Gibbs Sampling

Prof. Nicholas Zabaras
University of Notre Dame
Notre Dame, IN, USA

Email: nzabaras@gmail.com
URL: https:/www.zabaras.com/

October 2, 2017

Contents

- Brief Review of Importance Sampling, Sampling Importance Resampling, Solving Ax=b with Importance Sampling
- Incremental Strategies for Sampling, Iterative sampling
- Introduction to MCMC, autoregressive model
- The Gibbs sampler, systematic scan, random scan, Gibbs sampler examples, Block and Metropolized Gibbs, Application in variable/model selection in linear regression

- Monte Carlo Statistical Methods, C.P. Roberts and G. Casella, Chapter 3 (google books, slides, video)
- D Mackay, Introduction to MC methods, reprint.
- R Neal, Probabilistic Inference Using MCMC Methods, 1993.
- C. Andriew et al., An introduction to MCMC for Machine Learning, Machine Learning, 50, 5–43, 2003
- S. Brooks, MCMC methods and its applications, Journal of the Royal Statistical Society. Series D (The Statistician), Vol. 47, No. 1 (1998), pp. 69-100
- G. Casella and EI George, Explaining the Gibbs Sampler, The American Statistician, Vol. 46, 1992, 167-174
- S. Chib and E. Greenberg, Understanding the MH algorithm, *The American Statistician*, Vol. 49, No. 4 (Nov., 1995), pp. 327-335

Summary: Importance Sampling

- Simulation from f (the true density) is not necessarily optimal
- Alternative to direct sampling from f is importance sampling, based on the alternative representation

$$\mathbb{E}_f\left[h(x)\right] = \int_{\chi}\left[h(x)\frac{f(x)}{q(x)}\right]q(x)dx = \mathbb{E}_q\left[h(x)\frac{f(x)}{q(x)}\right]$$

which allows us to use other distributions than f

Importance Sampling Algorithm

Evaluation of $\mathbb{E}_f\left[h(x)\right] = \int_{\chi} h(x)\,f(x)dx$

by

- Generating a sample $x_1, \ldots, x_m$ from a distribution q
- Using the approximation $\frac{1}{m}\sum_{j=1}^{m}\frac{f(x_j)}{q(x_j)}h(x_j)$

Summary: Justification

- Convergence of the estimator

$$\frac{1}{m}\sum_{j=1}^{m}\frac{f(x_j)}{q(x_j)}h(x_j) \to \mathbb{E}_f\left[h(x)\right]$$

- Converges for any choice of the distribution q as long as $\text{supp}(q) \supset \text{supp}(f)$
- Instrumental distribution q chosen from distributions easy to simulate
- q should not be small or zero in regions where the target distribution is significant.
- Same sample (generated from q) can be used repeatedly, not only for different functions h, but also for different densities f

- C.P. Roberts and G. Casella, *Monte Carlo Statistical Methods*, Chapter 3 (google books, slides, video)
- J S Liu, Monte Carlo Strategies in Scientific Computing, Chapter 2
- A. Doucet, Statistical Computing and Monte Carlo Methods (2007)
- J-M Marin and C. P. Robert, Bayesian Core (Chapter 2)

Summary: Choice of Importance Function

- q can be any density but some choices better than others
- Finite variance only when

$$\mathbb{E}_f\left[h^2(x)\frac{f(x)}{q(x)}\right]=\int_{\mathcal{X}} h^2(x)\frac{f^2(x)}{q(x)}dx<\infty$$

- Distributions with tails lighter than those of f (that is, with sup $f/q=\infty$) not appropriate, because weights $f(x_j)/q(x_j)$ vary widely, giving too much importance to a few values x_j.
- If $\sup f/q = M < \infty$, the accept-reject algorithm can be used as well to simulate f directly.
- Importance Sampling suffers from the curse of dimensionality

Discussion

- Importance sampling is useful for a few non-standard distributions but does not work for most other problems.
- The key problem is the design of a proper proposal distribution.
- Sequential MC will be discussed later addressing this last problem.

Additional Readings on Importance Sampling:

- John Geweke, Bayesian Inference in Econometric Models using MC Integration, *Econometrica*, Vol. 57, No. 6 (Nov., 1989), pp. 1317-1339.
- Herman K. Van Dijk, J. Peter Hop and Adri S. Louter, An Algorithm for the Computation of Posterior Moments and Densities Using Simple Importance Sampling, *Journal of the Royal Statistical Society. Series D (The Statistician)*, Vol. 36, No. 2/3, (1987) pp. 83-90.
- Art Owen and Yi Zhou, Safe and effective importance sampling, *Journal of the American Statistical Association*, Vol. 95, No. 449 (Mar., 2000), pp. 135-143.

Sampling Importance Resampling (SIR)

- Let us draw unweighted samples from $p(x)$ by first using importance sampling (with proposal *q*) to generate a distribution of the form

$$p(\boldsymbol{x}) \approx \sum_s w_s \delta_{\boldsymbol{x}^s}(\boldsymbol{x}),\ \boldsymbol{x}^s \sim q(\boldsymbol{x})$$

- Here w_s are the normalized importance weights. We then sample with replacement from the Eq. above, where the probability that we pick $\boldsymbol{x^s}$ is w_s. This procedure induces a distribution denoted by $\hat{p}$.

- To see that this is valid, note that

$$\hat{p}(x \le x_0) = \sum_s \mathbb{I}(x^s \le x_0)\, w_s = \frac{\sum_s \mathbb{I}(x^s \le x_0) \frac{\tilde{p}(x^s)}{q(x^s)}}{\sum_s \frac{\tilde{p}(x^s)}{q(x^s)}} \to \frac{\int \mathbb{I}(x \le x_0) \frac{\tilde{p}(x)}{q(x)} q(x) dx}{\int \frac{\tilde{p}(x)}{q(x)} q(x) dx}$$

$$= \frac{\int \mathbb{I}(x \le x_0) \tilde{p}(x) dx}{\int \tilde{p}(x) dx} = \int \mathbb{I}(x \le x_0) p(x) dx = p(x \le x_0)$$

- This SIR result is an unweighted approximation $p(\boldsymbol{x}) \approx \frac{1}{S'} \sum_{s=1}^{S'} \delta_{\boldsymbol{x}^s}(\boldsymbol{x}), S' \ll S$.
  - Smith, A. F. M. and A. E. Gelfand (1992). Bayesian statistics without tears: A sampling-resampling perspective. *The American Statistician 46*(2), 84–88.

Solving Ax=b with Importance Sampling

- Consider the system of equations $Ax = b,\ A \in \mathbb{R}^{n \times n}$
- Multiply this linear system with an invertable matrix *G*:

$$GAx = Gb,\ where: GA = I - B\ with \quad \rho(B) < 1$$

h

$spectral\ radius\ of\ B$

- Then the solution of the linear system is:

$$x = \sum_{k=0}^{\infty} B^k h$$

- Or in component form:

$$x_i = \sum_{k=0}^{\infty} \sum_{i_1=1}^{n} \sum_{i_2=1}^{n} \dots \sum_{i_k=1}^{n} B_{ii_1} B_{i_1 i_2} \dots B_{i_{k-1} i_k} h_k$$

- G. E. Forsythe; Richard A. Leibler, Matrix Inversion by a Monte Carlo Method, *Mathematical Tables and Other Aids to Computation*, Vol. 4, No. 31. (Jul., 1950), pp. 127-129.
- J. H. Curtiss, Monte Carlo Methods for the Iteration of Linear Operators, *Journal of Mathematics and Physics*, Vol. 32 (1953) 209-232.
- John H. Halton, Sequential Monte Carlo techniques for the solution of linear systems, Journal of Scientific Computing, Vol. 9, Number 2 / June, (1994).

Solving Ax=b with Importance Sampling

$$x_i = \sum_{k=0}^{\infty} \sum_{i_1=1}^{n} \sum_{i_2=1}^{n} \dots \sum_{i_k=1}^{n} B_{ii_1} B_{i_1 i_2} \dots B_{i_{k-1} i_k} h_k$$

- Introducing the following sequence of indices:

$$\gamma_k = (i_1, i_2, \dots, i_k), \; i_i \in \{1, 2, \dots, n\}$$

we can write the above equation as follows:

$$x_i = \sum_{\gamma_k} a_i(\gamma_k), \; where: a_i(\gamma_k) = \begin{cases} B_{ii_1} B_{i_1 i_2} \dots B_{i_{k-1} i_k} h_k \; if \; k > 0 \\ h_i \; if \; k = 0 \end{cases}$$

- We see that x_i is the average of a_i with respect to the uniform distribution of indices γ_k of any length k.

$$x_i \sim \mathbb{E}_{\pi}\left[a_i(\gamma_k)\right]$$

Not known normalization constant for π

Solving Ax=b with Importance Sampling

$$x_i \sim \mathbb{E}_\pi \left[a_i(\gamma_k) \right]$$

- We use an importance sampling approach:

$$x_i = \sum_{\gamma_k} a_i(\gamma_k) = \sum_{\gamma_k} \frac{a_i(\gamma_k)}{q(\gamma_k)} q(\gamma_k) = \mathbb{E}_q \left[\frac{a_i(\gamma_k)}{q(\gamma_k)} \right]$$

- We define the density q using "a random walk of k steps on indices":

$$q(\quad \gamma_k \quad) = \underbrace{P_{ii_1} P_{i_1 i_2} \ldots P_{i_{k-1} i_k}}_{\text{transition probabilities}} \quad \underbrace{P_{i_k}}_{\substack{\text{stopping probability} \\ \text{at index } i_k}} \quad , \quad P_i = 1 - \underbrace{\sum_{j=1}^{n} P_{ij}}_{<1}$$

$$(i_1, i_2, \ldots, i_k)$$

- To obtain sequences of size k, we introduce a stopping probability at each state i.

Solving Ax=b with Importance Sampling

- Step 1: Draw *N* multi-indices from *q*

$$\gamma_k^{(j)} = \left(i_1^{(j)}, i_2^{(j)}, \ldots, i_k^{(j)} \right)$$

- Step 2: Compute

$$\hat{x}_i = \frac{1}{N} \sum_{j=1}^{N} \frac{a_i\left(\gamma_k^{(j)}\right)}{q\left(\gamma_k^{(j)}\right)}$$

where:

$$\frac{a_i(\gamma_k)}{q(\gamma_k)} = \begin{cases} \dfrac{B_{ii_1} B_{i_1 i_2} \ldots B_{i_{k-1} i_k} h_{i_k}}{P_{ii_1} P_{i_1 i_2} \ldots P_{i_{k-1} i_k} P_{i_k}} & if \;\; k > 0 \\ \dfrac{h_i}{p_i} \; if \;\; k = 0 \end{cases}$$

Solving Ax=b with Importance Sampling

- Consider the example:

$$\underbrace{\begin{bmatrix} 1.1 & -0.5 \\ -0.5 & 1.1 \end{bmatrix}}_{A} x = \underbrace{\begin{bmatrix} 1 \\ 1 \end{bmatrix}}_{b}$$

- Let

$$A = I - B \Rightarrow B = \begin{bmatrix} -0.1 & 0.5 \\ 0.5 & -0.1 \end{bmatrix} \Rightarrow x = \sum_k B^k b$$

- The analytical solution is:

$$x = \begin{bmatrix} 1.67 \\ 1.67 \end{bmatrix}$$

Solving Ax=b with Importance Sampling

- Transition kernel

$$\begin{array}{c} \\ 1 \\ 2 \end{array} \begin{array}{c} \begin{array}{ccc} 1 & 2 & \text{stop} \end{array} \\ \left[\begin{array}{ccc} 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \end{array}\right] \end{array}$$

- To estimate *x(1)*, we perform the algorithm in this way
  - step 1
    generate "a Markov Chain" from the transition kernel and starting from index 1, e.g. a chain such as

$$1 \xrightarrow[\Pr(1,i_1)]{B(1,i_1)} i_1 \xrightarrow[\Pr(i_1,i_2)]{B(i_1,i_2)} i_2 \xrightarrow[\Pr(i_2,i_3)]{B(i_2,i_3)} \cdots \xrightarrow[\Pr(i_{k-1},i_k)]{B(i_{k-1},i_k)} i_k \xrightarrow[\Pr(i_k,\text{stop})]{b(i_k)} \text{Stop}$$

    Then we get

$$x^{(n)} = \frac{B(1,i_1)B(i_1,i_2)\cdots B(i_{k-1},i_k)b(i_k)}{\Pr(1,i_1)\Pr(i_1,i_2)\cdots\Pr(i_{k-1},i_k)\Pr(i_k,\text{stop})}$$

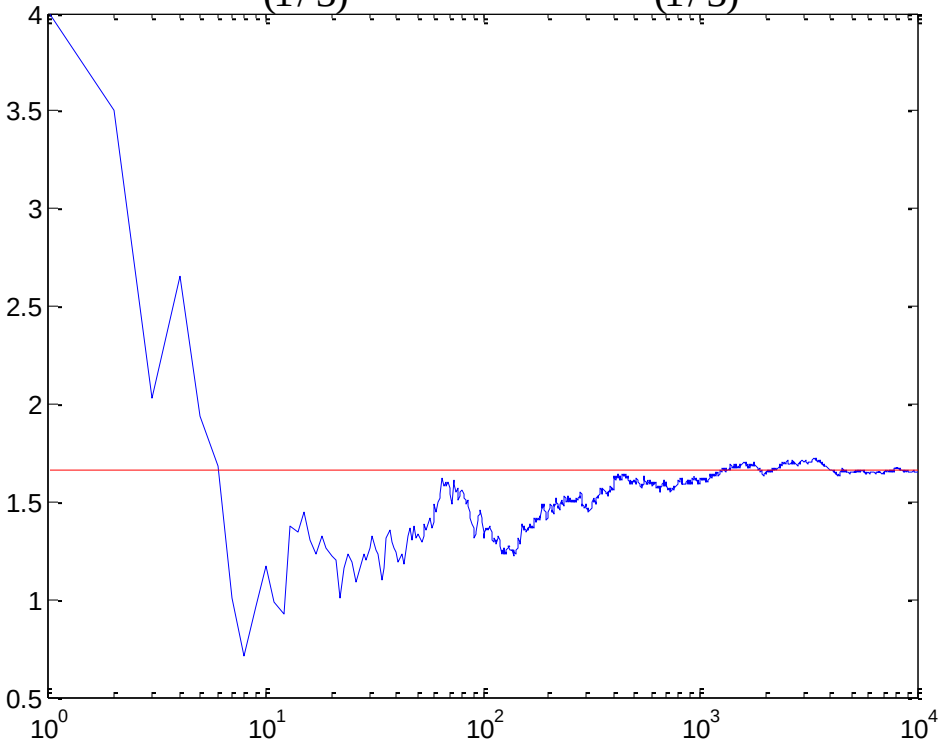
  - step 2: repeat step 1 and average on $x^{(n)}$

- Note that in the implementation, we don't need to define explicitly the length *k* of the chains. Since we have specified a stopping probability for each state (i=1,…n), the chains generated will automatically be with different *k*.

Solving Ax=b with Importance Sampling

- For example, for a Markov chain such that $1 \to 2 \to 2 \to 1 \to stop$

The estimated $x = \frac{B(1,2)B(2,2)B(2,1)b(1)}{(1/3)^4} = \frac{0.5 \times (-0.1) \times 0.5 \times 1}{(1/3)^4} = -2.025$



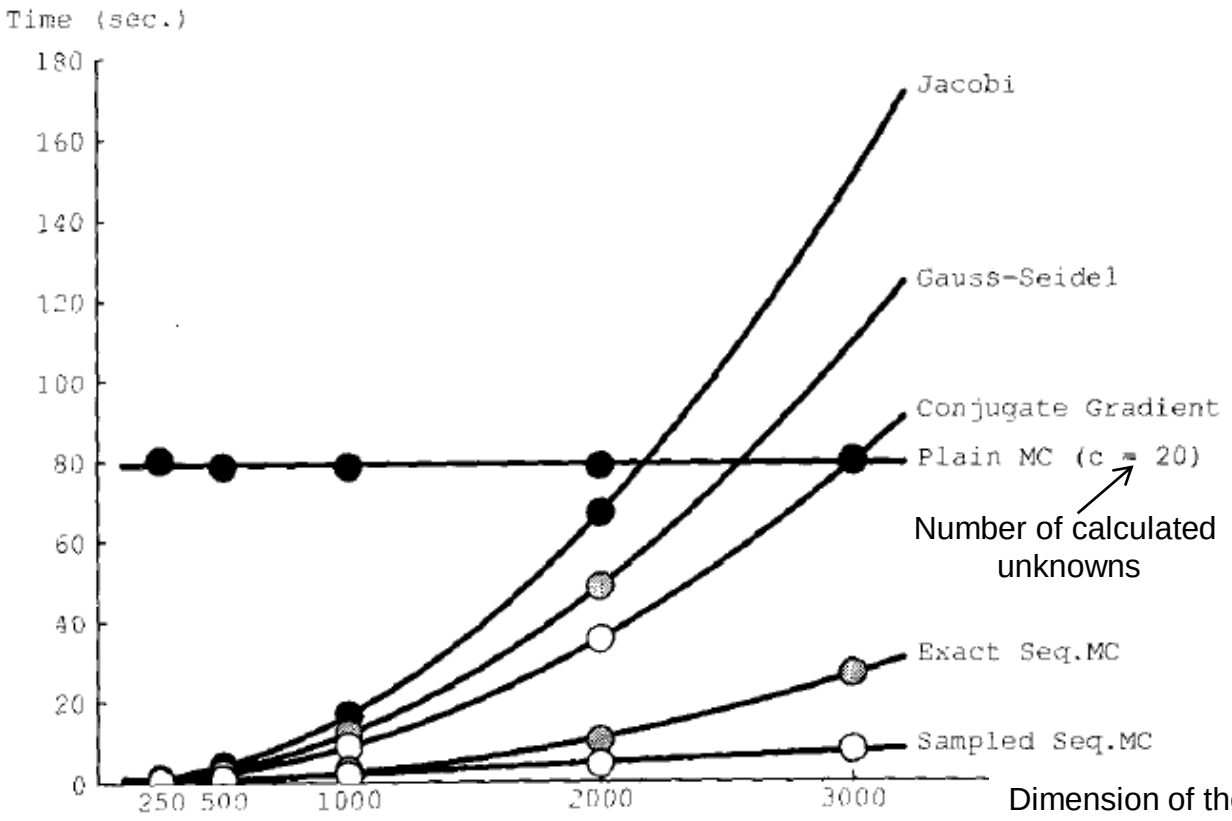
A MatLab implementation is provided here.

Another MatLab implementation for solving for both or a single variable is also available.

- The cost of this IS solver of linear equations is $\mathcal{O}$(nsN), N=# of samples, s=average length of MCMC walks

Performance of MC in Solving Linear Systems

- A comparison is given below of the MC solver versus classical methods.



- **Direct Methods** (Gauss elimination, LU, Cholesky): $\mathcal{O}(n^3)$
- **Iterative Methods** (Jacobi, Gauss-Seidel) $\mathcal{O}(n^2 s)$, s=number of iterations
- **Monte Carlo Importance Sampling with n unknowns:** $\mathcal{O}$ (n s N) for uniform transition kernel or $\mathcal{O}$ (n^2 s N) otherwise. Here, s is the average length of walks and N the number of samples.
- **Monte Carlo Importance Sampling with m<<n unknowns:** If we only want to compute m out of the n components of the vector ***x***, then the number of operations needed drops down to $\mathcal{O}$ (m s N) for uniform transition kernel or $\mathcal{O}$ (m n s N) otherwise. MC then becomes highly favorable.

- John H. Halton, Sequential Monte Carlo techniques for the solution of linear systems, Journal of Scientific Computing, Vol. 9, Number 2 / June, (1994).

Using Incremental Strategies for Sampling

- We have seen that both rejection sampling (RS) and importance sampling (IS) are limited to problems of moderate dimensions.
- The problem with these algorithms is that we try to sample all the components of a high-dimensional parameter simultaneously.
- We can learn next incremental strategies:

  - Iterative Methods: Markov chain Monte Carlo.

  - Sequential Methods: Sequential Monte Carlo.

A. Doucet, Statistical Computing: Monte Carlo Methods, Online course resource

Motivating Example

- Multiple failures in a nuclear plant:

Pump	1	2	3	4	5	6	7	8	9	10
Failures	5	1	5	14	3	19	1	1	4	22
Times	94.32	15.72	62.88	125.76	5.24	31.44	1.05	1.05	2.10	10.48

- Model: Failures of the i^{th} pump follow a Poisson process with parameter $\lambda_i, 1 \le \lambda_i \le 10$. For an observation time t_i, the number of failures p_i is a Poisson $\mathcal{P}(\lambda_t t_i)$ random variable.

- The unknowns consist of $\theta := (\lambda_1, \lambda_2, ..., \lambda_{10}, \beta)$ where β is a parameter in the hierarchical model introduced next.

Poisson	$\theta \sim \text{Poisson}(\lambda)$ $p(\theta) = \text{Poisson}(\theta \mid \lambda)$	'rate' $\lambda > 0$

$p(\theta) = \frac{1}{\theta!}\lambda^{\theta}\exp(-\lambda)$ $\quad$ $\mathrm{E}(\theta) = \lambda,\ \mathrm{var}(\theta) = \lambda$
$\theta = 0, 1, 2, \ldots$ $\quad$ $\mathrm{mode}(\theta) = \lfloor \lambda \rfloor$

Statistical Computing and MC Methods, A. Doucet, Lecture 10.

Motivating Example: Nuclear Pump Data

- Hierarchical Model:

$$\lambda_i \overset{i.i.d.}{\sim} \mathcal{Ga}(\alpha, \beta), \text{ and } \beta \sim \mathcal{Ga}(\gamma, \delta),$$

with $\alpha=1.8$, $\gamma=0.01$, $\delta=1$.

- The posterior distribution (see here $\mathcal{Ga}$ distribution)

$$\pi(\lambda_i, \beta \mid t_i, p_i) \propto \prod_{i=1}^{10} \left\{ \underbrace{(\lambda_i t_i)^{p_i} e^{-\lambda_i t_i}}_{\mathcal{P}(\lambda_i t_i)} \underbrace{\beta^{\alpha} \lambda_i^{\alpha-1} e^{-\beta\lambda_i}}_{\lambda_i \sim \mathcal{Ga}(\alpha,\beta)} \right\} \underbrace{\beta^{\gamma-1} e^{-\delta\beta}}_{\beta \sim \mathcal{Ga}(\gamma,\delta)} \propto$$

$$\prod_{i=1}^{10} \left\{ \lambda_i^{p_i+\alpha-1} e^{-\lambda_i (t_i+\beta)} \right\} \beta^{10\alpha+\gamma-1} e^{-\delta\beta}$$

- It is not obvious how the inverse CDF method or the accept/reject method or how importance sampling could be used for this multidimensional distribution!

Gamma	$\theta \sim \text{Gamma}(\alpha, \beta)$ $p(\theta) = \text{Gamma}(\theta \mid \alpha, \beta)$	shape $\alpha > 0$ inverse scale $\beta > 0$

$$p(\theta) = \frac{\beta^{\alpha}}{\Gamma(\alpha)} \theta^{\alpha-1} e^{-\beta\theta}, \quad \theta > 0$$

$$\text{E}(\theta) = \frac{\alpha}{\beta}$$
$$\text{var}(\theta) = \frac{\alpha}{\beta^2}$$
$$\text{mode}(\theta) = \frac{\alpha-1}{\beta}, \text{ for } \alpha \geq 1$$

Conditional Distributions

$$\pi(\lambda_i, \beta \mid t_i, p_i) \propto \prod_{i=1}^{10} \left\{ \lambda_i^{p_i+\alpha-1} e^{-\lambda_i(t_i+\beta)} \right\} \beta^{10\alpha+\gamma-1} e^{-\delta\beta}$$

- The conditionals can be obtained with direct observation from the above posterior:

$$\lambda_i \mid (\beta, t_i, p_i) \sim \mathcal{Ga}(p_i + \alpha, t_i + \beta) \ \ for\ 1 \le i \le 10$$

$$\beta \mid (\lambda_1, ..., \lambda_{10}) \sim \mathcal{Ga}(\gamma + 10\alpha, \delta + \sum_{i}^{10} \lambda_i)$$

Gamma	$\theta \sim \text{Gamma}(\alpha, \beta)$ $p(\theta) = \text{Gamma}(\theta \mid \alpha, \beta)$	shape $\alpha > 0$ inverse scale $\beta > 0$

$$p(\theta) = \frac{\beta^\alpha}{\Gamma(\alpha)} \theta^{\alpha-1} e^{-\beta\theta}, \ \ \theta > 0$$

$$\text{E}(\theta) = \frac{\alpha}{\beta}$$
$$\text{var}(\theta) = \frac{\alpha}{\beta^2}$$
$$\text{mode}(\theta) = \frac{\alpha-1}{\beta}, \text{ for } \alpha \ge 1$$

- Instead of directly sampling the vector $\theta = (\lambda_1, ..., \lambda_{10}, \beta)$ *at once,* one could suggest sampling it iteratively.
  - We can start with the λ_i's for a given guess of β, followed by an update of β given the new samples $(\lambda_1, ..., \lambda_{10})$.

Iterative Sampling

- Given a sample, at iteration t, $\theta^t = (\lambda_1^t, ..., \lambda_{10}^t, \beta^t)$, one could proceed as follows at iteration $t + 1$,

$$Step\ 1: \lambda_i^{t+1} \mid (\beta^t, t_i, p_i) \sim \mathcal{Ga}(p_i + \alpha, t_i + \beta^t) \ \ for\ 1 \le i \le 10$$

$$Step\ 2: \beta^{t+1} \mid (\lambda_1^{t+1}, ..., \lambda_{10}^{t+1}) \sim \mathcal{Ga}(\gamma + 10\alpha, \delta + \sum_{i=1}^{10} \lambda_i^{t+1})$$

- Note that instead of directly sampling in a space of dimension 11, one samples 11 times in spaces of dimension 1!

Iterative Sampling

- With this iterative procedure:
  - Are we sampling from the desired joint distribution of the 11 variables?
  - If yes, how many times should the iteration above be repeated?
- The validity of the approach described here is derived from the fact that the sequence $\left\{\theta^t\right\} := \left\{\lambda_1^t, \lambda_2^t, ..., \lambda_{10}^t, \beta^t\right\}$ is a Markov chain.

Introduction to Markov Chain Monte Carlo

- **Markov chain:** A sequence of random variables $\{X_n, n \in \mathbb{N}\}$ defined on $(\mathcal{X}, \mathcal{B}(\mathcal{X}))$ such that for any $A \in \mathcal{B}(\mathcal{X})$ the following probability condition is satisfied:

$$\mathbb{P}(X_n \in A \mid X_0, \ldots, X_{n-1}) = \mathbb{P}(X_n \in A \mid X_{n-1})$$

and we write:

$$Transition\ Kernel: P(x, A) = \mathbb{P}(X_n \in A \mid X_{n-1})$$

- **Markov Chain Monte Carlo (MCMC):** Given a target distribution π, we need to design a transition kernel P such that asymptotically

$$\frac{1}{N}\sum_{n=1}^{N} f(X_n) \overset{N\to\infty}{\to} \int f(x)\pi(x)dx \ and\ / or\ X_n \sim \pi$$

- It is easy to simulate the Markov Chain even if π is complex.

Autoregressive Model

- Consider the autoregression model for $|\alpha| < 1$

$$X_n = \alpha X_{n-1} + V_n, \text{ where } V_n \sim \mathcal{N}(0, \sigma^2)$$

- The limiting distribution is:

$$\pi(x) = \mathcal{N}\left(x; 0, \frac{\sigma^2}{1-\alpha^2}\right)$$

- To sample from π, we just sample the Markov chain and we know that asymptotically $X_n \sim \pi$

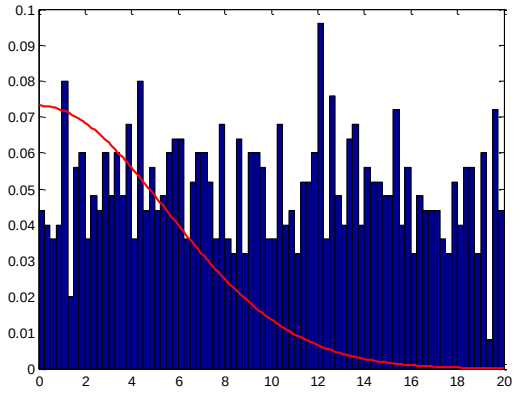
- Of course this problem is only to demonstrate the main idea of MCMC since we can here sample directly from π!

Autoregressive Model

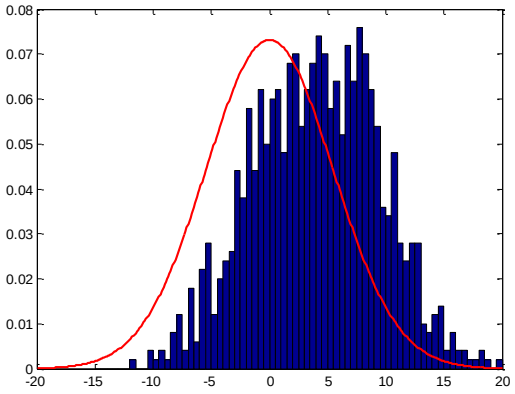
- Consider 100 independent Markov chains run in parallel.
- We assume that the initial distribution of these Markov chains is $\mathcal{U}[0,20]$.
- So initially, the Markov chains samples are not distributed according to π.
- In the following example, we choose $\alpha = 0.4, \quad \sigma = 5$ (see here for a MatLab implementation)

Example

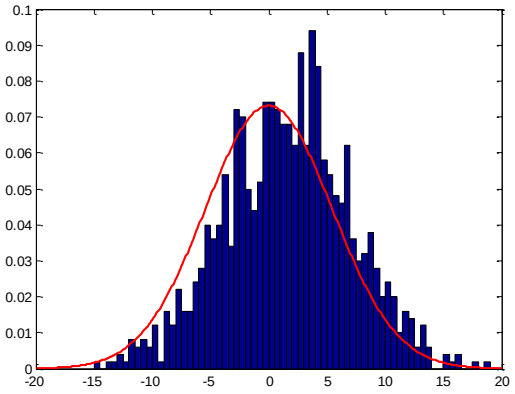
- A Markov chain with a normal distribution as target distribution.



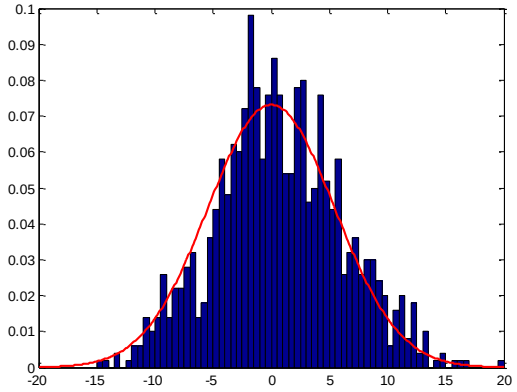
Initial distribution



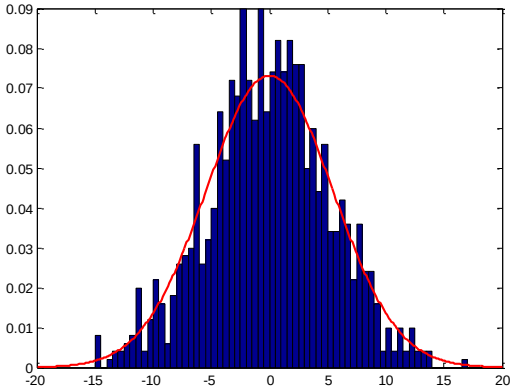
step=1



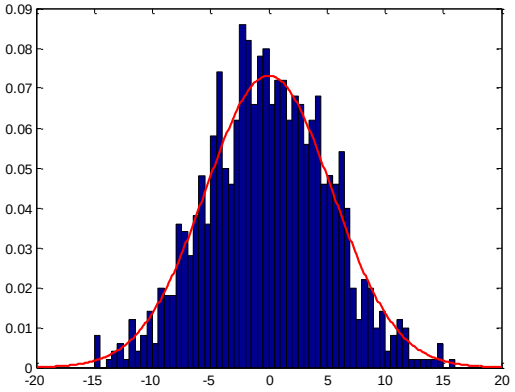
step=2



step=3



step=4

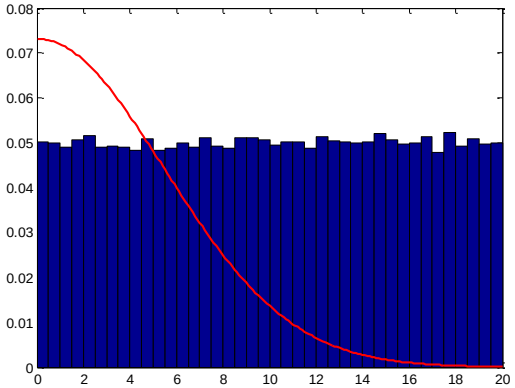


step=100

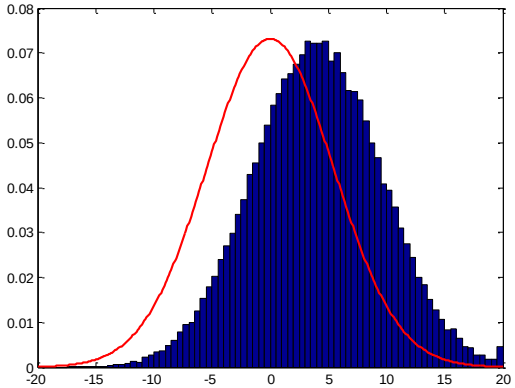
MatLab implementation

Example

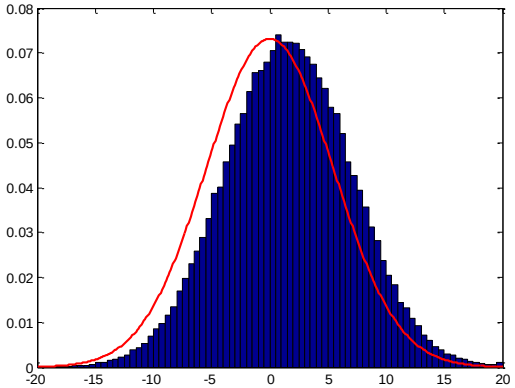
- Histograms of 100 independent Markov chains with a normal distribution as target distribution.



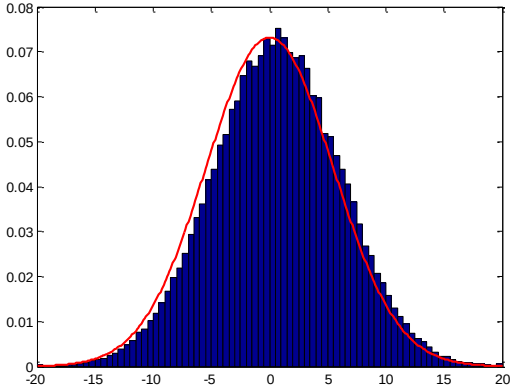
Initial distribution



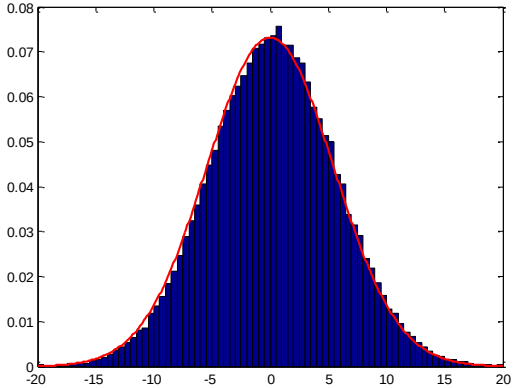
step=1



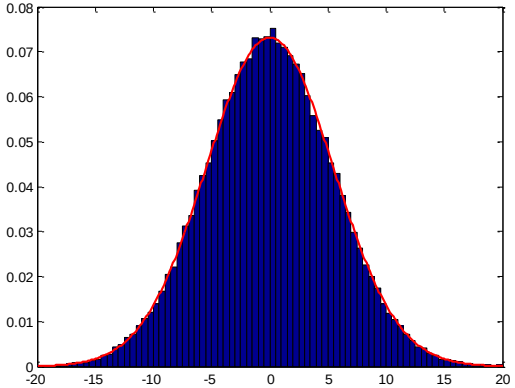
step=2



step=3



step=4



step=100

MatLab implementation

Example

- The target normal distribution seems to “attract” the distribution of the samples and even to be a fixed point of the algorithm.
- We have produced 100 independent samples from the normal distribution.
- We will see that it is not necessary to run *N* Markov chains in parallel in order to obtain 100 samples, but that one can consider a unique Markov chain, and build the histogram from this single Markov chain by forming histograms from one trajectory.

Markov Chain Monte Carlo

- The estimate of the target distribution, through the series of histograms, improves with the # of iterations.
- Assume that we have stored $\{X_n, n=1,...,N\}$ for *N* large and wish to estimate

$$\int_x f(x)\pi(x)dx.$$

- We suggest the estimator $\frac{1}{N}\sum_{n=1}^{N} f(X_n)$ which is the estimator we used before when $\{X_n, n=1,...,N\}$ were independent.
- Under relatively mild conditions, such an estimator is consistent despite the fact that the samples are not independent. Under additional conditions,the CLT also holds with a rate of convergence $1/\sqrt{N}$.

Markov Chain Monte Carlo

- We are interested in Markov chains with transition kernel *P* which has the following three important properties observed in the autoregressive example:

  A. The desired distribution π is an invariant distribution of the Markov chain, i.e.

$$\int_x \pi(x)P(x,y)dx = \pi(y)$$

  B. The successive distributions of the Markov chains converge towards π regardless of the starting point.

  C. The estimator $\frac{1}{N}\sum_{n=1}^{N} f(X_n)$ converges towards $\mathbb{E}_\pi(f(X))$ and asymptotically $X_n \sim \pi$ (stronger requirement)

Markov Chain Monte Carlo

- Since there is an infinite number of kernels *P(x, y)* which admit $\pi(x)$ as their invariant distribution, the main task in MCMC is coming up with good ones.

- Convergence is ensured under very weak assumptions -- irreducibility and aperiodicity.

- It is usually easy to establish that an MCMC sampler converges towards $\pi(x)$ but difficult to obtain rates of convergence.

The Gibbs Sampler

- The Gibbs sampler is a generic method to sample from a high dimensional distribution.

- It generates a Markov chain which converges to the target distribution under weak assumptions: irreducibility and aperiodicity.

The Two Component Gibbs Sampler

Consider the target distribution $\pi(\theta)$ such that $\theta=\{\theta^1, \theta^2\}$. The two component Gibbs sampler proceeds as follows:

- Initialization:

  Select deterministically or randomly $\theta_0 = \left(\theta_0^1, \theta_0^2\right)$

- Iteration i, i≥1.
  - Sample $\theta_i^1 \sim \pi\left(\theta^1 \mid \theta_{i-1}^2\right)$
  - Sample $\theta_i^2 \sim \pi\left(\theta^2 \mid \theta_i^1\right)$

- Sampling from conditionals is often feasible even when sampling from the joint is impossible (e.g. in the nuclear pump data).

Invariant Distribution

- Clearly $\left\{\left(\theta_i^1, \theta_i^2\right)\right\}$ is a Markov Chain. Its transition kernel is:

$$P\left(\left(\theta^1, \theta^2\right), \left(\tilde{\theta}^1, \tilde{\theta}^2\right)\right) = \pi\left(\tilde{\theta}^1 \middle| \theta^2\right)\pi\left(\tilde{\theta}^2 \middle| \tilde{\theta}^1\right)$$

- The detailed balance equation $\int_x \pi(x)P(x,y)dx = \pi(y)$ is satisfied:

$$\iint \pi\left(\theta^1, \theta^2\right)P\left(\left(\theta^1, \theta^2\right), \left(\tilde{\theta}^1, \tilde{\theta}^2\right)\right)d\theta^1\, d\theta^2 =$$

$$\iint \pi\left(\theta^1, \theta^2\right)\pi\left(\tilde{\theta}^1 \middle| \theta^2\right)\pi\left(\tilde{\theta}^2 \middle| \tilde{\theta}^1\right)d\theta^1\, d\theta^2 =$$

$$\int \pi\left(\theta^2\right)\pi\left(\tilde{\theta}^1 \middle| \theta^2\right)\pi\left(\tilde{\theta}^2 \middle| \tilde{\theta}^1\right)d\theta^2 =$$

$$\int \pi\left(\tilde{\theta}^1, \theta^2\right)\pi\left(\tilde{\theta}^2 \middle| \tilde{\theta}^1\right)d\theta^2 = \pi\left(\tilde{\theta}^1\right)\pi\left(\tilde{\theta}^2 \middle| \tilde{\theta}^1\right) = \pi\left(\tilde{\theta}^1, \tilde{\theta}^2\right)$$

Irreducibility

- The detailed balance does not ensure that the Gibbs sampler converges towards the invariant distribution.

- Additionally, it is required to ensure irreducibility: the Markov chain can move to any set *A* such that $\pi(A)>0$ from (almost) any starting point.

- This ensures that

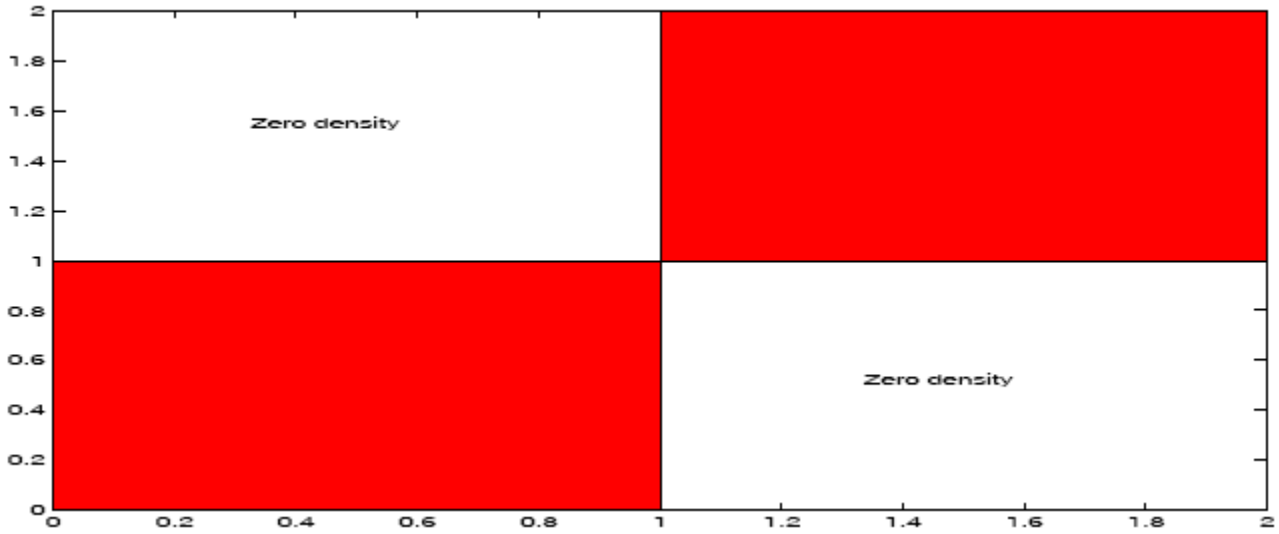
$$\frac{1}{N}\sum_{n=1}^{N} f\left(\theta_n^1,\theta_n^2\right) \rightarrow \int f\left(\theta^1,\theta^2\right)\pi\left(\theta^1,\theta^2\right)d\theta^1 d\theta^2$$

but not that asymptotically $\left(\theta_n^1,\theta_n^2\right) \sim \pi$

Irreducibility

- A distribution is shown here that leads to a reducible Gibbs sampler.

- Conditioning on $x_1 < 1$, the distribution of x_2 cannot produce a value in [1,2].



Aperiodicity

- Consider an example with $\mathcal{X} = \{1, 2\}$ and transition probabilities $P(1, 2) = P(2, 1) = 1$. The invariant distribution is clearly given by $\pi(1) = \pi(2) = 1/2$.

- However, we know that if the chain starts in $X_0 = 1$, then $X_{2n} = 1$ and $X_{2n+1} = 2$ for any n.

- We have

$$\frac{1}{N}\sum_{n=1}^{N} f\left(X_n\right) \rightarrow \int f(x)\pi(x)dx$$

but clearly X_n is not distributed according to π.

- You need to make sure that you do not explore the space in a periodic way to ensure that $X_n \sim \pi$ asymptotically.

Gibbs Sampler

- If $\theta = (\theta_1, \theta_2, ..., \theta_p)$ where *p>2*, the Gibbs sampler still applies.
- Initialization:
  - Select deterministically or randomly $\theta^{(0)} = (\theta_1^{(0)}, \theta_1^{(0)}, ..., \theta_p^{(0)})$
- Iteration i, i ≥ 1
  - For k=1:p
    - Sample $\theta_k^{(i)} \sim \pi(\theta_k \mid \theta_{-k}^{(i)})$

      where $\theta_{-k}^{(i)} = (\theta_1^{(i)}, ..., \theta_{k-1}^{(i)}, \theta_{k+1}^{(i-1)}, ..., \theta_p^{(i-1)})$

Systematic-Scan Gibbs Sampler

- Systematic Scan Gibbs: Let $\theta^{(i)} = \left(\theta_1^{(i)}, \theta_1^{(i)}, ..., \theta_p^{(i)}\right)$
  - Update $\theta_1^{(i)}$ from $\pi(.|\theta_2^{(i-1)}, ..., \theta_p^{(i-1)})$
  - Update $\theta_2^{(i)}$ from $\pi(.|\theta_1^{(i)}, \theta_3^{(i-1)}, ..., \theta_p^{(i-1)})$
  - ……
  - Update $\theta_p^{(i)}$ from $\pi(.|\theta_1^{(i)}, \theta_2^{(i)}, ..., \theta_{p-1}^{(i)})$

Random Scan Gibbs Sampler

- Consider again: $\theta = (\theta_1, \theta_2, ..., \theta_p)$ where $p>2$. We consider the following random scan Gibbs sampler.

- Initialization:
  - Select deterministically or randomly $\theta_0 = (\theta_1^{(0)}, \theta_2^{(0)}, ..., \theta_p^{(0)})$

- Iteration i, i ≥ 1
  - Sample $K \sim \mathcal{U}_{\{1,...,p\}}$.
  - Set $\theta_{-K}^{(i)} = \theta_{-K}^{(i-1)}$.
  - Sample $\theta_K^{(i)} \sim \pi(\theta_K \mid \theta_{-K}^{(i)})$

    where $\theta_{-K}^{(i)} = (\theta_1^{(i)}, ..., \theta_{K-1}^{(i)}, \theta_{K+1}^{(i)}, ..., \theta_p^{(i)})$

Random-Scan Gibbs Sampler

- Random scan Gibbs: Let $\theta^{(i)} = \left(\theta_1^{(i)}, \theta_2^{(i)}, ..., \theta_p^{(i)}\right)$ at step (iteration) i.

  - Draw *j* from *1* to *p* with probability $w_j = 1/p$

  - Draw new coordinate j, $\theta_j | \theta_{-j} \sim \pi(.|\theta_{-j})$ and leave the remaining components unchanged; that is, let

$$\theta_{-j}^{(i)} = \theta_{-j}^{(i-1)}$$

Gibbs Sampler: Example

- Consider the following bivariate target distribution:

$$\pi(x_1, x_2) = \mathcal{N}\left\{\begin{pmatrix}0\\0\end{pmatrix}, \begin{pmatrix}1 & \rho\\ \rho & 1\end{pmatrix}\right\} \propto \exp\left(-\frac{1}{2}\begin{pmatrix}x_1 & x_2\end{pmatrix}\frac{1}{1-\rho^2}\begin{bmatrix}1 & -\rho\\ -\rho & 1\end{bmatrix}\begin{pmatrix}x_1\\ x_2\end{pmatrix}\right)$$

- The marginal distribution is given as:

$$\pi(x_2) \propto \exp\left(-\frac{1}{2}x_2^2\right)$$

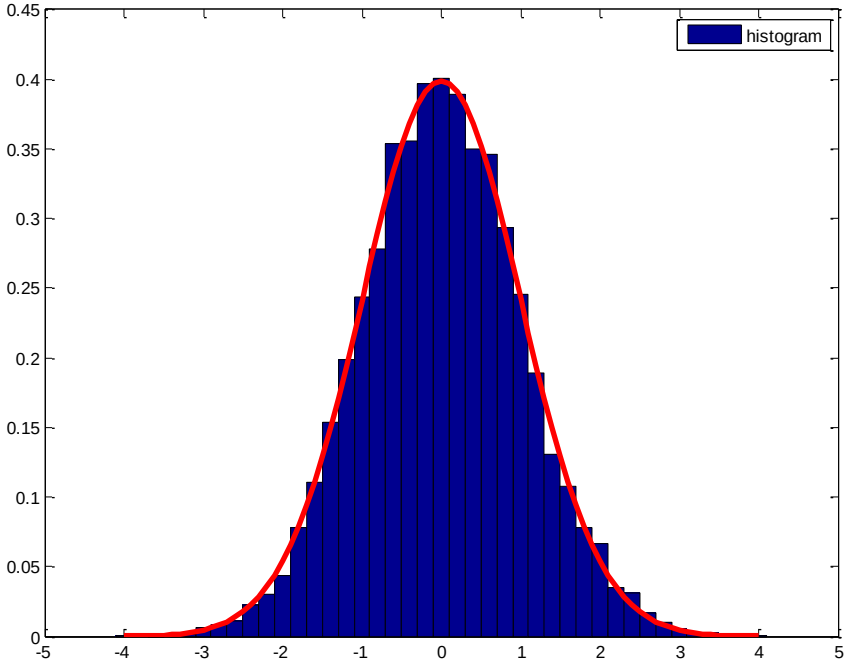
- A systematic-scan Gibbs sampler (see a C++ implementation) is generated with the following conditionals:

$$x_1^{t+1} \mid x_2^t \sim \mathcal{N}\left\{\rho x_2^t, 1-\rho^2\right\}$$

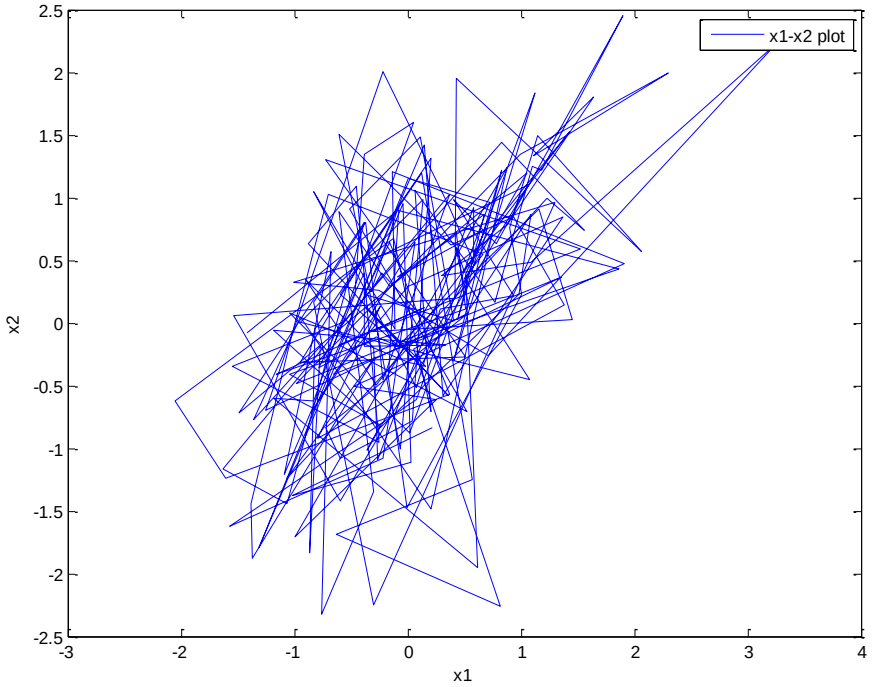
$$x_2^{t+1} \mid x_1^{t+1} \sim \mathcal{N}\left\{\rho x_1^{t+1}, 1-\rho^2\right\}$$

Gibbs Sampler: Example

- Set ρ=0.5, # of iterations 10000, and $(x_0, x_1)=(-3,-3)$



Histogram of x_1, the exact pdf of which is the standard Gaussian

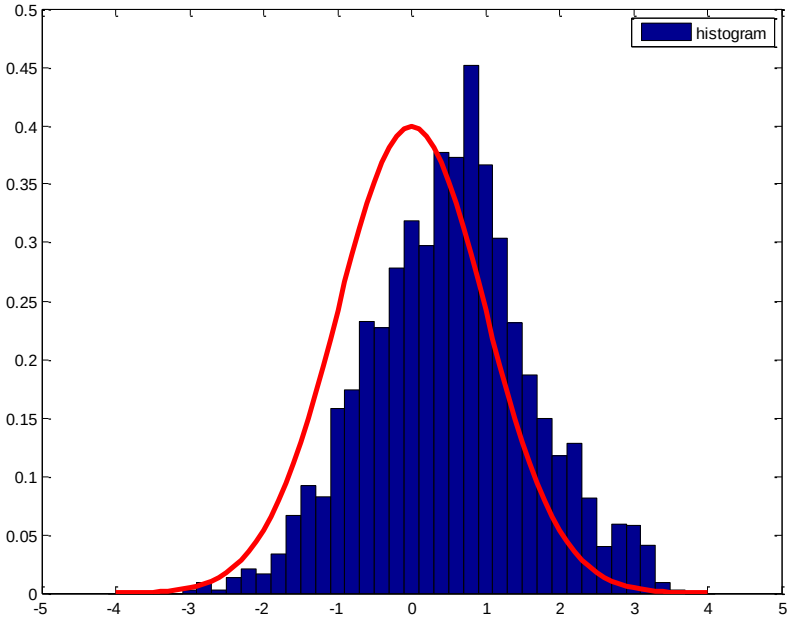


x_1-x_2 *plot*

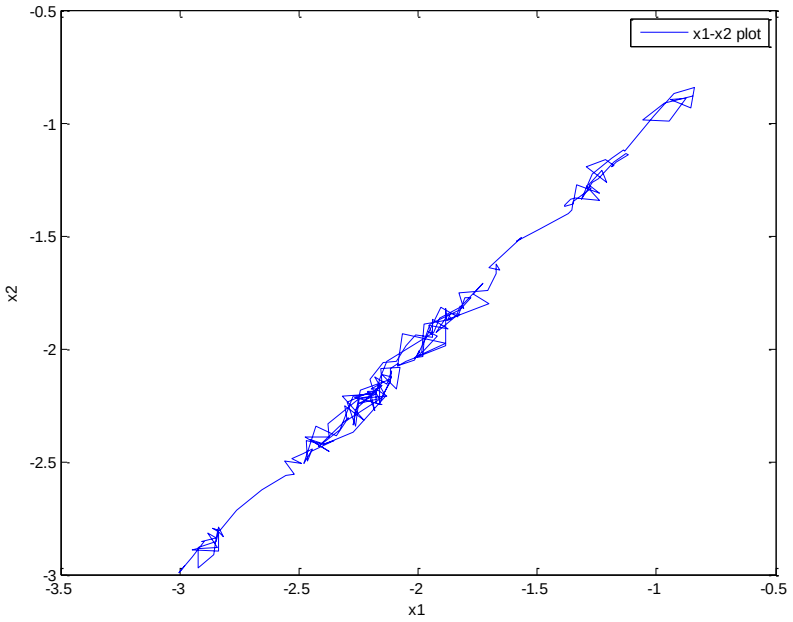
C++ programs are given here

Gibbs Sampler: Example

- Set ρ=0.999, # of iterations 10000, and $(x_0, x_1)=(-3,-3)$
- We can see that the sampling process in this case of highly correlated variables is inaccurate.



Histogram of x_1, the exact pdf of which is the standard Gaussian



x_1-x_2 *plot*

Convergence of the Gibbs Sampler

- Even when irreducibility and aperiodicity are ensured, the Gibbs sampler can still converge very slowly.

- Consider the target bivariate Gaussian distribution

$$\mathcal{N}(0, \begin{bmatrix} a & b \\ b & a \end{bmatrix})$$

- A systematic-scan Gibbs sampler is generated as

$$x_1^{t+1} \mid x_2^t \sim \mathcal{N}\left\{\frac{b}{a} x_2^t, a - \frac{b^2}{a}\right\}$$

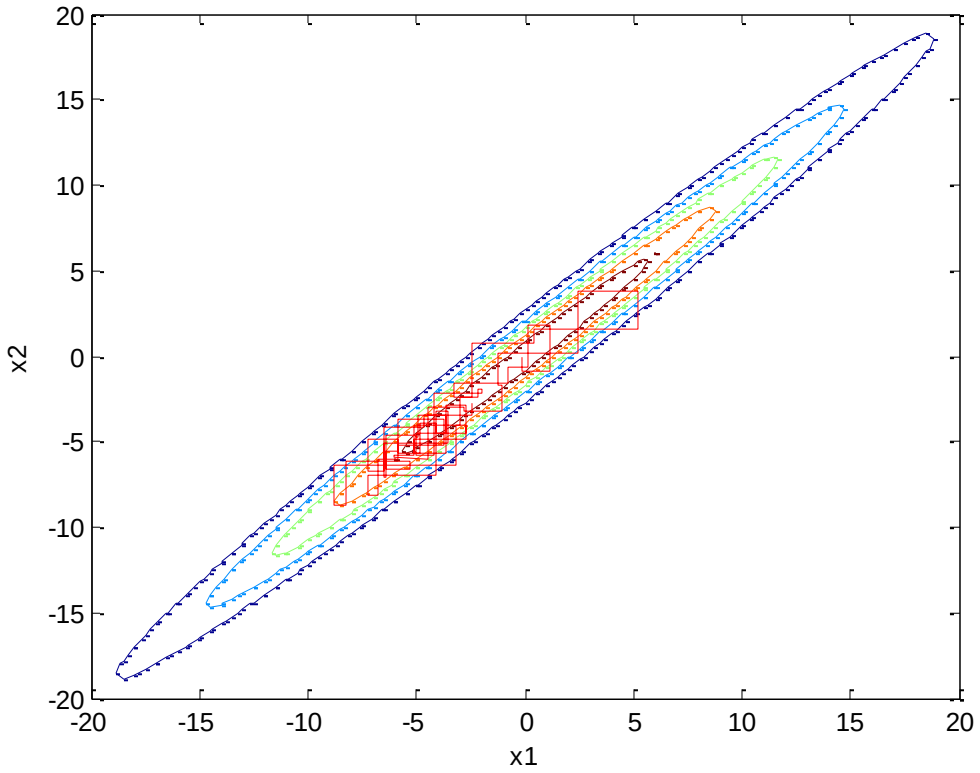
$$x_2^{t+1} \mid x_1^{t+1} \sim \mathcal{N}\left\{\frac{b}{a} x_1^{t+1}, a - \frac{b^2}{a}\right\}$$

- In this example, we set

$$\mathcal{N}(0, \begin{bmatrix} 100 & 99 \\ 99 & 100 \end{bmatrix})$$

Convergence of the Gibbs Sampler

- The Gibbs sampling path and equiprobability curves are plotted below.

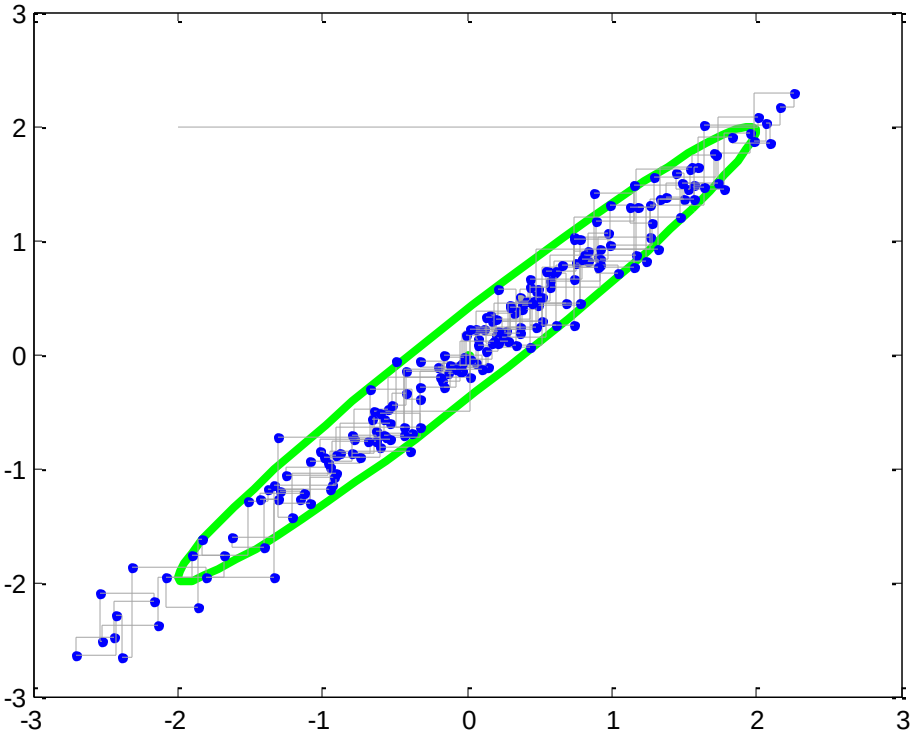


A C++ implementation can be found here

MatLab Implementation: Gibbs Sampler

- Consider the Gaussian $\pi(x_1, x_2) = \mathcal{N}\left(\mu, C\right)$. Following the conditionals shown earlier, it can be shown that the Gibbs sampler can proceed as follows:

$$x_1^{(t+1)} \leftarrow -\frac{C_{12}^{-1}}{C_{11}^{-1}} x_2^{(t)} + \frac{randn}{\sqrt{C_{11}^{-1}}},\ randn \sim \mathcal{N}(0,1),\ x_2^{(t+1)} \leftarrow -\frac{C_{12}^{-1}}{C_{11}^{-1}} x_1^{(t+1)} + \frac{randn}{\sqrt{C_{22}^{-1}}}$$

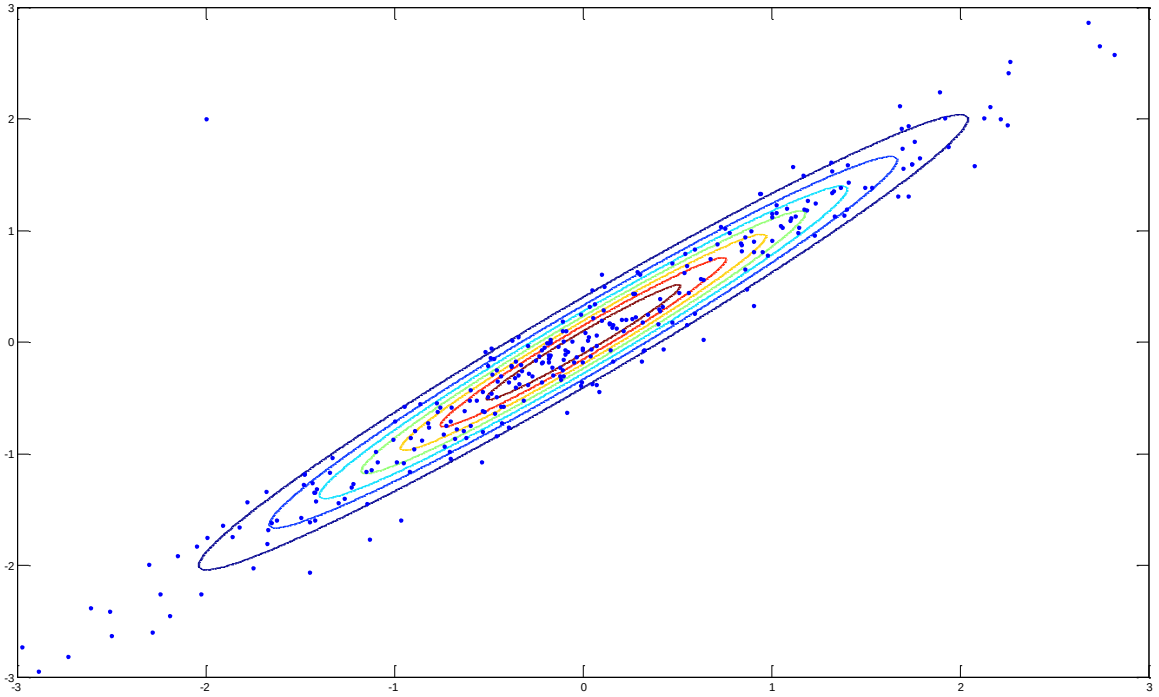


A MatLab implementation can be found here

MatLab Implementation: Gibbs Sampler

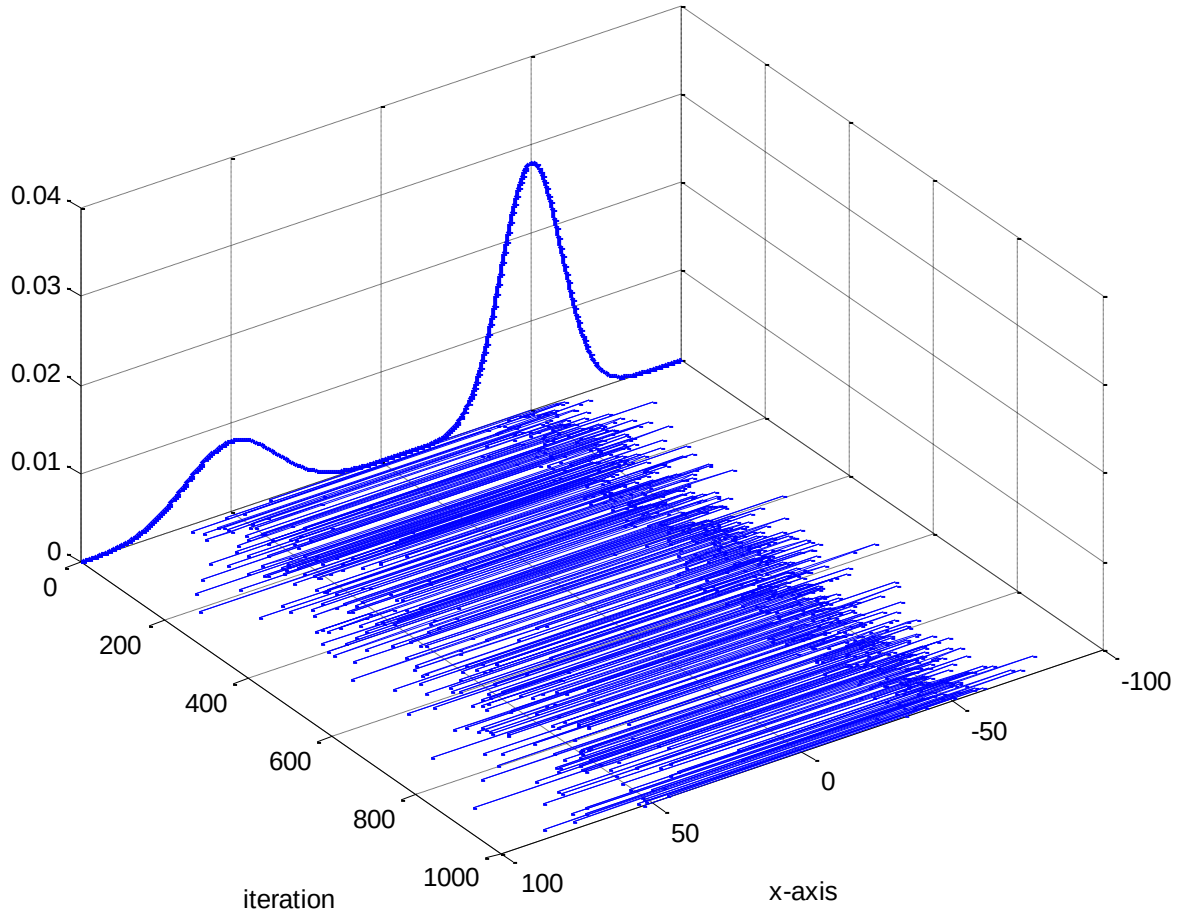
- Consider the Gaussian $\pi(x_1, x_2) = \mathcal{N}(\mu, C)$. Following the conditionals shown earlier, it can be shown that the Gibbs sampler can proceed as follows:

$$x_1^{(t+1)} \leftarrow -\frac{C_{12}^{-1}}{C_{11}^{-1}} x_2^{(t)} + \frac{randn}{\sqrt{C_{11}^{-1}}},\ randn \sim \mathcal{N}(0,1),\ x_2^{(t+1)} \leftarrow -\frac{C_{12}^{-1}}{C_{11}^{-1}} x_1^{(t+1)} + \frac{randn}{\sqrt{C_{22}^{-1}}}$$



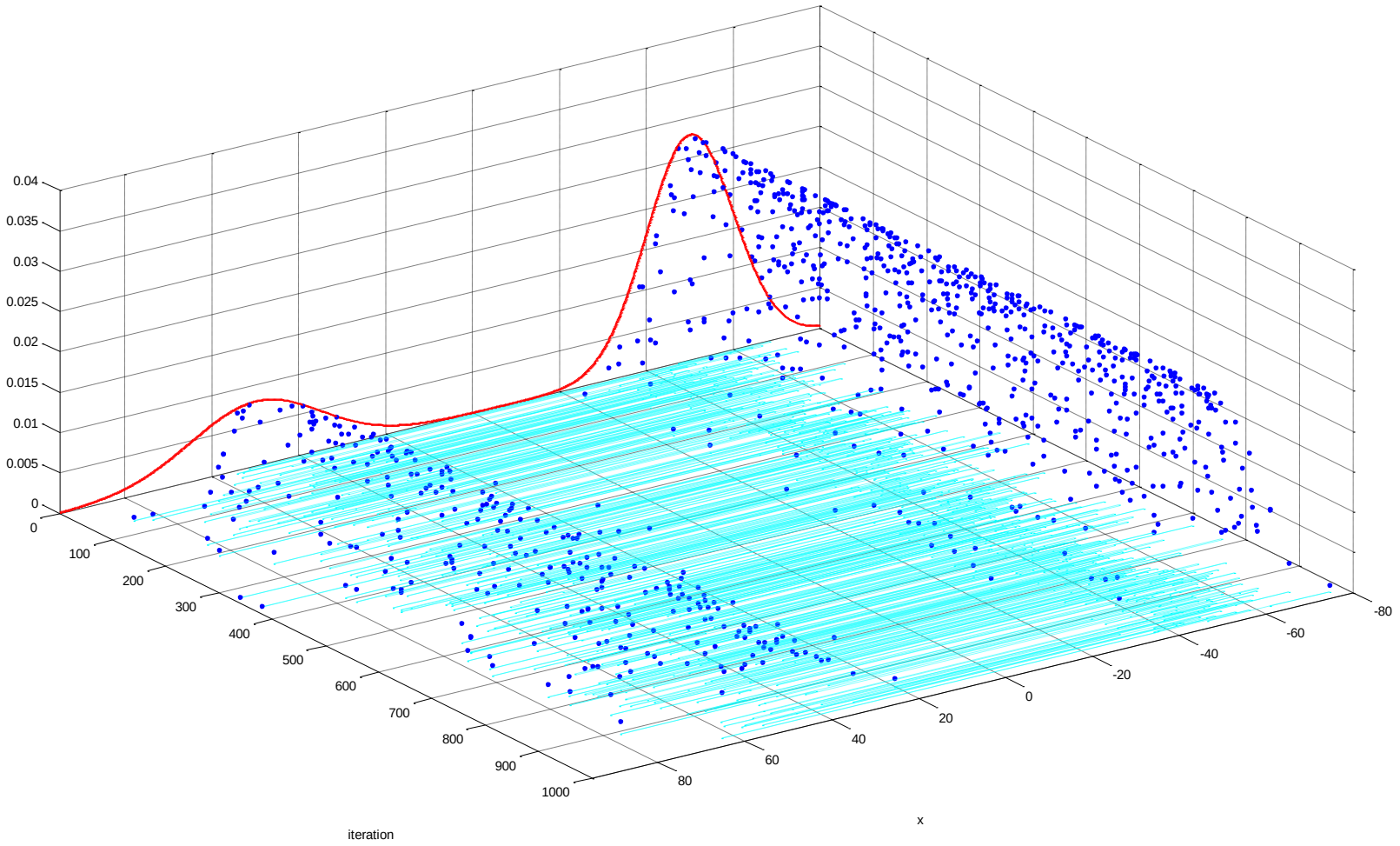
Another MatLab Implementation with movie frame animation can be found here.

Gibbs Sampler for Mixture of Gaussians



A MatLab implementation can be found here

Gibbs Sampler for Mixture of Gaussians



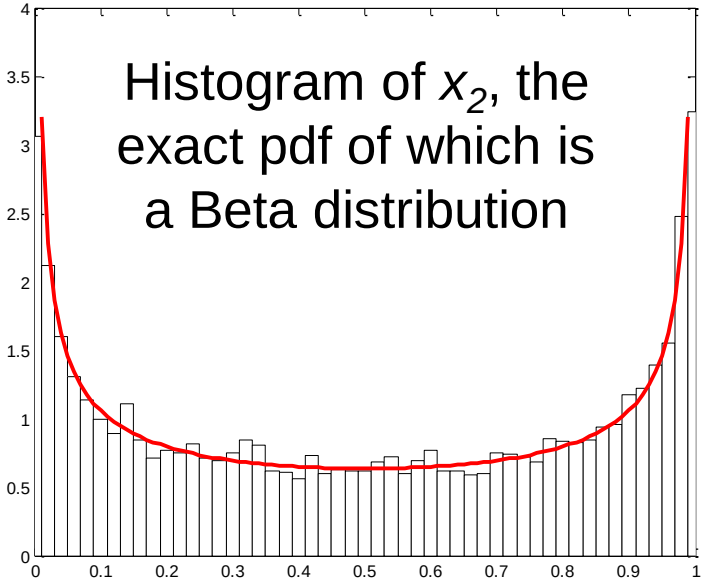
A MatLab implementation can be found here. This implementation works like a movie frame animation.

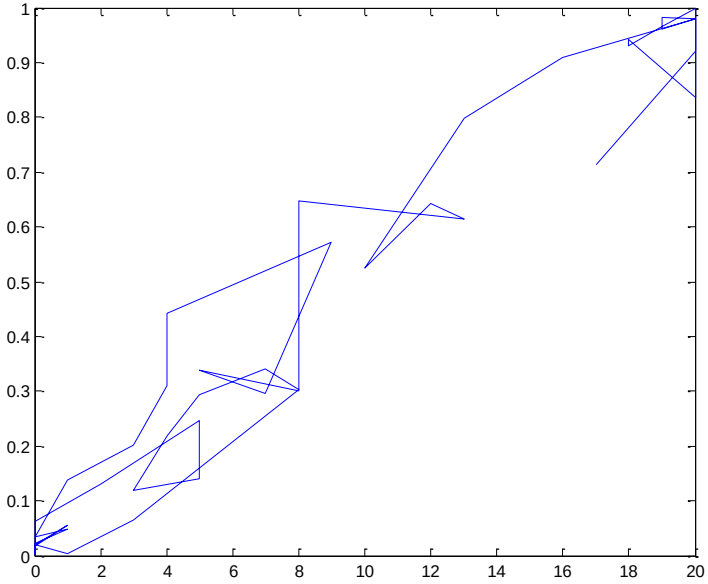
Gibbs Sampler: Example

- Consider the following target distribution $\pi(x_1, x_2) \sim \binom{n}{x_1} x_2^{x_1+\alpha+1}(1-x_2)^{n-x_1+\beta-1}$
- The two conditional distributions for the Gibbs sampler are

$$x_1 \mid x_2 \sim \mathcal{Binom}(n, x_2)$$
$$x_2 \mid x_1 \sim \mathcal{Be}(x_1+\alpha, n-x_1+\beta)$$

- We set $n = 20, \alpha = \beta = 0.5$, initial state (0,0), time of iterations 10000.
- See here for a C++ implementation and a MatLab implementation.





Gibbs Sampler: Example

- Consider a likelihood defined with the Cauchy distribution $\mathcal{C}(\mu,1)$ with two measurements as follows:

$$\ell(\mu \mid \mathcal{D}_n) = \prod_{i=1}^{n=2} f_\mu(x_i) = \frac{1}{\pi^2\left(1+(x_1-\mu)^2\right)\left(1+(x_2-\mu)^2\right)}$$

- We take as prior a normal distribution

$$\mu \sim \mathcal{N}(0,10)$$

- This leads to a posterior of the form:

$$\pi(\mu \mid \mathcal{D}) \sim \frac{e^{-\frac{\mu^2}{20}}}{\left(1+(x_1-\mu)^2\right)\left(1+(x_2-\mu)^2\right)}$$

- How do we use the Gibbs sampler to sample from this univariate distribution?

Gibbs Sampler: Example

$$\pi(\mu \mid \mathcal{D}) \sim \frac{e^{-\frac{\mu^2}{20}}}{\left(1+(x_1-\mu)^2\right)\left(1+(x_2-\mu)^2\right)}$$

- We can use Gibbs sampler by noticing:

$$\frac{1}{1+(x_i-\mu)^2} = \int_0^\infty e^{-\omega_i\left[1+(x_i-\mu)^2\right]} d\omega_i$$

- We can then think $\pi(\mu \mid \mathcal{D})$ as the marginal of $\pi(\mu, \omega_1, \omega_2 \mid \mathcal{D})$

$$\pi(\mu, \omega_1, \omega_2 \mid \mathcal{D}) \sim e^{-\frac{\mu^2}{20}} \prod_{i=1}^{2} e^{-\omega_i\left[1+(x_i-\mu)^2\right]}$$

- The Gibbs sampler is based on the following 2 steps:
  - Generate $\mu^{(t)} \sim \pi(\mu \mid \omega^{(t-1)}, \mathcal{D})$
  - Generate $\omega^{(t)} \sim \pi(\omega \mid \mu^{(t)}, \mathcal{D})$

Gibbs Sampler: Example

- The step $\mu^{(t)} \sim \pi(\mu|\omega^{(t-1)},\mathcal{D})$ is straight forward since

$$\pi\left(\mu \mid \omega, \mathcal{D}\right) \propto \mathcal{N}\left(\frac{\sum_i \omega_i x_i}{\sum_i \omega_i + 1/20}, \frac{1}{2\sum_i \omega_i + 1/10}\right)$$

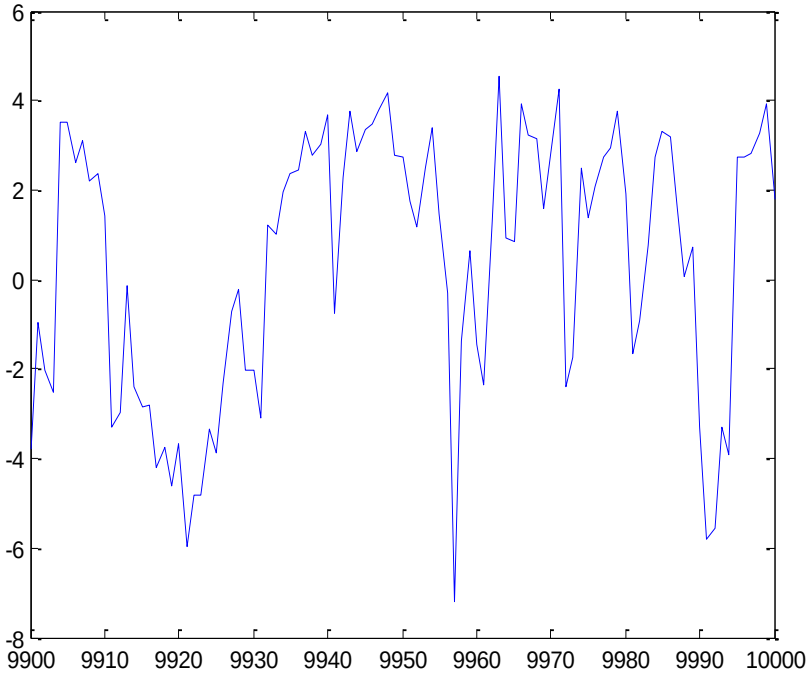
- The step $\omega^{(t)} \sim \pi(\omega|\mu^{(t)},\mathcal{D})$ is also straighrforward:

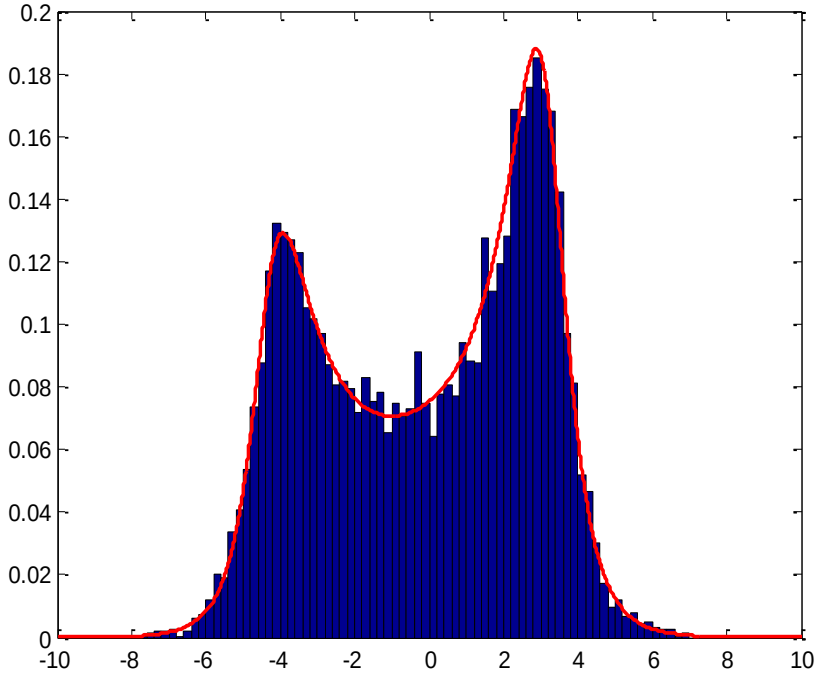
$$\pi\left(\omega \mid \mu^{(t)}, \mathcal{D}\right) \propto \mathcal{E}xp\left(1+(x_i-\mu^{(t)})^2\right)$$

- A MatLab implementation can be found here.

Gibbs sampler: Example

- On the left, the last 100 iterations of the chain ($\mu^{(t)}$); on the right, the histogram of the chain ($\mu^{(t)}$) and comparison with the target density for 10,000 iterations.





A MatLab implementation can be found here

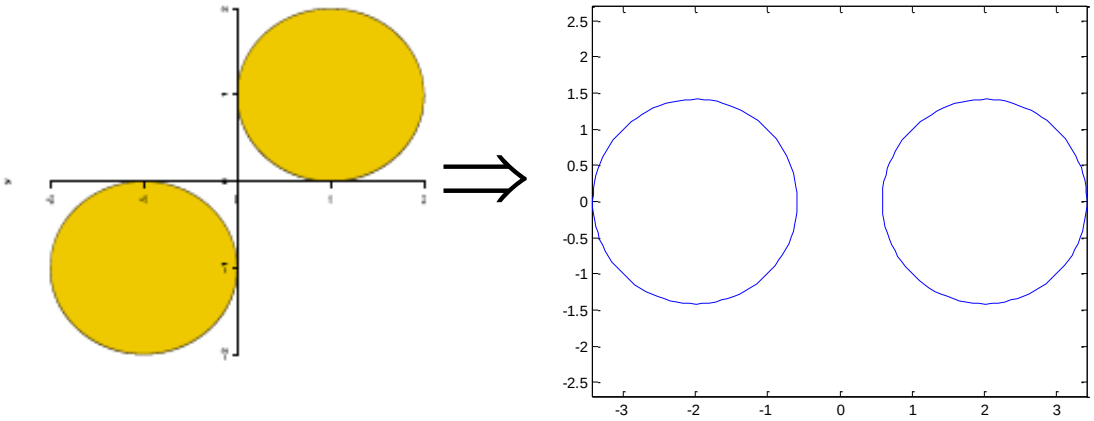
Block and Metropolized Gibbs

- Instead of updating single coordinates x_j, one can update blocks $\boldsymbol{x}_A$. This is more efficient but requires knowing the block conditionals $\pi(\boldsymbol{x}_A \mid \boldsymbol{x}_{-A})$ and being able to sample from them.
- Combinations of Gibbs and Metropolis Hastings (to be discussed in a follow up lecture) are popular.
  - In Metropolized Gibbs, for example, some coordinates are updated from conditionals and others using arbitrary proposals as in Metropolis-Hastings.
- Each transition kernel in Gibbs (which updates a single coordinate) is not irreducible nor aperiodic. However, their combination (random or systematic scan) might be!

Gibbs Sampling

- Consider a target $\pi(x_1, x_2)$ (e.g. a uniform distribution) with disconnected support as in the figure. Conditioning on $x_1 < 0$, the distribution of x_2 cannot produce a value in [0,1].
- You can make this type of problems to work by introducing a proper coordinate transformation.

$$y_1 = x_1 + x_2,\ y_2 = x_2 - x_1$$



Conditioning now on y_1 produces a uniform distribution on the union of a negative & of a positive interval. Therefore, one iteration of the Gibbs sampler is sufficient to jump from one disk to the other one.

Gibbs Sampler: Recommendation

- Have as few blocks as possible.
- Put the most correlated variables in the same block. If necessary, reparametrize the model to achieve this.
- Integrate analytically as many variables as possible.
- There is no general strategy that will work for all problems.

Bayesian Variable Selection in Regression

- We select the following regression model:

$$Y = \sum_{k=1}^{p} \beta_k X_k + \sigma V, \; where \; V \sim \mathcal{N}(0,1)$$

where we assume as priors $\mathcal{IG}\left(\sigma^2; \frac{\nu_0}{2}, \frac{\gamma_0}{2}\right)$ and for $\alpha^2 << 1$

$$\beta_k \sim \frac{1}{2}\mathcal{N}(0, \alpha^2\delta^2\sigma^2) + \frac{1}{2}\mathcal{N}(0, \delta^2\sigma^2)$$

- We introduce a latent variable $\gamma_k \in \{0,1\}$ such that:

$$\Pr(\gamma_k = 0) = \Pr(\gamma_k = 1) = \frac{1}{2}$$

$$\beta_k \mid \gamma_k = 0 \sim \mathcal{N}(0, \alpha^2\delta^2\sigma^2), \quad \beta_k \mid \gamma_k = 1 \sim \mathcal{N}(0, \delta^2\sigma^2)$$

A Bad Gibbs Sampler

- We have parameters $\left(\beta_{1:p}, \gamma_{1:p}, \sigma^2\right)$ and observe $\mathcal{D} = \left(x_i, y_i\right)_{i=1}^n$
- A potential Gibbs sampler consists of sampling iteratively from

$$p(\beta_{1:p} \mid \mathcal{D}, \gamma_{1:p}, \sigma^2)\,(Gaussian),\ p(\sigma^2 \mid \mathcal{D}, \gamma_{1:p}, \beta_{1:p})\,(inverse-Gamma),\ and$$

$$p(\gamma_{1:p} \mid \mathcal{D}, \beta_{1:p}, \sigma^2)$$

- In particular, $p(\gamma_{1:p} \mid \mathcal{D}, \beta_{1:p}, \sigma^2) = \prod_{k=1}^{p} p(\gamma_k \mid \mathcal{D}, \beta_k, \sigma^2)$ and

$$p(\gamma_k = 1 \mid \beta_k, \sigma^2) = \frac{\frac{1}{\sqrt{2\pi}\delta\sigma}\exp\left(-\frac{\beta_k^2}{2\delta^2\sigma^2}\right)}{\frac{1}{\sqrt{2\pi}\delta\sigma}\exp\left(-\frac{\beta_k^2}{2\delta^2\sigma^2}\right) + \frac{1}{\sqrt{2\pi}\alpha\delta\sigma}\exp\left(-\frac{\beta_k^2}{2\alpha^2\delta^2\sigma^2}\right)}$$

- The Gibbs sampler becomes reducible as α goes to zero.

Bayes Variable Selection

- This is the result of bad modeling. We consider $\alpha \simeq 0$ and write:

$$Y = \sum_{i=1}^{p} \gamma_k \beta_k X_k + \sigma V, \; where \; V \sim \mathcal{N}(0,1)$$

where $\gamma_k = 1$ if X_k is included or $\gamma_k = 0$ otherwise. However, this suggests that β_k is defined even when $\gamma_k = 0$.

- A neater way to write such models is

$$Y = \sum_{\{k:\gamma_k=1\}} \beta_k X_k + \sigma V = \beta_\gamma^T X_\gamma + \sigma V, \; where \; V \sim \mathcal{N}(0,1)$$

where, for a vector

$$\gamma = (\gamma_1, ..., \gamma_p), \; \beta_\gamma = \{\beta_k : \gamma_k = 1\}, \; X_\gamma = \{X_k : \gamma_k = 1\}, \; and \; n_\gamma = \sum_{k=1}^{p} \gamma_k$$

- Prior distributions

$$\pi_\gamma(\beta_\gamma, \sigma^2) = \mathcal{N}(\beta_\gamma; 0, \delta^2 \sigma^2 I_{n_\gamma}) \mathcal{IG}\left(\sigma^2; \frac{\nu_0}{2}, \frac{\gamma_0}{2}\right), \; and \; \pi(\gamma) = \prod_{k=1}^{p} \pi(\gamma_k) = 2^{-p}.$$

A Better Gibbs Sampler

- We are interested in sampling from the trans-dimensional distribution $\pi\left(\gamma, \beta_\gamma, \sigma^2 \mid \mathcal{D}\right)$
- However, we know that

$$\pi\left(\gamma, \beta_\gamma, \sigma^2 \mid \mathcal{D}\right) = \pi\left(\gamma \mid \mathcal{D}\right)\pi(\beta_\gamma, \sigma^2 \mid \mathcal{D}, \gamma)$$

where $\pi\left(\gamma \mid \mathcal{D}\right) \propto \pi\left(\mathcal{D} \mid \gamma\right)\pi(\gamma)$

and (see result from earlier lecture)

$$\pi(\mathcal{D} \mid \gamma) = \int \pi(\mathcal{D}, \beta_\gamma, \sigma^2 \mid \gamma) d\beta_\gamma d\sigma^2 \propto \Gamma(\frac{\nu_0 + n}{2}) \delta^{-n_\gamma} \left|\Sigma_\gamma\right|^{1/2} \left(\frac{\gamma_0 + \sum_{i=1}^{n} y_i^2 - \mu_\gamma^T \Sigma_\gamma^{-1} \mu_\gamma^T}{2} \right)^{-(\frac{\nu_0 + n}{2})}$$

with

$$\mu_\gamma = \Sigma_\gamma \left(\sum_{i=1}^{n} y_i x_{\gamma,i} \right), \Sigma_\gamma^{-1} = \delta^{-2} I_{n_\gamma} + \sum_{i=1}^{n} x_{\gamma,i} x_{\gamma,i}^T$$

A Better Gibbs Sampler

- The full conditional distribution for $\pi\left(\beta_\gamma, \sigma^2 \mid \mathcal{D}, \gamma\right)$ is

$$\pi_\gamma(\beta_\gamma, \sigma^2 \mid \mathcal{D}) = \mathcal{N}\left(\beta_\gamma; \mu_\gamma, \sigma^2 \Sigma_\gamma\right) \times$$

$$\mathcal{IG}\left(\sigma^2; \frac{\nu_0 + n}{2}, \frac{\gamma_0 + \sum_{i=1}^{n} y_i^2 - \mu_\gamma^T \Sigma_\gamma^{-1} \mu_\gamma^T}{2}\right)$$

where

$$\mu_\gamma = \Sigma_\gamma \left(\sum_{i=1}^{n} y_i x_{\gamma,i} \right), \Sigma_\gamma^{-1} = \delta^{-2} I_{n_\gamma} + \sum_{i=1}^{n} x_{\gamma,i} x_{\gamma,i}^T$$

- The derivation of the above conditional is already given in an earlier lecture.

A Better Gibbs Sampler

- Popular alternative prior models for γ_i include

$$\gamma_i \sim \mathcal{B}(\lambda), \text{ where } \lambda \sim \mathcal{U}[0,1]$$

$$\gamma_i \sim \mathcal{B}(\lambda_i), \text{ where } \lambda \sim \mathcal{B}e(\alpha, \beta)$$

- g-prior (Zellner)

$$\beta_\gamma \,|\, \sigma^2 \sim \mathcal{N}\left(\beta_\gamma; 0, \delta^2\sigma^2 (X_\gamma^T X_\gamma)^{-1}\right)$$

where here for robustness we additionally use

$$\delta^2 \sim \mathcal{IG}\left(\frac{a_0}{2}, \frac{b_0}{2}\right)$$

- Such variations in the priors are very important and can affect the performance of the Bayesian model.

Bayesian Variable Selection Example

- $\pi(\gamma|\mathcal{D})$ is a discrete probability distribution with 2^p potential values. We assume δ^2 is known here.
- We can use the Gibbs sampler to sample from it.
- Initialization:
  - Select deterministically or randomly $\gamma^{(0)} = (\gamma_1^{(0)}, \ldots, \gamma_p^{(0)})$

- Iteration i, i ≥ 1
  For k=1:p
  - Sample $\gamma_k^{(i)} \sim \pi(\gamma_k \mid \mathcal{D}, \gamma_{-k}^{(i)})$,

    where $\gamma_{-k}^{(i)} = (\gamma_1^{(i)}, \ldots, \gamma_{k-1}^{(i)}, \gamma_{k+1}^{(i-1)}, \ldots, \gamma_p^{(i-1)})$

  - Optional step: Sample

$$\left(\beta_\gamma^{(i)}, \sigma^{2(i)}\right) \sim \pi(\beta_\gamma, \sigma^2 \mid \mathcal{D}, \gamma^{(i)})$$

Bayesian Variable Selection Example

- Consider the case where δ^2 is unknown.
- Initialization:
  - Select deterministically or randomly $\left(\gamma^{(0)}, \beta_\gamma^{(0)}, \sigma^{2(0)}, \delta^{2(0)}\right)$
- Iteration i, i ≥ 1

  For k=1:p
  - Sample $\gamma_k^{(i)} \sim \pi(\gamma_k \mid \mathcal{D}, \gamma_{-k}^{(i)}, \delta^{2(i-1)})$

    where $\gamma_{-k}^{(i)} = (\gamma_1^{(i)}, \ldots, \gamma_{k-1}^{(i)}, \gamma_{k+1}^{(i-1)}, \ldots, \gamma_p^{(i-1)})$
  - Sample $\left(\beta_\gamma^{(i)}, \sigma^{2(i)}\right) \sim \pi(\beta_\gamma, \sigma^2 \mid \mathcal{D}, \gamma^{(i)}, \delta^{2(i)})$
  - Sample $\delta^{2(i)} \sim \pi(\delta^{2(i)} \mid \beta_\gamma^{(i)})$

Bayesian Variable Selection Example

- This very simple sampler is much more efficient than the ones where γ is sampled conditional upon (β, σ^2)

- However, it mixes very slowly because the components are updated one at a time.

- Updating correlated components together would increase significantly the convergence speed of the algorithm at the cost of an increased complexity.

- We provide further implementation details of the variable selection caterpillar example in this lecture.

Bayesian Variable Selection Example

- Top five most likely models for the selection models discussed:

$\pi(\gamma\|x)$ (Ridge $\delta^2 = 10$)	$\pi(\gamma\|x)$ (g-p $\delta^2 = 10$)	$\pi(\gamma\|x)$ (g-p, δ^2 estimated)
0,1,2,4,5/0.1946	0,1,2,4,5/0.2316	0,1,2,4,5/0.0929
0,1,2,4,5,9/0.0321	0,1,2,4,5,9/0.0374	0,1,2,4,5,9/0.0325
0,12,4,5,10/0.0327	0,1,9/0.0344	0,1,2,4,5,10/0.0295
0,1,2,4,5,7/0.0306	0,1,2,4,5,10/0.0328	0,1,2,4,5,7/0.0231
0,1,2,4,5,8/0.0251	0,1,4,5/0.0306	0,1,2,4,5,8/0.0228

Results from: Statistical Computing and MC Methods, A. Doucet.