Notes on "MCMC Techniques for Parameter Estimation of ODE Based Models in Systems Biology"

Maya Watanabe
Christina Catlett

Summer 2020

1 Introduction

- This paper seeks to compare MCMC algorithms for fitting parameters of nonlinear ODE systems using data: Metropolis-Hastings, parallel tempering, adaptive, and parallel adaptive
- Optimization algs. are often used, however, due to non-linearity, there is no way to guarantee that a global max is found vs. local max
- In theory, averages taken over the MCMC process converge to expected values, but assessing convergence in practice is more difficult (ergodicity)
-
  - Speed of converged varies with chosen algorithm
  - Parallel and Adaptive MCMCs aim to enhance convergence speed: Parallel MCMC covers a larger postion of the parameter space as one time
- Goal: to compare Metropolis-Hastings and adaptive MCMC to parallel tempering and parallel adaptive MCMC using specifically designed informative priors

2 Methods

2.1 ODE Models

- Model 1: 8 variables + 22 parameters
  - Positive and negative feedback loops in the MAPK phosphorylation cascade
  - Uses Michalis-Menten and Hill kinetics
- Model 2: 33 variables + 26 parameters
  - Representation of ERK phosphorylation in CFU-E
  - Based on mass action
- Model 3: 10 variables + 13 parameters
  - Describes Smad sign signaling
  - Uses Michalis-Menten and Hill kinetics
- Model 4: 9 variables + 18 parameters
  - Describes role of receptor endocytosis in the feedback mechanism involved in insulin resistance in T2D
  - Includes measured data under 9 experimental conditions

- Model 5: 14 variables + 22 parameters
  - Represents a signaling pathway of the Hodgkin and Primary Mediastinal B-Cell Lymphome induced by IL13

2.2 Simulation of Data

- Data for first three models simulated by solving ODE systems (ICs at steady state values) at 17 values of t (time): $t = 0, .25, .5, 1.5, 3, 4.5, 6, 7.5, 9, 10.5, 12, 18, 24, 36, 48, 100$
  - *How did they know what the steady state of each system was?*
- They added different levels of Gaussian noise at each time point of the ODE predictions:

$$\epsilon_{s_i} = |max_t x_s(t) - min_t(T)|U \text{ where } U \sim N(0, \tau) \text{ and } \tau = 0.001, 0.005, 0.01, 0.05, 0.1, 0.15, 0.25$$

- The noise added, ϵ_{s_i} denotes a percentage of the range of measurement, hence taking the range of the ODE's output multiplied by U
- Three replicate measurements taken *How do these play in? How are they formatted?*

2.3 Likelihood Model

- Experimental data is denoted by:

$$D = \{y_{s,t,r,c} | s = 1, ..., n; t = 1, ..., T; r = 1, ..., R, c = 1, ..., C\}$$

  where n is the number of species, T is the length of the time series, R is the number of replicate measurements per time point (3), and C is the number of experimental conditions.
- Because of, as described above, the assumption of Gaussian noise, the likelihood function of the data given the set of parameters K, $P(D|K, \sigma)$ is normal [1] and thus the log-likelihood is given as:

$$log(p(D|k, \sigma)) \propto \sum_{c=1}^{C} \sum_{s=1}^{n} \sum_{t=1}^{T} \sum_{r=1}^{R} (-\frac{(y_{s,t,r,c} - x_{s,c}(t))^2}{2\sigma^2_{s,t,c}})$$

  where $x_{s,c}(t)$ is the output of the ODE model for species s at time t under condition c. $\sigma^2_{s,t,c}$ is the measurement noise, and the vector of all $\sigma^2_{s,t,c}$ is known as σ.
- /emphWhat do they mean by "vector of all $\sigma^2_{s,t,c}$"?
- Since the number of replicate measurements R for each species is low (¡5), they impose an inverse gamma distribution prior for $\sigma^2_{s,t,c}$
  - *We hypothesize that this is because of the low number of samples and thus 'unknown-ness' of the true distribution and standard deviation of the data*

$$\sigma^2_{s,t,c} \sim IG(\alpha, \beta)$$

- The marginal likelihood $(D|K, \sigma)p(\sigma)d\sigma$ can be solved for analytically, again taking to log for summation vs multuplication purposes:

$$\log(p(D|K)) = \sum_{c=1}^{C} \sum_{s=1}^{n} \sum_{t=1}^{t} \sum_{r=1}^{R} -(\alpha + \tfrac{1}{2}) log(1 + \tfrac{1}{2\beta}(y_{s,t,r,c} - x_{s,c}(t))^2)$$

  - *What are the purposes of the likelihood vs the marginal likelihood functions in the MCMC? We know from Baye's Theorem how we use the likelihood to find th posterior, but what about the marginal likelihood?*

[1] This is also described in the astrostats tutorial

2.4 An Informative Prior for Kinetic Rate Parameters

- Based on prior knowledge regarding kinetic rate parameters, they suppose that they follow a log-normal distribution: $\log(k)\ N(\mu, \rho^2)$
- $\mu and \rho$ are free parameters, that is they are chosen based on theory or experimental data
- These parameters are estimated using databases: Brenda, BioModels

3 Markov Chain Monte Carlo (MCMC)

3.1 General

- Goal: Given data, D, estimate parameters, K while circumventing non-liner optimization
- Acceptance Criteria: using a Gaussian transition kernel q in log-parameter space, a candiate parameter set K' is accepted with probability:

$$a = \min(1, \frac{p(D|K')p(K')}{p(D|K)p(K)})$$

  - Accept a set of parameters K' with probability 1 if the likelihood*prior of K' is greater than the likelihood*prior of the current K
  - Otherwise, accept K' with a probability of the ratio of likelihood*prior of K' over likelihood*prior of the current K
  - *I still feel like we don't really know where this ratio is coming from:*
    - We want to accept the set of parameters that is most likely, but we don't want to outright reject those K' whose unnorm posterior ¡ the current K's unnorm posterior
- Transition kernel q can be univariate or multivariate
  - *What exactly is a transition kernel?* Is it just the covariance matrix? If not, how is it related to the covariance matrix? (Reasoning: in our MH alg, we set up the covariance matrix to be just a diagonal, thus univariate)
  - MH and parallel tempering rely on a univariate transition kernel (i.e. a single rate parameter is selected first and then the transition kernel is applied)
  - Adaptive MCMC uses a multivariate Gaussian transition kernel in parameter space

3.2 Parallel MCMC

- This paper is asking: Do parallel MCMCs offer an advantage for parameter estimation?
- Best guess: Parallel MCMC consists of running multiple synced MCMC chains, not only selecting the most likely move from one chain, but all chains (swap operations). Each chain has an "energy" (the log-likelihood), which is divided by a "temperature", T. A large T predicts that MCMC 'moves' (accepting new parameters) become more likely. This makes sense because dividing by a large number favors lower probabilities (log is greater in magnitude), and thus moves from low probabilities are more likely than staying at the current params.
- *Where does T come from?*

3.3 Convergence

- To compare MCMC methods, we need to assess and compare each method's convergence to the true posterior distribution (when the alg. can start drawing samples from the target distribution, i.e. the steady state of the Markov chain)
- Trace plots of the marginal log-likelihoods can be used as visual indicators, but *Geweke's test* is used to more computationally compare convergences
- Geweke's test
  - Compare the means of the first 10% and last 50% of the (single) Markov chain
  - If both means are equal, the MCMC converged to a stationary distribution
  - Geweke's test statistic should be close to 0
  - Test can be applied sequentially: for a chain divided into m segments, the z-score can be calculated leaving out the first n segments. Iterating n, it can be seen where the z-score, and hence the chain converge.
- *If multiple methods converge, can we use Geweke's test statistic to determine the better of the two convergences? Maybe on the basis of time it took to converge?*
- Gelman-Rubin convergence diagnostic (for parallel MCMC)
  - This diagnostic compares *within-chain* variance to *between-chain* variance
  - The resulting factor indicates lack of convergence if it substantially larger than 1
- These convergence tests are applied to individual parameters. Usually their results are inspected manually, and more iterations are run if necessary, but this doesn't work systematically. So, the paper designates the *optimal number of burn-in iterations* by finding the first iteration where Geweke's Test, Gelman-Rubin statistic (for parallel chains) indicate convergence
  - This number can be different for the different parameters estimated, so the paper used the max number of burn-in required among all params for every param.
- Data was "thinned" at every 100th point
  - *Thinning: discarding every 100th sample to make the resulting data set smaller and more manageable*

3.4 Effective Sample Size

- ESS: number of "effectively independent" draws from the posterior distribution that the Markov Chain is equivalent to
- *According to more research, the ESS is a measure of how complete the posterior is; a low ESS implies a poor posterior. Additionally the amount by which autocorrelation (non-independence) within the chains increases uncertainty in estimates (ie uncertainty in the posterior or a poor posterior) is measured by ESS*
- It is given by:

$$ESS = T\frac{\sigma^2}{\rho}$$

  where T denotes the number of drawn samples, σ^2 the variance, and σ, the autocorrelation.
  - *What is the range of values for autocorrelation? (How does this impact relationship)*

3.5 Evaluation of Estimated Parameters

- The fraction of true parameters falling into the 95% Bayesian credible interval of the MCMC samples, that is the posterior distribution of the parameter resulting from the MCMC, was used to evaluate the quality of parameter estimations

4 Results

4.1 Convergence Diagnostics

- Different MCMC strategies were run with uniformative priors (both for noise, parameters) to make comparison more accurate
- Figure 3: Insight into convergence times, accuracy; Single adaptive MCMC outperformed MH in both accuracy and convergence time
- Figure 4: Execution times for different MCMC algs: Metropolis Hastings generally was fastest, parallel adaptive was slowest; all models took hours to run (1.5- 16)
- Analyzed all the convergence of all the MCMC algs using Geweke and Gelman-Rubing stats
- Table 1: Convergence analysis - convergence occurred in all parallel tempering and single adaptive MCMC, but did not occur in multi-chain adaptive. No observed dependency between noise level and convergence failure
  - Authors note the limitations to the test statistics - mostly that they can verify some but not all conditions for convergence
  - Also it is interesting to note that some but not all parameters may fail to converge

4.2 Influence of Informative Priors on Convergence

- Goal: Do informative priors for params, noise influence convergence?
- Analysis performed for noise levels 5, 10, 15, and 25% because informative priors help negate noise, especially when noise is significant
- Table 2: No clear influence of informative priors on convergence for MH, but if an informative prior is used for the parameters in single adaptive MCMC, all parameters were shown to converge.

4.3 Effective Sample Size

4.3.1 Parallel MCMC Yields More Effective Sampling

- Comparison of ESS of dif. MCMC algs. with uniformative priors and noise level of 10% indicates the most effective sampling by parallel tempering for models 1, 4, and 5. For models 2 and 3, parallel adaptive seems to be most efficient.
- Parallel MCMC seem to be more advantageous than single chain methods

4.3.2 Influence of Informative Priors on Effective Sample Size

- Goal: Do informative priors enhance ESS?
- An advantage was only observed for adpative MCMC (Figure 6), but not for one specific type of prior

4.4 Parameter Estimates

- Compared the fraction of true model parameters lying within the 95% Bayesian credible interval of the posterior distribution (i.e. how well do the true parameters fall into the computed posterior distribution: fraction close to 1 indicates posterior is good, close to 0 indicates posterior is poor)
- *How is the Bayesian credible interval calculated?*
- Results indicate benefits for the parallel adaptive MCMC model for all noise levels and models
  - Parallel adaptive model is highly robust against noise; the fraction of parameters within 95% Bayesian credible interval was not obviously affected by different noise levels
  - Also effective for experimental data (largely within credible interval) even with small number of chains (5) tested
- Parallel tempering showed "unsystematic" dependence on noise level, and covered a lower portion of true model parameters in the Bayesian credible interval
  - Conclusion: Learning the covariance structure of model parameters in parallel adaptive MCMC is beneficial for an "effective exploration of different modes of the posterior distribution" *("exploration of different modes" =? finding global maxima vs. local)*
    - *What does this mean? Specifically, what is 'covariance structure'?*
    - *A guess might be the covariance matrix/kernel from the intro that was used to choose the candidate parameter set K'*
- Figure 8: Shows suggested informative priors could improve parameter estimation while focusing on single chain MCMC methods
  - Incorporating informative priors for both noise and parameters result in a larger fraction of true model parameters in the Bayesian 95% credible interval

5 Conclusion

- We need MCMC methods (full Bayesian inference) to infer structural non-identifiable parameters of biological system ODEs from experimental data, but this can be a costly computation
- Designing a good sampling algorithm involves: sufficiently fast convergence, high ESS, and a high probability to converging to true parameters
- Limitations of MCMC:
  - Computationally costly with many ODEs in the system, many evaluations of the likelihood (can perform approximations to lessen computation)
- In summary: parallel adaptive MCMC + suggested variance and parameter priors seem to be the best approach for ODE models in systems biology because of benefits in fast convergence, ESS, and a high probability of convergence to true parameters.

6 R Code

The R code from the paper used several different MCMC packages to run its models. Though they did not write their own algorithm, the code worked by:

1. Simulating data by using an ODE solver and adding varying, normally distributed noise
2. Estimating priors for models with experimental data from databases, otherwise given
3. Computing the log likelihoods given different combinations of priors: no informative priors, informative priors for either the params or the noise, and both
4. Running chosen MCMC alg

7 Understanding the different MCMC algorithms

We found helpful pseudocode at `https://www.cs.ubc.ca/~nando/540b-2011/projects/8.pdf` and the math behind the algorithm at `https://arxiv.org/pdf/1205.1076.pdf`

7.1 MH MCMC

We have a good understanding of this algorithm from the astrostats tutorial

7.2 Parallel Tempered MCMC

- Input: t_k temperature set; M num of temps; N_{iter} num of iterations per sweep; N_{sweep} num of sweeps per PT
- Output: x_k^i chains of samples

for i = $1...N_{sweep}$\{
for k = $1...M$\}
x_k^i = Metropolis-Hastings($\frac{exp(-energy)}{eachtemp}$ over the whole parameter space, MH random walk sampling, N_{iter}) *on each chain perform MH MCMC and sample from a parameter space defined by the specific temperature in each chain*
}
for i = $1...M-1$\{
Swap x_k^i with x_{k+1}^i with probability $\alpha_k = min(1, \frac{p_k(x_{k+1})p_{k+1}(x_k)}{p_k(x_k)p_{k+1}(x_{k+1})}$ *swap chains of samples between chains with different samples with an acceptance criterion based on the ratio* = $\frac{(energy\ of\ the\ prop.\ param\ at\ cur\ temp)(energy\ of\ cur\ param\ at\ next\ temp)}{energy\ of\ cur\ param\ at\ cur\ temp)(energy\ of\ cand.\ param\ at\ next\ temp}$
}
}

7.3 Parallel Adaptive MCMC

- Input: t_k, initial parameter set; M, the number of parameters; N_{iter} the number of feedback iterations; N_{sweep}, number of sweeps of parallel tempering within an iteration
- Output: t_k, the optimized parameter set

begin \{
for $i = 1...\ N_{iter}$ do \{
Parallel Tempering for N_{sweep} steps
Calculate F: $F(t_k) = \frac{n_{up}(t_k)}{n_{up}(t_k)+n_{down}(t_k)}$
Change k based on making $\Delta F = F(t_k) - F(t_{k+1})$ constant (i.e making the fraction of samples moving upward in temperature linearly increase based on temperature, making each replica equally informative)
}
}