Discussion: "Bayesian Optimization for Adaptive MCMC"

David Dunson

Department of Statistical Science, Duke University

January 3, 2011

What I'd Like to See in Adaptive MCMC Algorithms

- Algorithms are not overly complicated (to program, to describe, etc)

What I'd Like to See in Adaptive MCMC Algorithms

- Algorithms are not overly complicated (to program, to describe, etc)
- Very general purpose & completely automate the tuning process instead of just introducing new tuning parameters

What I'd Like to See in Adaptive MCMC Algorithms

- Algorithms are not overly complicated (to program, to describe, etc)
- Very general purpose & completely automate the tuning process instead of just introducing new tuning parameters
- Provide a tool that can replace MCMC in broad settings & substantially improve computational efficiency

What I'd Like to See in Adaptive MCMC Algorithms

- Algorithms are not overly complicated (to program, to describe, etc)
- Very general purpose & completely automate the tuning process instead of just introducing new tuning parameters
- Provide a tool that can replace MCMC in broad settings & <u>substantially</u> improve computational efficiency
- Lead to quantifiable theoretical gains in efficiency - not as interesting if only seems to do better in a narrow problem

Simplified Overview of MHF Algorithm

- Approach for automated tuning parameter choice in Metropolis-Hastings (MH)

Simplified Overview of MHF Algorithm

- Approach for automated tuning parameter choice in Metropolis-Hastings (MH)
- θ = tuning parameters in MH proposal, $h(\theta)$ = objective fn

Simplified Overview of MHF Algorithm

- Approach for automated tuning parameter choice in Metropolis-Hastings (MH)
- θ = tuning parameters in MH proposal, $h(\theta)$ = objective fn
- $h(\theta)$ high if $\theta \rightarrow$ low area under autocorrelation (AC) fn up to specified lag

Simplified Overview of MHF Algorithm

- Approach for automated tuning parameter choice in Metropolis-Hastings (MH)
- θ = tuning parameters in MH proposal, $h(\theta)$ = objective fn
- $h(\theta)$ high if $\theta \rightarrow$ low area under autocorrelation (AC) fn up to specified lag
- Assign h GP prior, run short chain for given θ_i value to obtain error prone measurement of $h(\theta_i)$ & choose θ_{i+1} to max acquistion fn (*relies on GP predictive having simple form*)

Simplified Overview of MHF Algorithm

- Approach for automated tuning parameter choice in Metropolis-Hastings (MH)
- θ = tuning parameters in MH proposal, $h(\theta)$ = objective fn
- $h(\theta)$ high if $\theta \rightarrow$ low area under autocorrelation (AC) fn up to specified lag
- Assign h GP prior, run short chain for given θ_i value to obtain error prone measurement of $h(\theta_i)$ & choose θ_{i+1} to max acquistion fn (*relies on GP predictive having simple form*)
- Repeated for I θ_i values & MH transition kernel = mixture over θ_i values (*weighted by exponentiated GP objective fn*)

Comments on the GP - Part I

- Clever to use GP "emulator" for the unknown objective fn

Comments on the GP - Part I

- Clever to use GP "emulator" for the unknown objective fn
- Closely related to GP emulators used for design of computer experiments - they also use adaptive design for choice of the next points

Comments on the GP - Part I

- Clever to use GP "emulator" for the unknown objective fn
- Closely related to GP emulators used for design of computer experiments - they also use adaptive design for choice of the next points
- MHF propose to let $z_i = h(\theta_i) + \epsilon_i$ with $\epsilon_i \sim N(0, \sigma_\eta^2)$, with z_i based on an empirical estimate of the AC fn

Comments on the GP - Part I

- Clever to use GP "emulator" for the unknown objective fn
- Closely related to GP emulators used for design of computer experiments - they also use adaptive design for choice of the next points
- MHF propose to let $z_i = h(\theta_i) + \epsilon_i$ with $\epsilon_i \sim N(0, \sigma_\eta^2)$, with z_i based on an empirical estimate of the AC fn
- How to choose σ_η^2? This <u>tuning</u> parameter may be important

Comments on the GP - Part I

- Clever to use GP "emulator" for the unknown objective fn
- Closely related to GP emulators used for design of computer experiments - they also use adaptive design for choice of the next points
- MHF propose to let $z_i = h(\theta_i) + \epsilon_i$ with $\epsilon_i \sim N(0, \sigma_\eta^2)$, with z_i based on an empirical estimate of the AC fn
- How to choose σ_η^2? This <u>tuning</u> parameter may be important
- Seems inefficient to use separate empirical estimates for AC at each lag & for each θ_i value

Comments on the GP - Part II

- As an alternative, could place a GP prior on the AC fn indexed by θ

Comments on the GP - Part II

- As an alternative, could place a GP prior on the AC fn indexed by θ
- This could provide smoothing over lags and borrowing of information for similar θ values

Comments on the GP - Part II

- As an alternative, could place a GP prior on the AC fn indexed by θ
- This could provide smoothing over lags and borrowing of information for similar θ values
- One could easily obtain an induced posterior for summaries such as the area under the AC fn

Comments on the GP - Part II

- As an alternative, could place a GP prior on the AC fn indexed by θ
- This could provide smoothing over lags and borrowing of information for similar θ values
- One could easily obtain an induced posterior for summaries such as the area under the AC fn
- Potentially, one could further improve efficiency by parameterizing the AC fn and/or including monotone decreasing constraints

Comments on the GP - Part II

- As an alternative, could place a GP prior on the AC fn indexed by θ
- This could provide smoothing over lags and borrowing of information for similar θ values
- One could easily obtain an induced posterior for summaries such as the area under the AC fn
- Potentially, one could further improve efficiency by parameterizing the AC fn and/or including monotone decreasing constraints
- Monotonicity constraints may remove simple form of predictive BUT can get around this using isotonic regression transformations as in Dunson & Neelon (03)

Comments on the GP - Part III

- Well known that GP computation bogs down as the number of evaluation points increases

Comments on the GP - Part III

- Well known that GP computation bogs down as the number of evaluation points increases
- $O(i^3)$ computation involved in the matrix inversion

Comments on the GP - Part III

- Well known that GP computation bogs down as the number of evaluation points increases
- $O(i^3)$ computation involved in the matrix inversion
- Huge concern with MHF algorithm, as calculating GP predictive involves repeated calculation of such inverses with dimension increasing as sampling proceeds

Comments on the GP - Part III

- Well known that GP computation bogs down as the number of evaluation points increases
- $O(i^3)$ computation involved in the matrix inversion
- Huge concern with MHF algorithm, as calculating GP predictive involves repeated calculation of such inverses with dimension increasing as sampling proceeds
- Additional computation offset by increased efficiency in selecting good tuning parameters?

Comments on the GP - Part III

- Well known that GP computation bogs down as the number of evaluation points increases
- $O(i^3)$ computation involved in the matrix inversion
- Huge concern with MHF algorithm, as calculating GP predictive involves repeated calculation of such inverses with dimension increasing as sampling proceeds
- Additional computation offset by increased efficiency in selecting good tuning parameters?
- Example has only two tuning parameters - run few enough samples to avoid "bogging down" in this case but not in higher dims?

Comments on the GP - Part IV

- GP covariance used has a separate hyperparameter for each MH tuning parameter - how to choose these & does it impact the performance?

Comments on the GP - Part IV

- GP covariance used has a separate hyperparameter for each MH tuning parameter - how to choose these & does it impact the performance?
- Efficient computation for precisions in the GP covariance is notoriously difficult - may require MH with automated tuning!

Comments on the GP - Part IV

- GP covariance used has a separate hyperparameter for each MH tuning parameter - how to choose these & does it impact the performance?
- Efficient computation for precisions in the GP covariance is notoriously difficult - may require MH with automated tuning!
- Potentially tricks for efficient computation in GP regression can be borrowed - subset of regressors, predictive process, random projections, etc

Comments on the GP - Part IV

- GP covariance used has a separate hyperparameter for each MH tuning parameter - how to choose these & does it impact the performance?
- Efficient computation for precisions in the GP covariance is notoriously difficult - may require MH with automated tuning!
- Potentially tricks for efficient computation in GP regression can be borrowed - subset of regressors, predictive process, random projections, etc
- Possibility: focus on $h(\theta)$ for $\theta \in \Theta_i$, with support narrowed as sampler is run & unpromising regions of the tuning parameter space are ruled out

Acquisition Function & Exploring Tuning Parameter Space

- Based on the GP predictive for $h(\cdot)$, the algorithm chooses an optimal new θ_{i+1} to maximize an acquisition fn

Acquisition Function & Exploring Tuning Parameter Space

- Based on the GP predictive for $h(\cdot)$, the algorithm chooses an optimal new θ_{i+1} to maximize an acquisition fn
- Acquisition fn is chosen so that $h(\theta_{i+1})$ tends to have high variance and/or a high expected value

Acquisition Function & Exploring Tuning Parameter Space

- Based on the GP predictive for $h(\cdot)$, the algorithm chooses an optimal new θ_{i+1} to maximize an acquisition fn
- Acquisition fn is chosen so that $h(\theta_{i+1})$ tends to have high variance and/or a high expected value
- Seems reasonable but I wonder about the practical performance in finding good tuning parameter values when there are more than a few tuning parameters

Acquisition Function & Exploring Tuning Parameter Space

- Based on the GP predictive for $h(\cdot)$, the algorithm chooses an optimal new θ_{i+1} to maximize an acquisition fn
- Acquisition fn is chosen so that $h(\theta_{i+1})$ tends to have high variance and/or a high expected value
- Seems reasonable but I wonder about the practical performance in finding good tuning parameter values when there are more than a few tuning parameters
- MH algorithms for complex models having few tuning parameters may have insufficiently flexible kernels

Comments on the Sampling Phase

- At the end of the adaptation phase, they have predictive of $h(\theta)$ as well as short MCMC chains for different θ values

Comments on the Sampling Phase

- At the end of the adaptation phase, they have predictive of $h(\theta)$ as well as short MCMC chains for different θ values
- Use these components to build a MH transition kernel as mixture over the θ_is

Comments on the Sampling Phase

- At the end of the adaptation phase, they have predictive of $h(\theta)$ as well as short MCMC chains for different θ values
- Use these components to build a MH transition kernel as mixture over the θ_is
- Mixture weights are normalized exponentiated obj fn $h(\cdot)$ - seems somewhat ad hoc & may not provide optimal weights

Summary Comments

- Congratulations to the authors on a thought-provoking paper

Summary Comments

- Congratulations to the authors on a thought-provoking paper
- Step in the right direction but many questions remain

Summary Comments

- Congratulations to the authors on a thought-provoking paper
- Step in the right direction but many questions remain
- One set of tuning parameters is replaced by another set

Summary Comments

- Congratulations to the authors on a thought-provoking paper
- Step in the right direction but many questions remain
- One set of tuning parameters is replaced by another set
- Multi-stage form of algorithm involving repeated chains, GPs, etc seems computationally very intensive

Summary Comments

- Congratulations to the authors on a thought-provoking paper
- Step in the right direction but many questions remain
- One set of tuning parameters is replaced by another set
- Multi-stage form of algorithm involving repeated chains, GPs, etc seems computationally very intensive
- Needs more theory & applications assessing efficiency in broad problems