Model Calibration and Uncertainty Propagation for an SIR Model

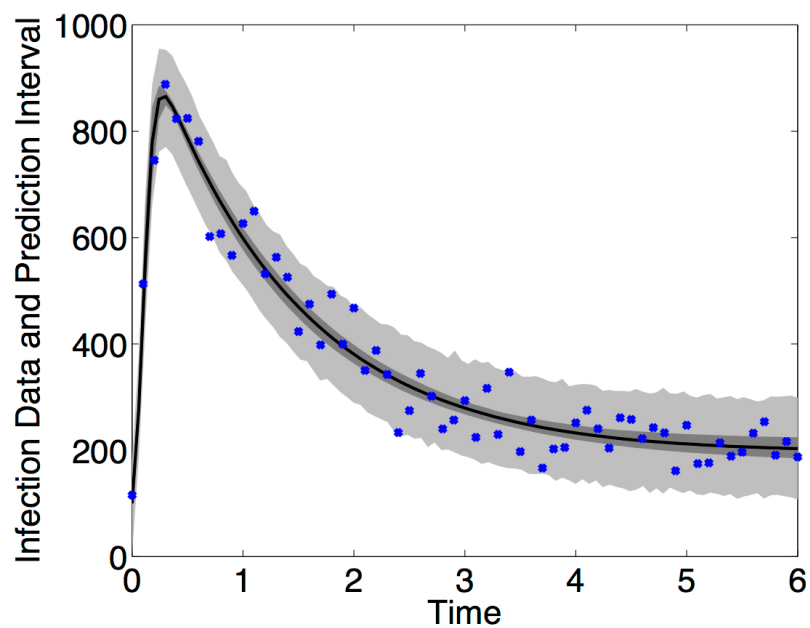
Ralph C. Smith
Department of Mathematics
North Carolina State University

SIR Model

$$\frac{dS}{dt} = \delta N - \delta S - \underline{\gamma k} IS \quad , \; S(0) = S_0$$

$$\frac{dI}{dt} = \underline{\gamma k} IS - (r + \delta) I \quad , \; I(0) = I_0$$

$$\frac{dR}{dt} = rI - \delta R \quad , \; R(0) = R_0$$



SIR Disease Example

SIR Model:

$$\frac{dS}{dt} = \delta N - \delta S - \gamma k I S \quad , \; S(0) = S_0$$ Susceptible

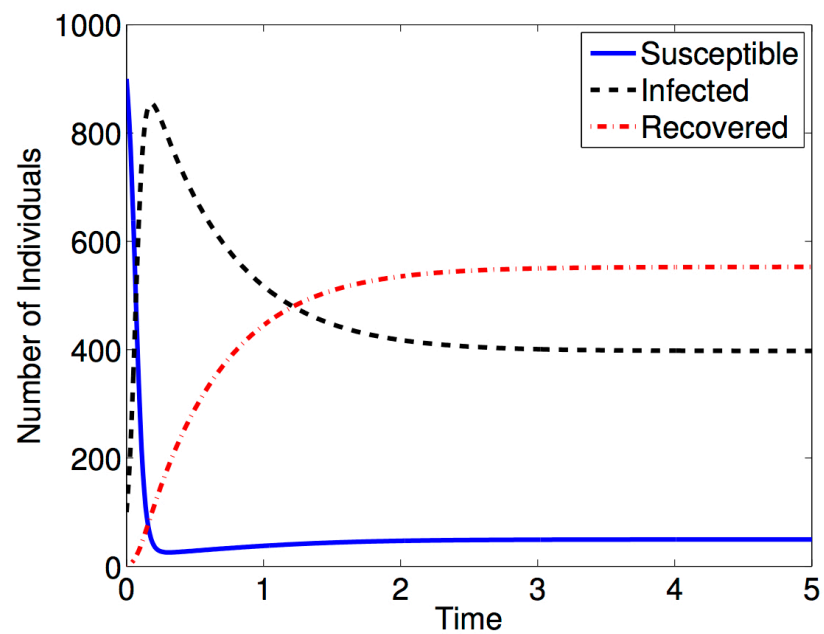
$$\frac{dI}{dt} = \gamma k I S - (r + \delta) I \quad , \; I(0) = I_0$$ Infectious

$$\frac{dR}{dt} = r I - \delta R \quad , \; R(0) = R_0$$ Recovered

Note: Parameter set $q = [\gamma, k, r, \delta]$ is not identifiable

Objectives: Employ Bayesian analysis for

- Parameter selection
- Model calibration
- Uncertainty propagation



Statistical Inference

Goal: The goal in statistical inference is to make conclusions about a phenomenon based on observed data.

Frequentist: Observations made in the past are analyzed with a specified model. Result is regarded as confidence about state of real world.

- Probabilities defined as frequencies with which an event occurs if experiment is repeated several times.
- Parameter Estimation:
  - Relies on estimators derived from different data sets and a specific sampling distribution.
  - Parameters may be unknown but are fixed and deterministic.

Bayesian: Interpretation of probability is subjective and can be updated with new data.

- Parameter Estimation: Parameters are considered to be random variables having associated densities.

Bayesian Model Calibration

Bayes' Theorem:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Bayesian Model Calibration:

- Parameters assumed to be random variables

$$\pi(q|\upsilon) = \frac{\pi(\upsilon|q)\pi_0(q)}{\int_{\mathbb{R}^p} \pi(\upsilon|q)\pi_0(q)dq}$$

Example: Coin Flip

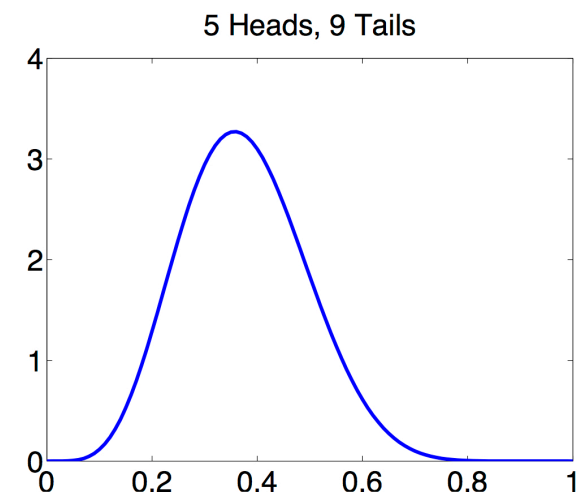
$$\Upsilon_i(\omega) = \begin{cases} 0 & , \quad \omega = T \\ 1 & , \quad \omega = H \end{cases}$$

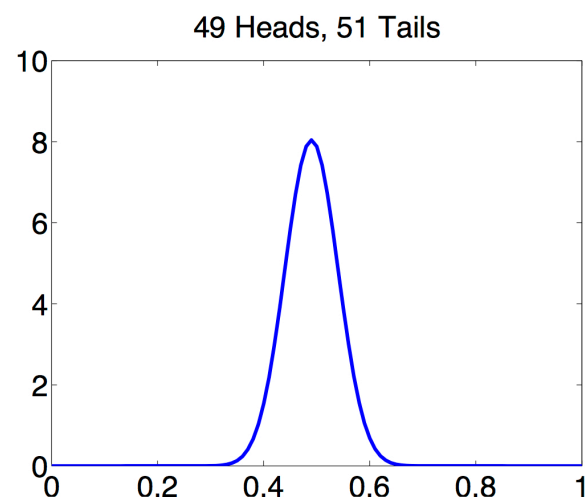
Likelihood:

$$\begin{aligned} \pi(\upsilon|q) &= \prod_{i=1}^{N} q^{\upsilon_i}(1-q)^{1-\upsilon_i} \\ &= q^{\sum \upsilon_i}(1-q)^{N-\sum \upsilon_i} \\ &= q^{N_1}(1-q)^{N_0} \end{aligned}$$

Posterior with Noninformative Prior: $\pi_0(q) = 1$

$$\pi(q|\upsilon) = \frac{q^{N_1}(1-q)^{N_0}}{\int_0^1 q^{N_1}(1-q)^{N_0}dq} = \frac{(N+1)!}{N_0!N_1!}q^{N_1}(1-q)^{N_0}$$



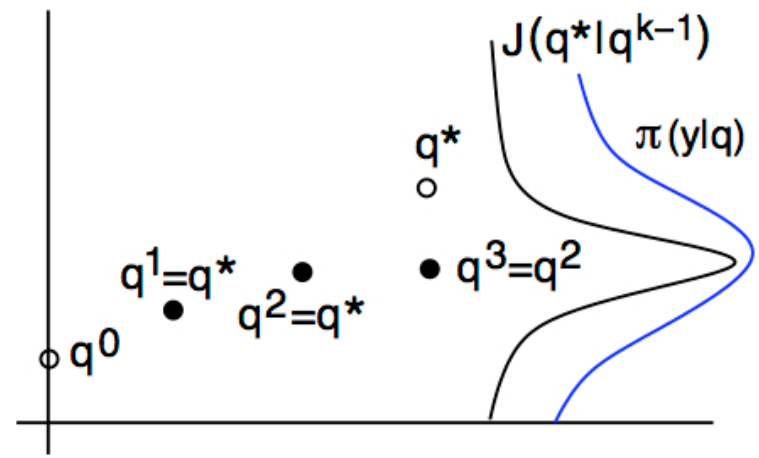


Bayesian Model Calibration

Bayesian Model Calibration:

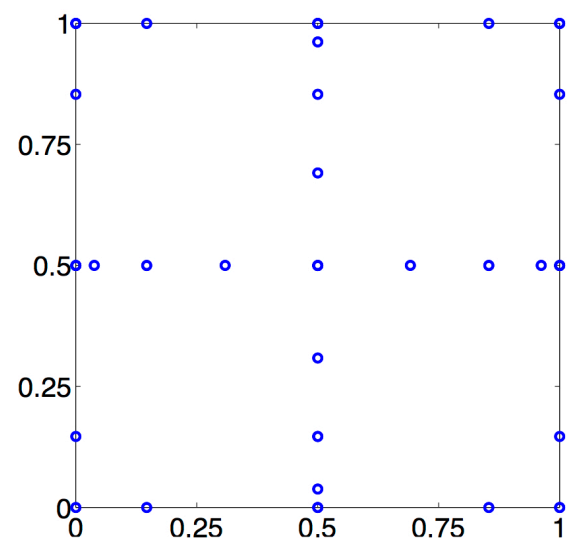
- Parameters considered to be random variables with associated densities.

$$\pi(q|\upsilon) = \frac{\pi(\upsilon|q)\pi_0(q)}{\int_{\mathbb{R}^p} \pi(\upsilon|q)\pi_0(q)dq}$$



Problem:

- Often requires high dimensional integration;
  - e.g., p = 6-23 for HIV model
  - p = hundreds to thousands for some models



Strategies:

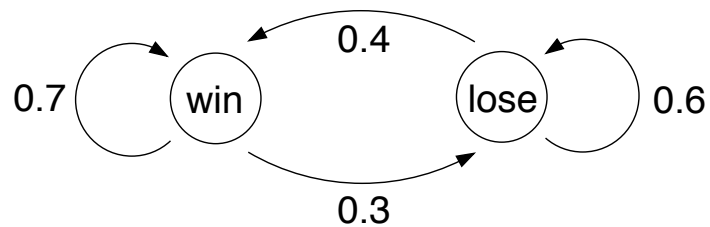
- Sampling methods
- Sparse grid quadrature techniques

Markov Chain Techniques

Markov Chain: Sequence of events where current state depends only on last value.

Baseball: States are S = {win,lose}. Initial state is $p^0 = [0.8, 0.2]$.

- Assume that team which won last game has 70% chance of winning next game and 30% chance of losing next game.
- Assume losing team wins 40% and loses 60% of next games.



- Percentage of teams who win/lose next game given by

$$p^1 = [0.8\ ,\ 0.2] \begin{bmatrix} 0.7 & 0.3 \\ 0.4 & 0.6 \end{bmatrix} = [0.64\ ,\ 0.36]$$

- Question: does the following limit exist?

$$p^n = [0.8\ ,\ 0.2] \begin{bmatrix} 0.7 & 0.3 \\ 0.4 & 0.6 \end{bmatrix}^n$$

Markov Chain Techniques

Baseball Example: Solve constrained relation

$$\pi = \pi P \quad , \quad \sum \pi_i = 1$$

$$\Rightarrow [\pi_{win}\,,\,\pi_{lose}] \begin{bmatrix} 0.7 & 0.3 \\ 0.4 & 0.6 \end{bmatrix} = [\pi_{win}\,,\,\pi_{lose}] \quad , \quad \pi_{win} + \pi_{lose} = 1$$

to obtain

$$\pi = [0.5714\,,\,0.4286]$$

Markov Chain Techniques

Baseball Example: Solve constrained relation

$$\pi = \pi P \quad , \quad \sum \pi_i = 1$$

$$\Rightarrow [\pi_{win}\,,\,\pi_{lose}] \begin{bmatrix} 0.7 & 0.3 \\ 0.4 & 0.6 \end{bmatrix} = [\pi_{win}\,,\,\pi_{lose}] \quad , \quad \pi_{win} + \pi_{lose} = 1$$

to obtain

$$\pi = [0.5714\,,\,0.4286]$$

Alternative: Iterate to compute solution

n	p^n	n	p^n	n	p^n
0	$[0.8000\,,\,0.2000]$	4	$[0.5733\,,\,0.4267]$	8	$[0.5714\,,\,0.4286]$
1	$[0.6400\,,\,0.3600]$	5	$[0.5720\,,\,0.4280]$	9	$[0.5714\,,\,0.4286]$
2	$[0.5920\,,\,0.4080]$	6	$[0.5716\,,\,0.4284]$	10	$[0.5714\,,\,0.4286]$
3	$[0.5776\,,\,0.4224]$	7	$[0.5715\,,\,0.4285]$		

Notes:

- Forms basis for Markov Chain Monte Carlo (MCMC) techniques
- Goal: construct chains whose stationary distribution is the posterior density

Markov Chain Monte Carlo Methods

Strategy: Markov chain simulation used when it is impossible, or computationally prohibitive, to sample q directly from

$$\pi(q|\upsilon) = \frac{\pi(\upsilon|q)\pi_0(q)}{\int_{\mathbb{R}^p} \pi(\upsilon|q)\pi_0(q)dq}$$

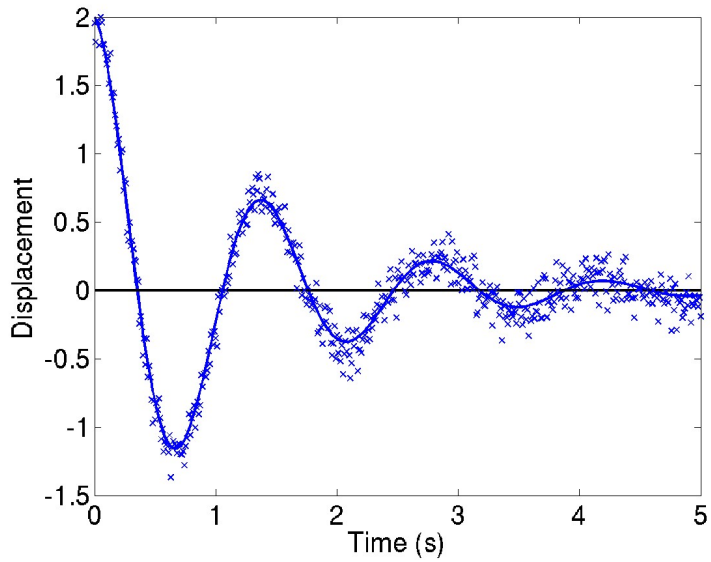
- Create a Markov process whose stationary distribution is $\pi(q|\upsilon)$.

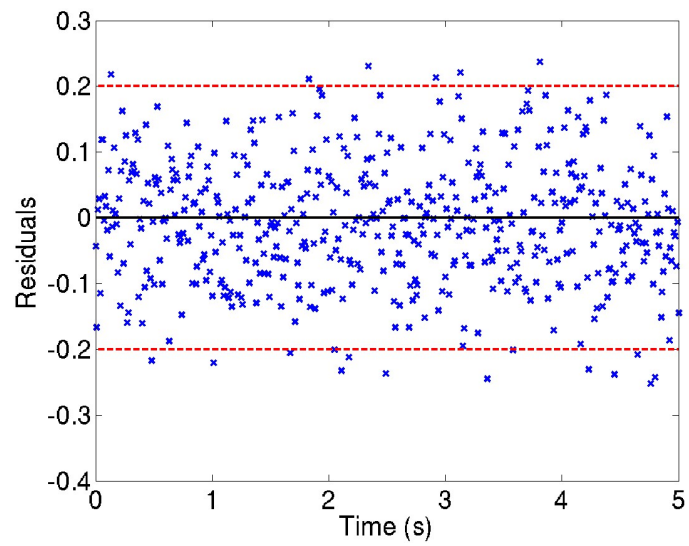
Note:

- In Markov chain theory, we are given a Markov chain, P, and we construct its equilibrium distribution.
- In MCMC theory, we are "given" a distribution and we want to construct a Markov chain that is reversible with respect to it.

Model Calibration Problem

Assumption: Assume that measurement errors are iid and $\varepsilon_i \sim N(0, \sigma^2)$





Likelihood:

$$\pi(\upsilon|q) = L(q, \sigma|\upsilon) = \frac{1}{(2\pi\sigma^2)^{n/2}} e^{-SS_q/2\sigma^2}$$

where

$$SS_q = \sum_{i=1}^{n} [\upsilon_i - f_i(q)]^2$$

is the sum of squares error.

Markov Chain Monte Carlo Methods

General Strategy:

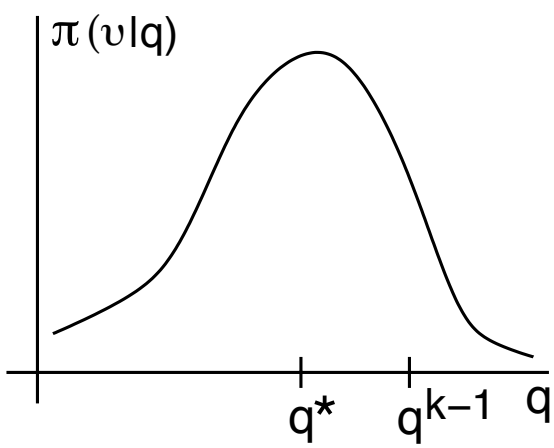
- Current value: $X_{k-1} = q^{k-1}$
- Propose candidate $q^* \sim J(q^*|q^{k-1})$ from proposal (jumping) distribution
- With probability $\alpha(q^*, q^{k-1})$, accept q^*; i.e., $X_k = q^*$
- Otherwise, stay where you are: $X_k = q^{k-1}$

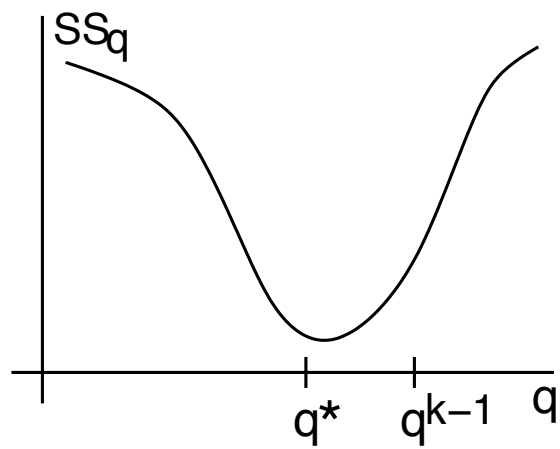
Intuition: Recall that

$$\pi(q|\upsilon) = \frac{\pi(\upsilon|q)\pi_0(q)}{\int_{\mathbb{R}^p} \pi(\upsilon|q)\pi_0(q)dq}$$

where

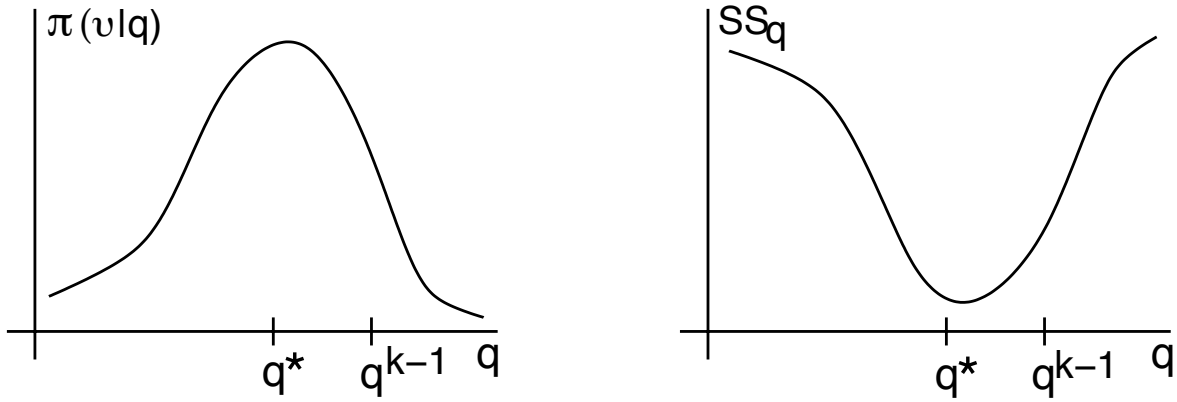
$$\pi(\upsilon|q) = \frac{1}{(2\pi\sigma^2)^{n/2}} e^{-\sum_{i=1}^n [\upsilon_i - f_i(q)]^2/2\sigma^2} = \frac{1}{(2\pi\sigma^2)^{n/2}} e^{-SS_q/2\sigma^2}$$





Markov Chain Monte Carlo Methods

Intuition:

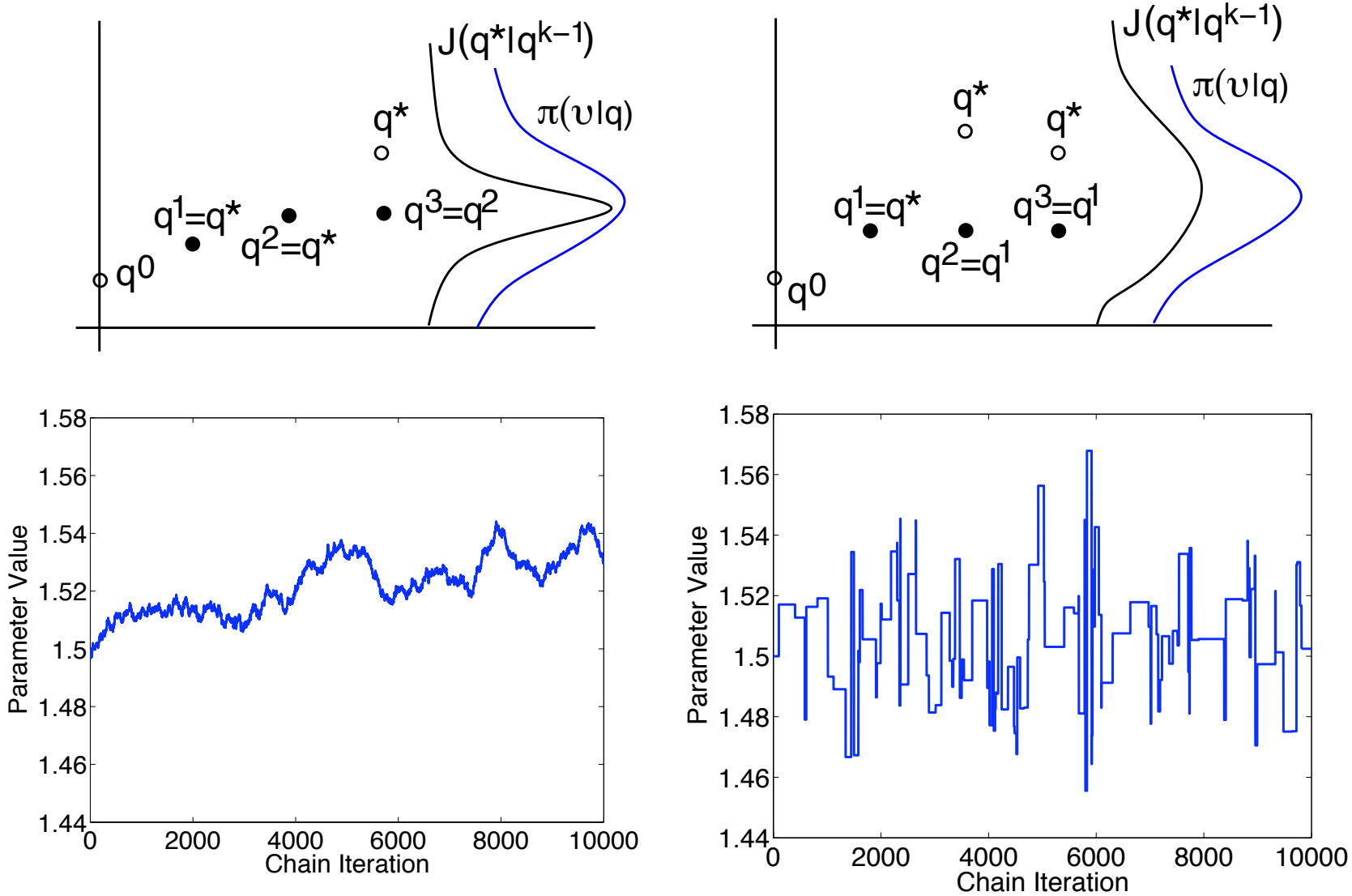


- Consider $r(q^*|q^{k-1}) = \frac{\pi(q^*|\upsilon)}{\pi(q^{k-1}|\upsilon)} = \frac{\pi(\upsilon|q^*)\pi_0(q^*)}{\pi(\upsilon|q^{k-1})\pi_0(q^{k-1})}$
  - If $r < 1 \Leftrightarrow \pi(\upsilon|q^*) < \pi(\upsilon|q^{k-1})$, accept with probability $\alpha = r$
  - If $r > 1$, accept with probability $\alpha = 1$

Note: Narrower proposal distribution yields higher probability of acceptance.

Markov Chain Monte Carlo Methods

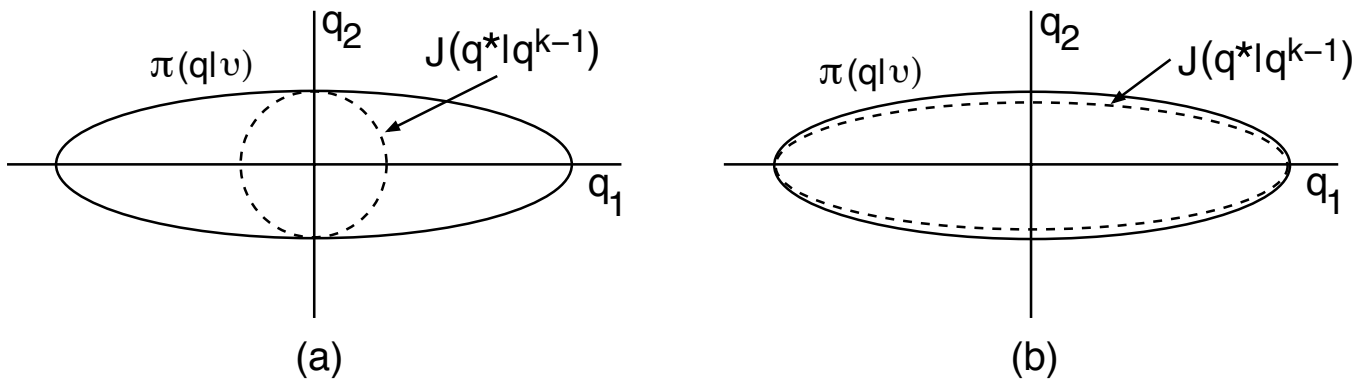
Note: Narrower proposal distribution yields higher probability of acceptance.



Proposal Distribution

Proposal Distribution: Significantly affects mixing

- Too wide: Too many points rejected and chain stays still for long periods;
- Too narrow: Acceptance ratio is high but algorithm is slow to explore parameter space
- Ideally, it should have similar "shape" to posterior distribution.



Problem:

- Anisotropic posterior, isotropic proposal;
- Efficiency nonuniform for different parameters

Result:

- Recovers efficiency of univariate case

Proposal Distribution

Proposal Distribution: Two basic approaches

- Choose a fixed proposal function
  - o Independent Metropolis
- Random walk (local Metropolis)

$$q^* = q^{k-1} + Rz$$

o Two (of several) choices: $z \sim N(0,1)$

$$\text{(i) } R = cI \Rightarrow q^* \sim N(q^{k-1}, cI)$$

$$\text{(ii) } R = \text{chol}(V) \Rightarrow q^* \sim N(q^{k-1}, V)$$

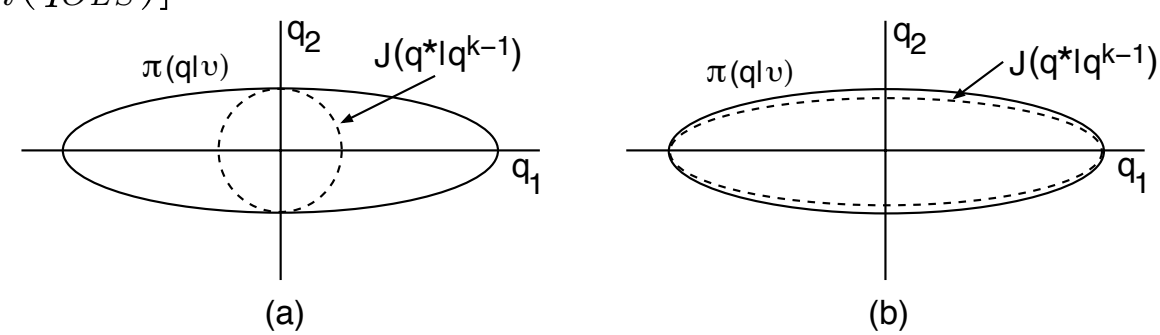
where

$$V = \sigma^2_{OLS}\left[\mathcal{X}^T(q_{OLS})\mathcal{X}(q_{OLS})\right]^{-1}$$

Sensitivity Matrix – Constructed on Monday

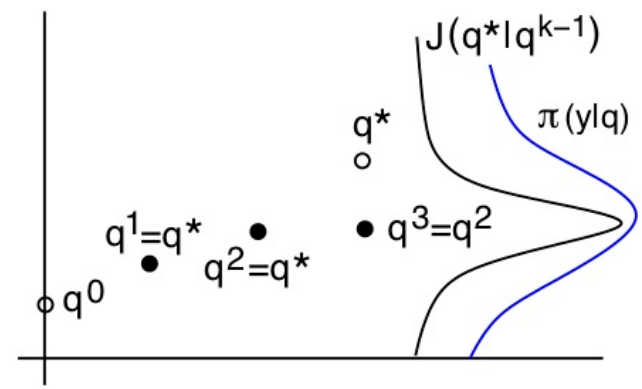
$$\mathcal{X}_{ik}(q_{OLS}) = \frac{\partial f_i(q_{OLS})}{\partial q_k}$$

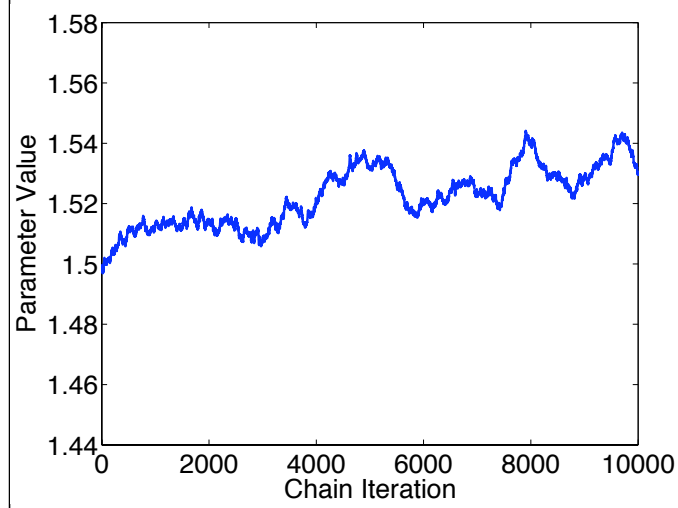
$$\sigma^2_{OLS} = \frac{1}{n-p}\sum_{i=1}^{n}\left[\upsilon_i - f_i(q_{OLS})\right]^2$$



Random Walk Metropolis Algorithm for Parameter Estimation

1. Set number of chain elements M and design parameters n_s, σ_s
2. Determine $q^0 = \arg\min_q \sum_{i=1}^{N} [\upsilon_i - f_i(q)]^2$
3. Set $SS_{q^0} = \sum_{i=1}^{N} [\upsilon_i - f_i(q^0)]^2$
4. Compute initial variance estimate: $s_0^2 = \frac{SS_{q^0}}{n-p}$
5. Construct covariance estimate $V = s_0^2 [\mathcal{X}^T(q^0)\mathcal{X}(q^0)]^{-1}$ and $R = \mathrm{chol}(V)$
6. For $k = 1, \cdots, M$
   (a) Sample $z_k \sim N(0,1)$
   (b) Construct candidate $q^* = q^{k-1} + Rz_k$
   (c) Sample $u_\alpha \sim \mathcal{U}(0,1)$
   (d) Compute $SS_{q^*} = \sum_{i=1}^{N} [\upsilon_i - f_i(q^*)]^2$
   (e) Compute
   $$\alpha(q^*|q^{k-1}) = \min\left(1, e^{-[SS_{q^*} - SS_{q^{k-1}}]/2s_{k-1}^2}\right)$$
   (f) If $u_\alpha < \alpha$,
   Set $q^k = q^*$, $SS_{q^k} = SS_{q^*}$
   else
   Set $q^k = q^{k-1}$, $SS_{q^k} = SS_{q^{k-1}}$
   endif
   (g) Update $s_k \sim \text{Inv-gamma}(a_{val}, b_{val})$ where
   $$a_{val} = 0.5(n_s + n) \ , \ b_{val} = 0.5(n_s\sigma_s^2 + SS_{q^k})$$





Delayed Rejection Adaptive Metropolis (DRAM)

Adaptive Metropolis:

- Update chain covariance matrix as chain values are accepted.

$$V_k = s_p \mathrm{cov}(q^0, q^1, \cdots, q^{k-1}) + \varepsilon I_p$$

- *Diminishing adaptation* and *bounded convergenc*e required since no longer Markov chain.
- Employ recursive relations

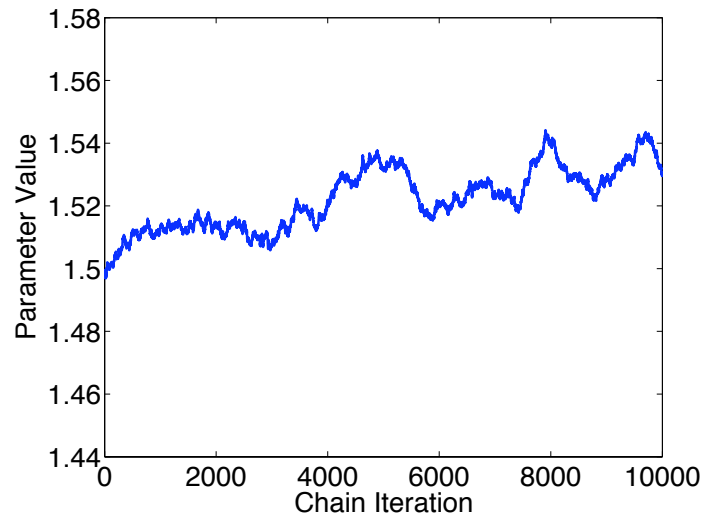
$$\begin{aligned}
\bar{q}^k &= \frac{1}{k+1} \sum_{i=0}^{k} q^i \\
&= \frac{k}{k+1} \cdot \frac{1}{k} \sum_{i=0}^{k-1} q^i + \frac{1}{k+1} q^k \\
&= \frac{k}{k+1} \bar{q}^{k-1} + \frac{1}{k+1} q^k
\end{aligned}$$

$$V_{k+1} = \frac{k-1}{k} V_k + \frac{s_p}{k} \left[k \bar{q}^{k-1} (\bar{q}^{k-1})^T - (k+1) \bar{q}^k (\bar{q}^k)^T + q^k (q^k)^T + \varepsilon I_p \right]$$

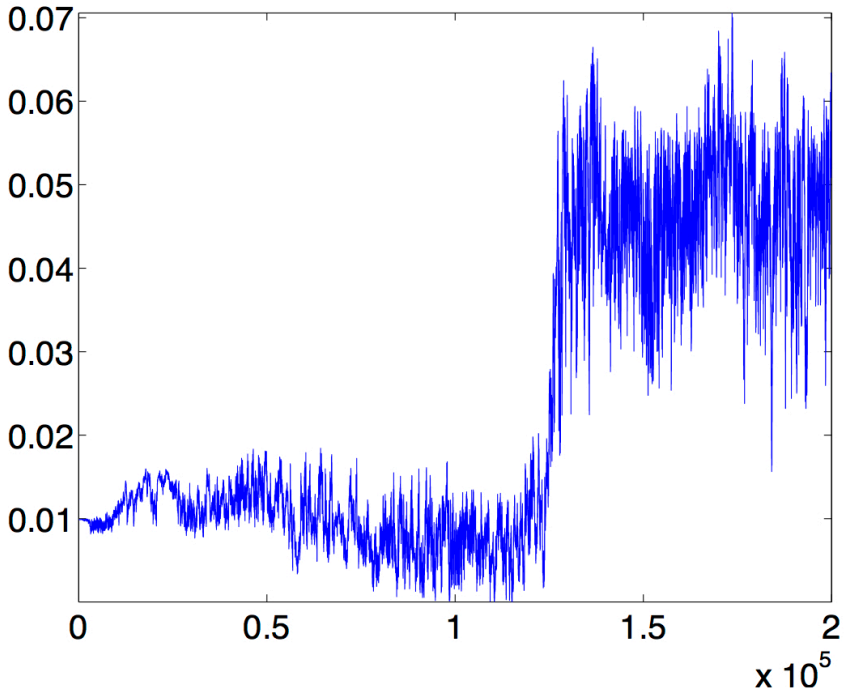
Chain Convergence (Burn-In)

Techniques:

- Visually check chains
- Statistical tests
- Often abused in the literature



Chain not converged



Chain for nonidentifiable parameter

Delayed Rejection Adaptive Metropolis (DRAM)

Websites

- https://rsmith.math.ncsu.edu/UQ_TIA/CHAPTER8/index_chapter8.html
- http://helios.fmi.fi/~lainema/mcmc/

Examples

- Examples on using the toolbox for some statistical problems.

Delayed Rejection Adaptive Metropolis (DRAM)

We fit the Monod model

$$y = \theta_1 \frac{1}{\theta_2 + 1} + \epsilon \quad , \quad \epsilon \sim N(0, I\sigma^2)$$

to observations

x (mg / L COD): 28 55 83 110 138 225 375

y (1 / h): 0.053 0.060 0.112 0.105 0.099 0.122 0.125

First clear some variables from possible previous runs.

```
clear data model options
```

Next, create a data structure for the observations and control variables. Typically one could make a structure data that contains fields xdata and ydata.

```
data.xdata = [28    55    83    110   138   225   375]';   % x (mg / L COD)
data.ydata = [0.053 0.060 0.112 0.105 0.099 0.122 0.125]'; % y (1 / h)
```

Construct model

```
modelfun = @(x,theta) theta(1)*x./(theta(2)+x);
ssfun = @(theta,data) sum((data.ydata-modelfun(data.xdata,theta)).^2);
model.ssfun = ssfun;
model.sigma2 = 0.01^2;
```

Delayed Rejection Adaptive Metropolis (DRAM)

Input parameters

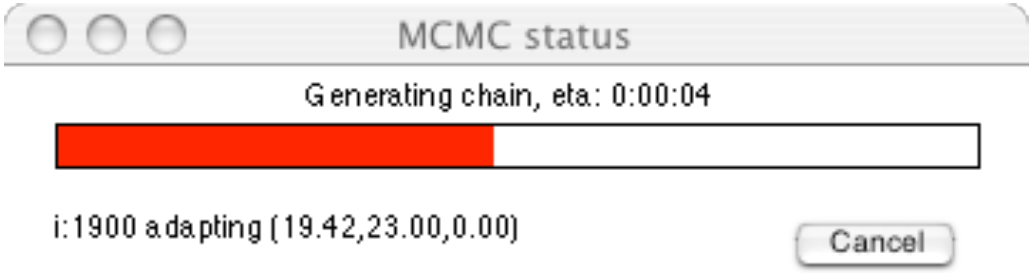
```
 params = {
     {'theta1', tmin(1), 0}
     {'theta2', tmin(2), 0} };
```

and set options

```
options.nsimu = 4000;
options.updatesigma = 1;
options.qcov = tcov;
```

Run code

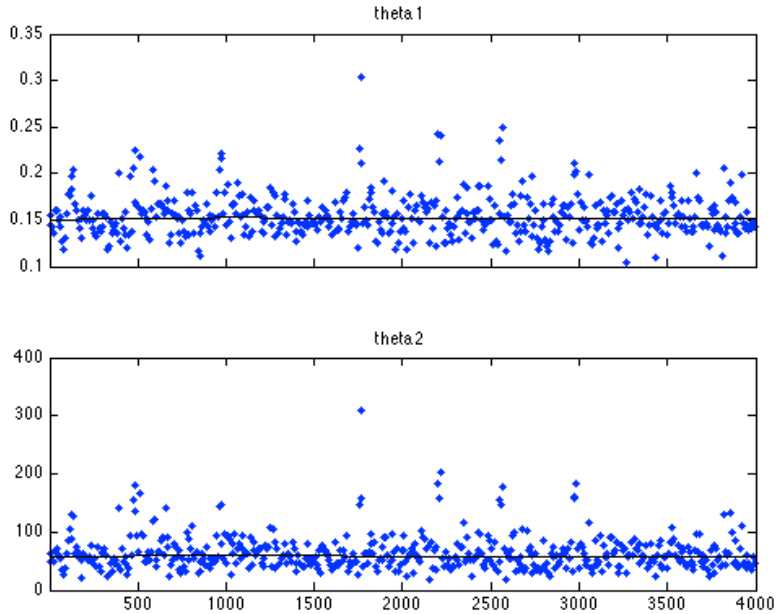
```
[res,chain,s2chain] = mcmcrun(model,data,params,options);
```

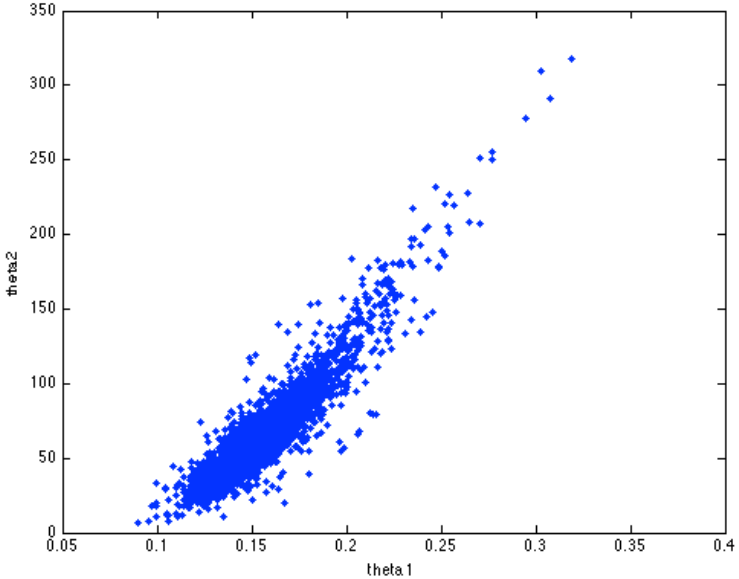


Delayed Rejection Adaptive Metropolis (DRAM)

Plot results

```
figure(2); clf
mcmcplot(chain,[],res,'chainpanel');
figure(3); clf
mcmcplot(chain,[],res,'pairs');
```





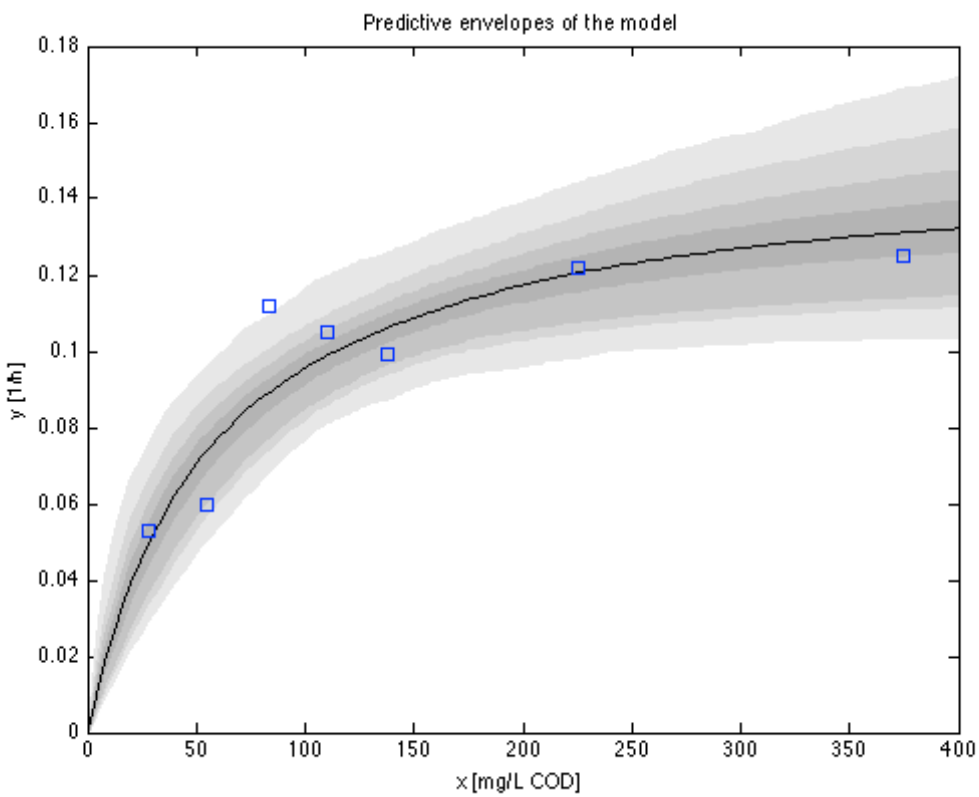
Examples:

- Several available in MCMC_EXAMPLES
- ODE solver illustrated in algae example

Delayed Rejection Adaptive Metropolis (DRAM)

Construct credible and prediction intervals

```
figure(5); clf
out = mcmcpred(res,chain,[],x,modelfun);
mcmcpredplot(out);
hold on
plot(data.xdata,data.ydata,'s'); % add data points to the plot
xlabel('x [mg/L COD]');
ylabel('y [1/h]');
hold off
title('Predictive envelopes of the model')
```



DRAM for SIR Example

SIR Model:

$$\frac{dS}{dt} = \delta N - \delta S - \gamma k I S \quad , \; S(0) = S_0$$ Susceptible

$$\frac{dI}{dt} = \gamma k I S - (r + \delta) I \quad , \; I(0) = I_0$$ Infectious

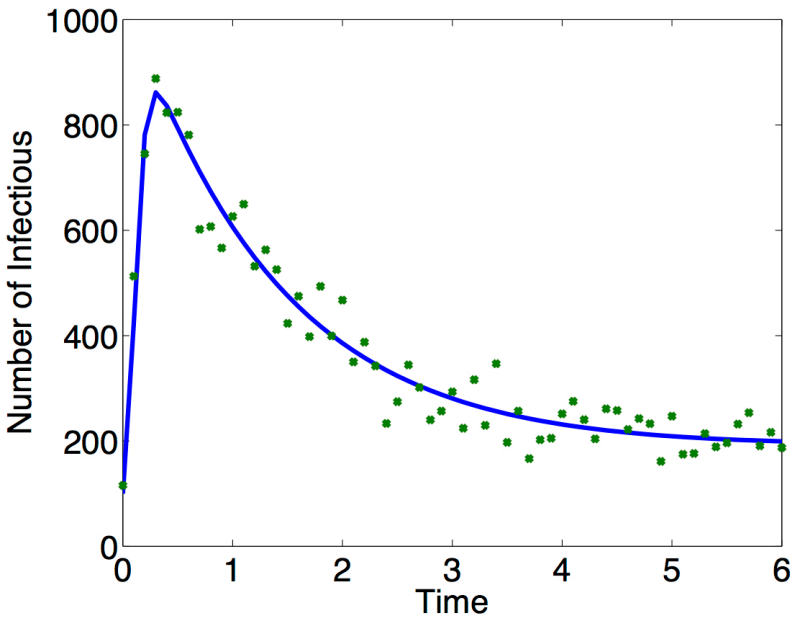
$$\frac{dR}{dt} = r I - \delta R \quad , \; R(0) = R_0$$ Recovered

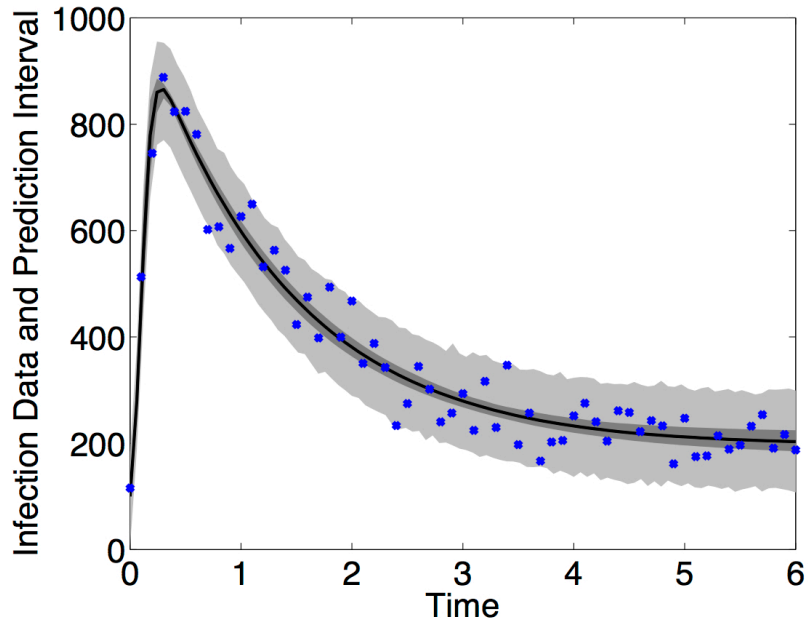
Note: Parameter set $q = [\gamma, k, r, \delta]$ is not identifiable

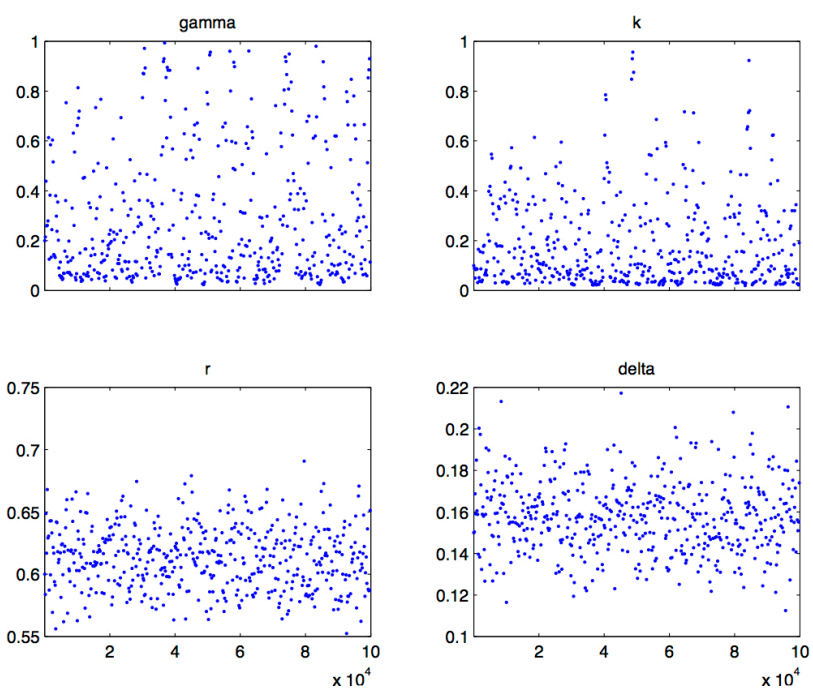
Website

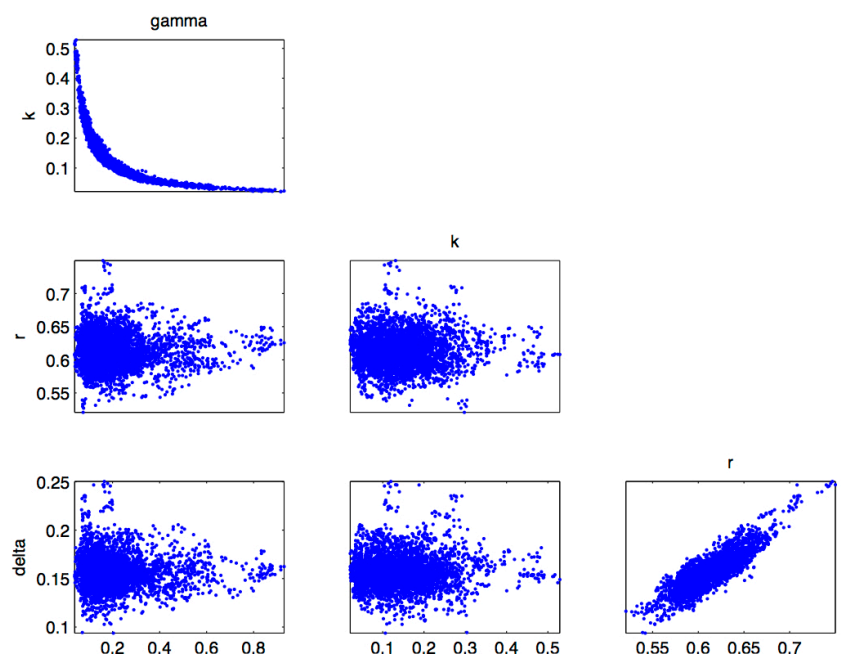
- http://helios.fmi.fi/~lainema/mcmc/
- https://rsmith.math.ncsu.edu

DRAM for SIR Example: Results









SIR Assignment

3 Parameter SIR Model:

$$\frac{dS}{dt} = \delta N - \delta S - \gamma I S \quad , \; S(0) = S_0 \qquad \text{Susceptible}$$

$$\frac{dI}{dt} = \gamma I S - (r + \delta) I \quad , \; I(0) = I_0 \qquad \text{Infectious}$$

$$\frac{dR}{dt} = r I - \delta R \quad , \; R(0) = R_0 \qquad \text{Recovered}$$

Note: Parameter set $q = [\gamma, r, \delta]$ is now identifiable

Assignment:

• Modify the posted 4 parameter code for the 3 parameter model. How do your chains and results compare?

• Consider various chain lengths to establish burn-in.

• Construct and incorporate the covariance matrix based on the sensitivity matrices. See Alun Lloyd's Tutorial.