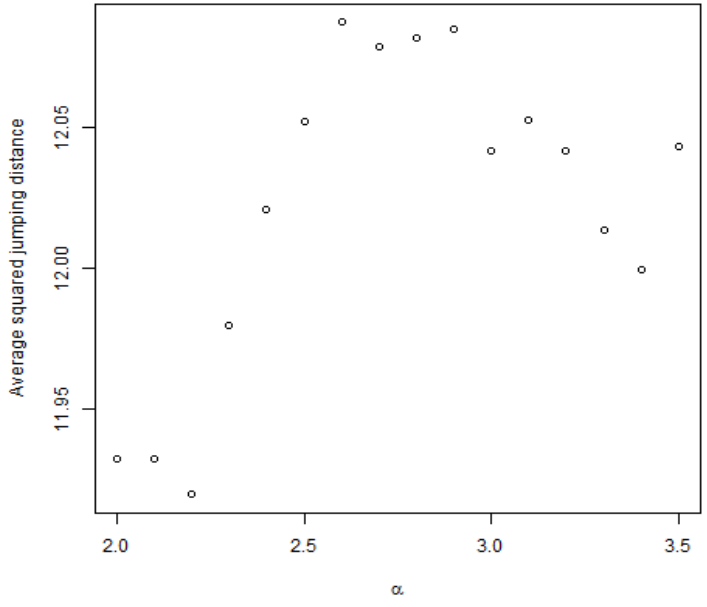
Order of σ's: Red $<$ Blue $<$ Green $<$ Yellow $<$ Purple

Choice of α

What is the effect of α on acceptance/jump distance?



1 run



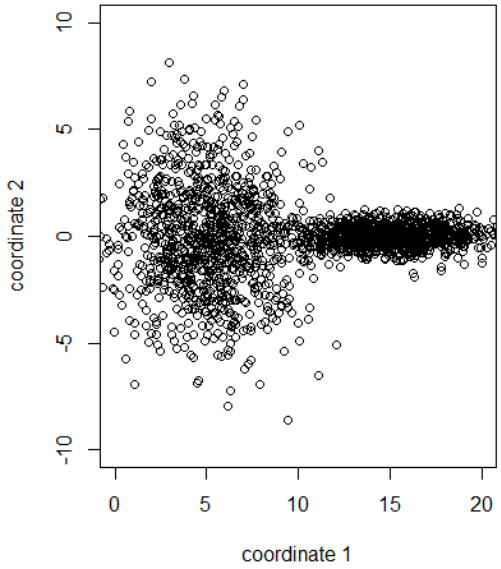
Averaged over 100 runs

We fix $\alpha = 2.9$ throughout.

CMTM Selects "Locally Better" Proposal Distributions

Coordinate 1

	$\sigma_j^{(k)}$				
——	1	2	4	8	16
$X_{n,1} < 10$	0.05	0.15	0.28	0.31	0.22
$X_{n,1} \geq 10$	0.04	0.12	0.25	0.34	0.25



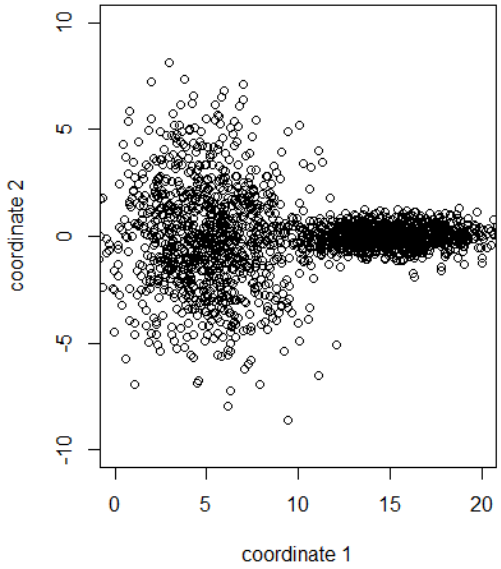
Coordinate 2

	$\sigma_j^{(k)}$				
——	1	2	4	8	16
$X_{n,1} < 10$	0.06	0.19	0.33	0.26	0.16
$X_{n,1} \geq 10$	0.37	0.29	0.19	0.11	0.05

Acceptance frequencies Conditional on Selection

Coordinate 1

	$\sigma_j^{(k)}$				
—	1	2	4	8	16
$X_{n,1} < 10$	0.39	0.50	0.51	0.53	0.50
$X_{n,1} \geq 10$	0.40	0.44	0.44	0.45	0.38



Coordinate 2

	$\sigma_j^{(k)}$				
—	1	2	4	8	16
$X_{n,1} < 10$	0.46	0.52	0.50	0.47	0.44
$X_{n,1} \geq 10$	0.44	0.41	0.30	0.28	0.28

Adaptive MCMC

- Adaptive MCMC tune on the go the simulation parameters of the sampling algorithm.
- Most MCMC methods perform local adaptation but "dance" around the Markovian property and manage to preserve it. Not AMCMC!
- Makes validation of an adaptive scheme more involved, but...
- ... Frees us to seek other practically useful designs!

Adaptive ideas for CMTM

- Selection frequencies are more indicative of how good a scale is in a given region.
- Acceptance frequencies are less indicative.
- If one of the extreme scales is over-selected we need to make changes.
- A small goal: avoid choosing only among poorly calibrated proposal distributions in any region of the space.
- The chain's performance is measured in terms of $ACT \propto ESS^{-1}$

Adaptive CMTM

- For coordinate $k \in \{1, \ldots, d\}$ we use m Gaussian proposals with sd's $S_k = \{\sigma_1^{(k)} < \sigma_2^{(k)} < \ldots < \sigma_m^{(k)}\}$.
- Changes in the transition kernel occur at fixed points in the simulation process, called *adaption points* and only if an *alarm is triggered*.
- Alarm: If proposals with sd $= \sigma_1^{(k)}$ (or $\sigma_m^{(k)}$) are selected more than $100 \cdot \frac{2}{m}\%$ or less than $100 \cdot \frac{1}{2m}\%$ then $\sigma_1^{(k)}$ (or $\sigma_m^{(k)}$) is doubled, respectively halved. The remaining sd's are recalculated to be equidistant on the log scale.
- The idea is for $[\sigma_1^{(k)}, \sigma_m^{(k)}]$ to contain some reasonable values for whatever region the chain visits.

Adaptive CMTM

- Diminishing Adaptation:

$$\lim_{n\to\infty} D_n = 0 \text{ in probability },$$

where $D_n = \sup_{X\in\mathcal{X}} ||\, T_{\gamma_{n+1}}(X,\cdot) - T_{\gamma_n}(X,\cdot)||_{TV}$
$(||\mu-\nu||_{TV} = \sup_{A\in\mathcal{X}} |\mu(A)-\nu(A)|)$.

- We check for an alarm at the rth adaption point with probability $p_r \leq 1$ where

$$p_r = \max\left(0.99^{r-1}, \frac{1}{\sqrt{r}}\right), \qquad r = 1,2,3,\ldots$$

- Diminishing Adaptation is guaranteed by design.
- We continue to adapt indefinitely, but less and less often.

How important is adaption anyway?

- $0.5N(\mu_1, \Sigma_1) + 0.5N(\mu_2, \Sigma_2)$, where

$$\begin{aligned} \mu_1 &= (5, 5, 0, 0)^T \\ \mu_2 &= (15, 15, 0, 0)^T \\ \Sigma_1 &= diag(6.25, 6.25, 6.25, 0.01) \\ \Sigma_2 &= diag(6.25, 6.25, 0.25, 0.01). \end{aligned}$$

- $m = 20$

Without Adaptation

	Min.	Median	Mean	Max.
sq. jump	6.20	6.62	6.62	7.07
	coord1	coord2	coord3	coord4
ACT	41.96	41.25	1.64	1.64

With Adaptation

	Min.	Median	Mean	Max.
sq. jump	8.88	10.15	10.04	10.76
	coord1	coord2	coord3	coord4
ACT	22.55	22.46	1.43	1.00

How important is adaption anyway? - Final $\sigma_j^{(k)}$'s

No Adaptation:

$$\sigma_j^{(k)} = 2^{-11+j}$$

for $j = 1, \ldots, 20$ and $\forall k$.

	Adaptation			
j	co1	co2	co3	co4
1	4.00	4.00	2.00	0.12
2	4.14	4.14	2.07	0.13
3	4.30	4.30	2.15	0.14
4	4.46	4.46	2.23	0.15
5	4.62	4.62	2.31	0.16
6	4.80	4.80	2.40	0.18
7	4.97	4.97	2.48	0.19
8	5.16	5.16	2.58	0.20
9	5.35	5.35	2.67	0.22
10	5.55	5.55	2.77	0.24
11	5.76	5.76	2.88	0.25
12	5.97	5.97	2.98	0.27
13	6.19	6.19	3.09	0.30
14	6.42	6.42	3.21	0.32
15	6.66	6.66	3.33	0.34
16	6.91	6.91	3.45	0.37
17	7.17	7.17	3.58	0.40
18	7.43	7.43	3.71	0.43
19	7.71	7.71	3.85	0.46
20	8.00	8.00	4.00	0.50

How important is adaption anyway? - Selection frequencies

No Adaptation:

$\sigma_j^{(k)}$	co1	co2	co3	co4
2^{-10}	0.00	0.00	0.00	0.00
2^{-9}	0.00	0.00	0.00	0.00
2^{-8}	0.00	0.00	0.00	0.00
2^{-7}	0.00	0.00	0.00	0.00
2^{-6}	0.00	0.00	0.00	0.00
2^{-5}	0.00	0.00	0.00	0.04
2^{-4}	0.00	0.00	0.00	0.12
2^{-3}	0.00	0.00	0.02	0.24
2^{-2}	0.00	0.00	**0.06**	0.26
2^{-1}	0.01	0.01	**0.20**	0.17
2^{0}	0.05	0.06	**0.24**	0.08
2^{1}	0.14	0.14	**0.20**	0.04
2^{2}	0.26	0.26	**0.13**	0.02
2^{3}	0.24	0.25	**0.08**	0.01
2^{4}	0.14	0.14	**0.03**	0.00
2^{5}	0.09	0.07	0.02	0.00
2^{6}	0.04	0.04	0.01	0.00
2^{7}	0.02	0.02	0.01	0.00
2^{8}	0.01	0.01	0.00	0.00
2^{9}	0.00	0.00	0.00	0.00

Adaptation:

j	co1	co2	co3	co4
1	0.04	0.05	0.05	0.04
2	0.05	0.05	0.05	0.05
3	0.05	0.05	0.05	0.05
4	0.05	0.04	0.05	0.05
5	0.05	0.05	0.05	0.05
6	0.05	0.05	0.05	0.05
7	0.05	0.05	0.05	0.04
8	0.05	0.05	0.05	0.05
9	0.05	0.05	0.05	0.06
10	0.05	0.05	0.05	0.05
11	0.05	0.05	0.04	0.05
12	0.05	0.05	0.05	0.05
13	0.05	0.05	0.05	0.06
14	0.05	0.05	0.05	0.05
15	0.05	0.05	0.05	0.05
16	0.05	0.05	0.05	0.05
17	0.05	0.05	0.05	0.05
18	0.05	0.05	0.05	0.04
19	0.05	0.05	0.05	0.04
20	0.05	0.05	0.05	0.04

How important is adaption anyway? - Acceptance Probs

No Adaptation:

$\sigma_j^{(k)}$	co1	co2	co3	co4
2^{-10}	NaN	NaN	NaN	NaN
2^{-9}	NaN	NaN	NaN	NaN
2^{-8}	NaN	NaN	NaN	NaN
2^{-7}	NaN	NaN	NaN	0.17
2^{-6}	NaN	NaN	NaN	0.52
2^{-5}	NaN	NaN	1.00	0.44
2^{-4}	0.50	NaN	0.50	0.52
2^{-3}	0.00	0.00	0.42	0.50
2^{-2}	0.17	0.43	0.53	0.47
2^{-1}	0.49	0.38	0.58	0.47
2^{0}	0.54	0.45	0.49	0.44
2^{1}	0.57	0.52	0.52	0.45
2^{2}	0.51	0.49	0.49	0.37
2^{3}	0.48	0.45	0.47	0.41
2^{4}	0.46	0.45	0.48	0.33
2^{5}	0.41	0.48	0.48	0.33
2^{6}	0.40	0.35	0.50	0.43
2^{7}	0.45	0.31	0.45	0.38
2^{8}	0.47	0.24	0.35	0.00
2^{9}	0.33	0.45	0.61	NaN

Adaptation:

j	co1	co2	co3	co4
1	0.58	0.66	0.49	0.60
2	0.57	0.58	0.58	0.60
3	0.60	0.65	0.62	0.60
4	0.63	0.55	0.59	0.60
5	0.61	0.59	0.58	0.65
6	0.65	0.53	0.60	0.60
7	0.59	0.59	0.60	0.62
8	0.64	0.65	0.58	0.60
9	0.58	0.57	0.59	0.60
10	0.57	0.61	0.60	0.56
11	0.61	0.66	0.59	0.54
12	0.57	0.54	0.62	0.66
13	0.53	0.54	0.66	0.60
14	0.55	0.58	0.57	0.61
15	0.61	0.60	0.58	0.55
16	0.58	0.61	0.60	0.60
17	0.54	0.65	0.61	0.57
18	0.58	0.61	0.58	0.53
19	0.56	0.56	0.62	0.60
20	0.61	0.63	0.66	0.59

Technical conditions

- **(A1)** We choose a (very large) non-empty compact subset $K \subset \mathcal{X}$, and force $X_n \in K$ for all n. Specifically, we reject all proposals $Y_{n+1} \notin K$ (but if $Y_{n+1} \in K$, then we still accept/reject Y_{n+1} by the usual rule for the CMTM algorithm. The chain is started in K.
- **(A2)** We choose a (very large) constant $L > 0$ and a (very small) constant $\epsilon > 0$, and force the proposal scalings $\sigma_{k,j}$ to always be in $[\epsilon, L]$. Specifically, if $\sigma_{t,j}^{(k)}$ is the value of $\sigma_j^{(k)}$ used at the t-th iteration in our adaptive CMTM algorithm, then if $\sigma_{t,j}^{(k)}$ would be greater than L, we instead set $\sigma_{t,j}^{(k)} = L$, while if $\sigma_{t,j}^{(k)}$ would be less than ϵ, we instead set $\sigma_{t,j}^{(k)} = \epsilon$. Correspondingly, the initial values $\sigma_{0,j}^{(k)}$ should all be chosen in $[\epsilon, L]$.

Validity

Theorem

Consider the adaptive CMTM algorithm with open state space $\mathcal{X} \subset \mathbb{R}^d$. Let π be a target probability distribution, which has a continuous positive density on K with respect to Lebesgue measure. Then, the adaptive CMTM algorithm converges to stationarity as in

$$\lim_{n \to \infty} \sup_{A \in \mathcal{F}} |\mathbf{P}(X_n \in A) - \pi(A)| = 0. \tag{1}$$

Comparison with other AMCMC

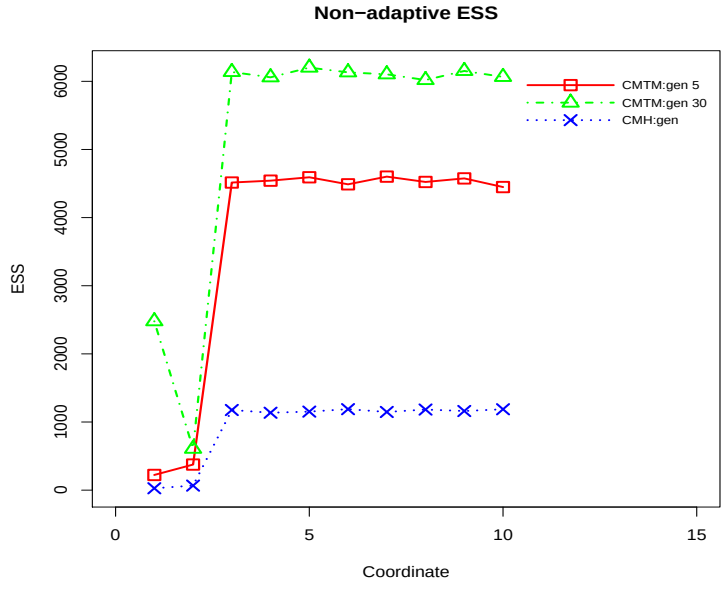
- CMTM and ACMTM.
- Component-wise MH (CMH) requires less computational effort as only one proposal is considered.
- Adaptive CMH - σ's are adjusted up or down to achieve a given component-wise acceptance rate of 44%.
- All chains are run for 10,000 iterations with half used as burn-in.

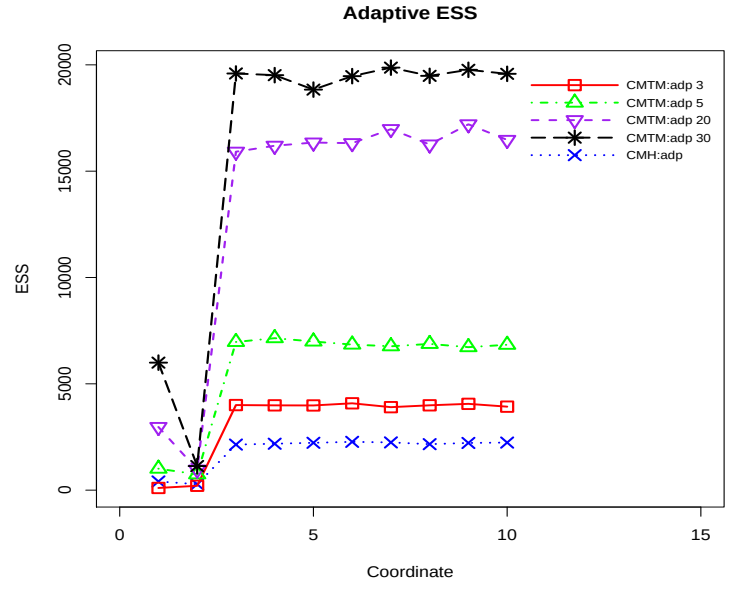
Banana shaped distribution

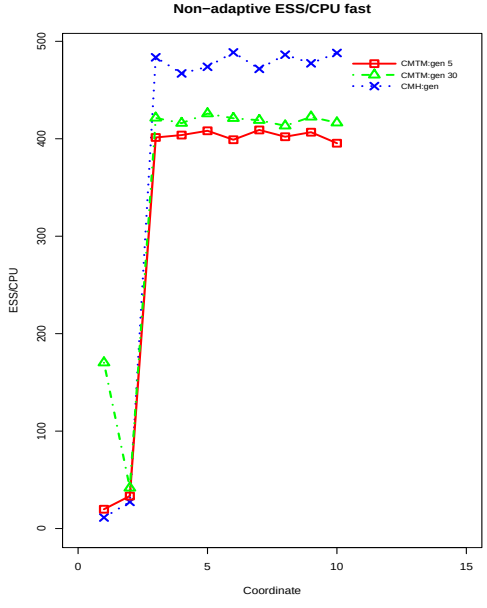
$$f_B(x_1, x_2, \ldots, x_d) \propto exp[-x_1^2/200 - \frac{1}{2}(x_2 + Bx_1^2 - 100B)^2 - \\ - \frac{1}{2}(x_3^2 + x_4^2 + \ldots + x_d^2)].$$

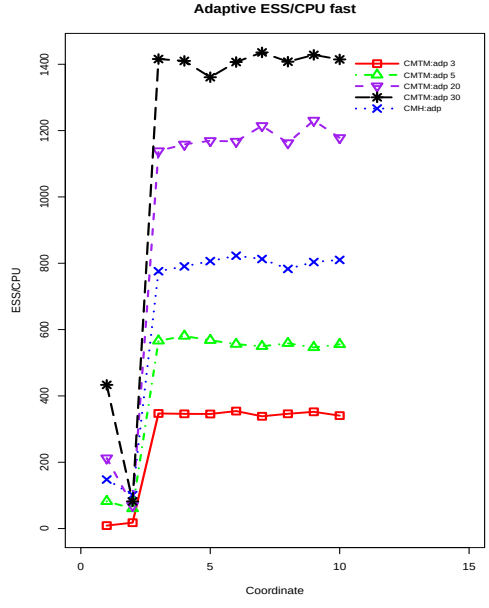
Set $B = 0.01$ and $d = 10$ and the starting $\sigma_j^{(k)}$'s equal to $0.1, 0.2, 0.4, 0.8$ and 1.6

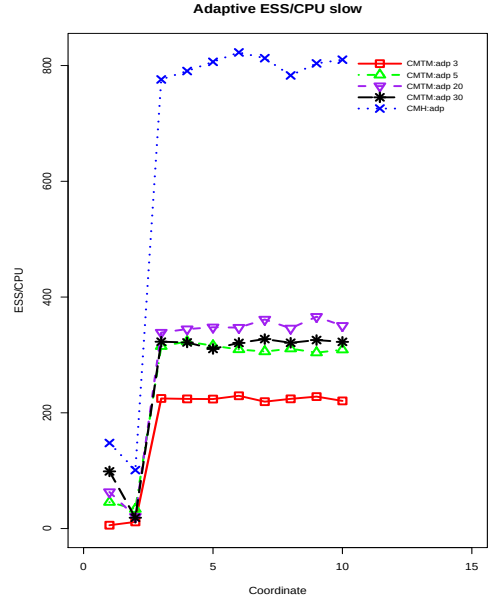
Banana shaped distribution











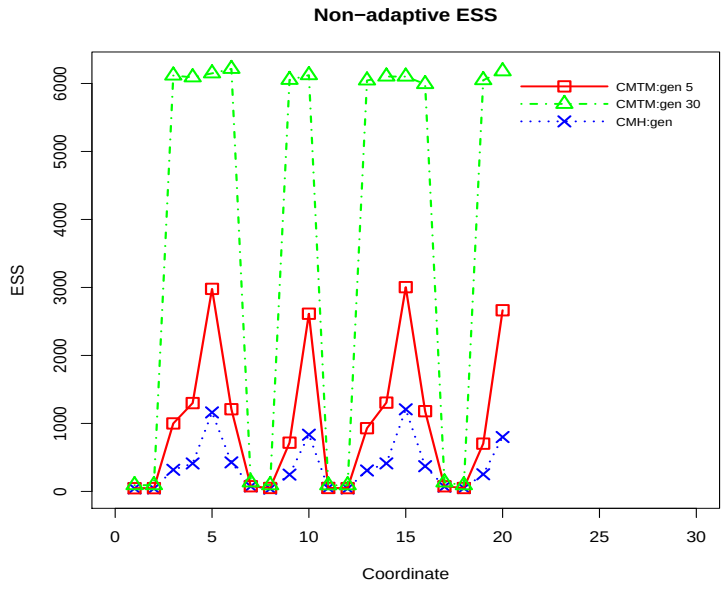
Mixture of 20-dim Gaussians

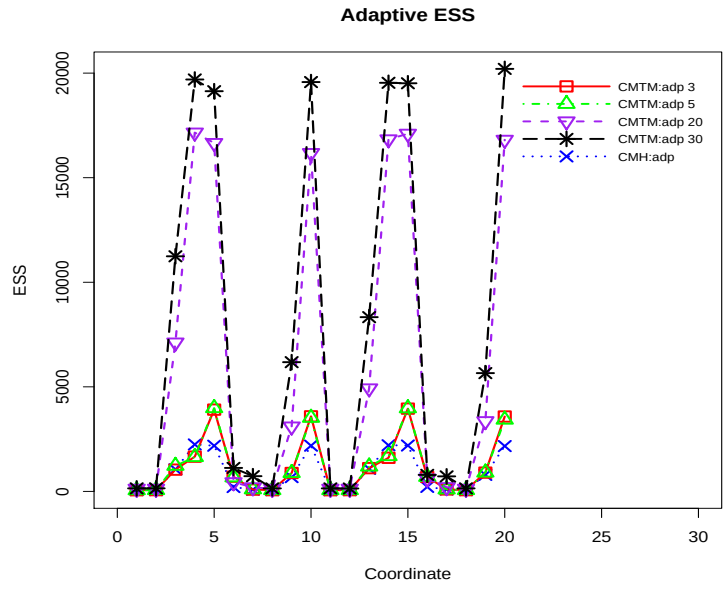
$$0.5N_{20}(\mu_1, \Sigma_1) + 0.5N_{20}(\mu_2, \Sigma_2)$$

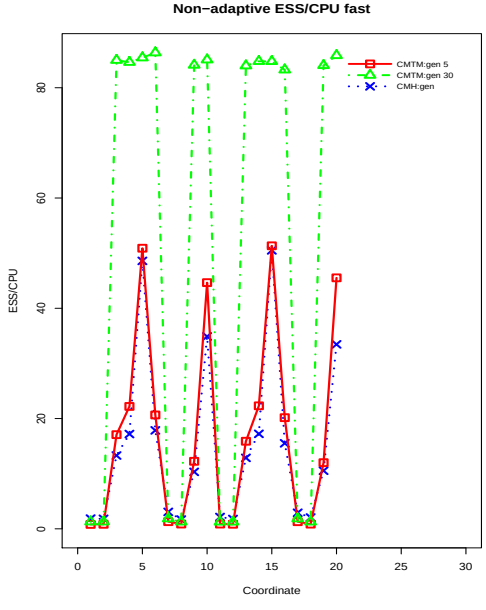
where

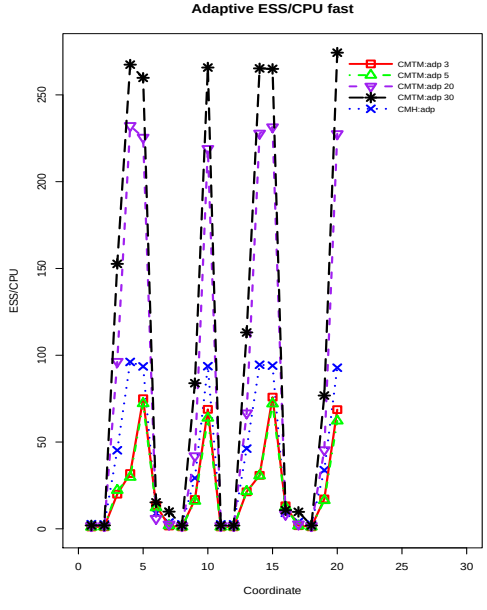
$$\begin{aligned}
\mu_1 =& (5, 5, 0, 0, 0, 0, 10, 15, 0, 0, 5, 5, 0, 0, 0, 0, 10, 15, 0, 0), \\
\mu_2 =& (10, 10, 0, 0, 0, 0, 7, 20, 0, 0, 10, 10, 0, 0, 0, 0, 7, 20, 0, 0), \\
\Sigma_1 =& \text{diag}(16.00, 16.00, 0.25, 4.00, 1.00, 0.01, 9.00, 16.00, 9.00, \\
& 0.01, 16.00, 16.00, 0.25, 4.00, 1.00, 0.01, 9.00, 16.00, 9.00, 0.01), \\
\Sigma_2 =& \text{diag}(16.00, 16.00, 6.25, 4.00, 1.00, 4.41, 9.00, 16.00, 0.25, \\
& 0.01, 16.00, 16.00, 6.25, 4.00, 1.00, 4.41, 9.00, 16.00, 0.25, 0.01).
\end{aligned}$$

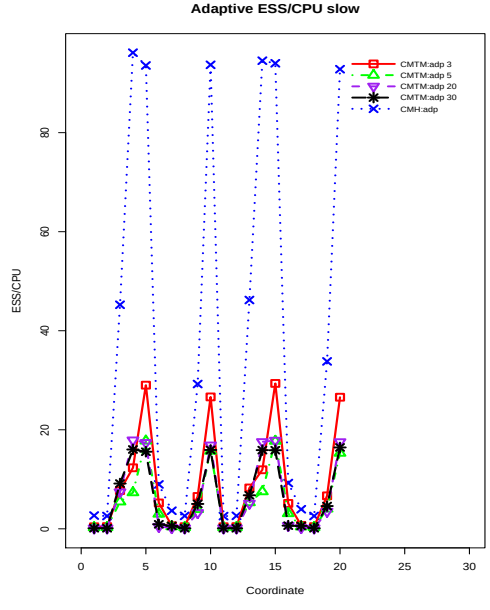
Mixture of 20-dim Gaussians











Conclusions and Future Work

- Comparisons with CMH and other adaptive MCMC samplers show that ACMTM is a solid bet.
- The multiple proposal kernel is expensive in general. Can we leverage the multiple proposals for calculation of fewer likelihood terms?
- Can we parallelize CMTM?
- Some components updates do not require multiple proposals. Can we identify them reliably?

Merçi beaucoup!

Collaborators:

- Evgeny Levi, PhD student
- Jinyoung Yang, PhD student
- Jeffrey Rosenthal

Paper available on my website:
`www.utstat.toronto.edu/craiu/Papers/`