

# Bayesian Learning

## Lecture 7 - Monte Carlo and Gibbs sampling

Mattias Villani

Department of Statistics  
Stockholm University

Department of Computer and Information Science  
Linköping University



# Lecture overview

- Monte Carlo simulation
- Gibbs sampling
- Data augmentation
  - ▶ Mixture models
  - ▶ Probit regression
- Regularized regression

# Monte Carlo sampling

- If  $\theta^{(1)}, \dots, \theta^{(N)}$  is an **iid sequence** from  $p(\theta)$ , then

$$\bar{\theta} = \frac{1}{N} \sum_{t=1}^N \theta^{(t)} \rightarrow E(\theta)$$

$$\bar{g}(\theta) = \frac{1}{N} \sum_{t=1}^N g(\theta^{(t)}) \rightarrow E[g(\theta)]$$

for some function  $g(\theta)$  of interest.

- $\mathbb{V}(\bar{g}(\theta)) = \frac{c}{N}$  for some constant  $c$ .
- Easy to compute **tail probabilities**  $\Pr(\theta \leq c)$  by letting

$$g(\theta) = I(\theta \leq c)$$

and

$$\frac{1}{N} \sum_{t=1}^N g(\theta^{(t)}) = \frac{\# \theta\text{-draws smaller than } c}{N}.$$

# Direct sampling by the inverse CDF method

■ Let  $F(x)$  be the CDF of  $X$ . **Inverse CDF method:**

1 Generate  $u$  from the uniform distribution on  $[0, 1]$ .

2 Compute  $x = F^{-1}(u)$ .

■ **Exponential distribution:**

$$u = F(x) = 1 - \exp(-\lambda x)$$

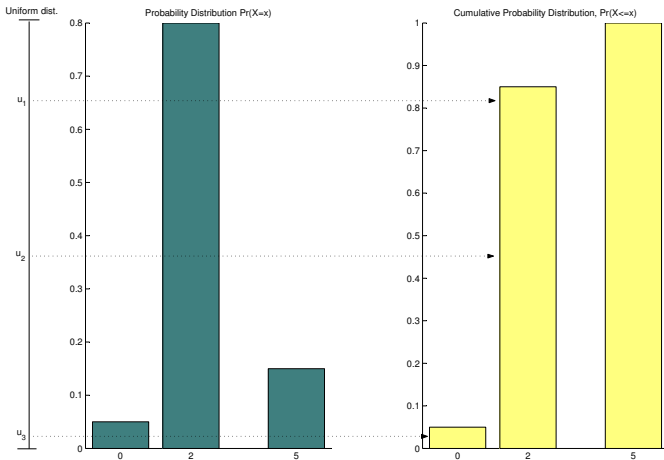
Inverting gives

$$x = -\ln(1 - u)/\lambda$$

■ So, if  $u \sim U(0, 1)$  then

$$x = -\ln(1 - u)/\lambda \sim \text{Expon}(\lambda)$$

# Inverse CDF method, discrete case



# Direct sampling by the inverse CDF method

## ■ Cauchy distribution:

$$f(x) = \frac{1}{\pi} \frac{1}{1+x^2}$$
$$u = F(x) = \frac{1}{2} + \frac{1}{\pi} \arctan(x)$$

Inverting ...

$$x = \tan[\pi(u - 1/2)].$$

## ■ Use relations:

$$y, z \text{ are indep } N(0, 1) \Rightarrow \frac{y}{z} \sim \text{Cauchy}(0, 1)$$

## ■ Chi-square. If $x_1, \dots, x_v \stackrel{iid}{\sim} N(0, 1)$ , then $\sum_{i=1}^v x_i^2 \sim \chi_v^2$ .

# Gibbs sampling

- Easily implemented methods for **sampling from multivariate distributions**,  $p(\theta_1, \dots, \theta_k)$ .
- Requirements: Easily sampled **full conditional distributions**:
  - ▶  $p(\theta_1|\theta_2, \theta_3, \dots, \theta_k)$
  - ▶  $p(\theta_2|\theta_1, \theta_3, \dots, \theta_k)$
  - ▶  $\vdots$
  - ▶  $p(\theta_k|\theta_1, \theta_2, \dots, \theta_{k-1})$
- Gibbs sampling is a special case of **Metropolis-Hastings** (see Lecture 8).
- Metropolis-Hastings is a **Markov Chain Monte Carlo (MCMC)** algorithm.

# The Gibbs sampling algorithm

- Choose initial values  $\theta_2^{(0)}, \theta_3^{(0)}, \dots, \theta_k^{(0)}$ .
- Repeat for  $j = 1, \dots, N$ :
  - ▶ Draw  $\theta_1^{(j)}$  from  $p(\theta_1 | \theta_2^{(j-1)}, \theta_3^{(j-1)}, \dots, \theta_k^{(j-1)})$
  - ▶ Draw  $\theta_2^{(j)}$  from  $p(\theta_2 | \theta_1^{(j)}, \theta_3^{(j-1)}, \dots, \theta_k^{(j-1)})$
  - ▶  $\vdots$
  - ▶ Draw  $\theta_k^{(j)}$  from  $p(\theta_k | \theta_1^{(j)}, \theta_2^{(j)}, \dots, \theta_{k-1}^{(j)})$
- Return draws:  $\theta^{(1)}, \dots, \theta^{(N)}$ , where  $\theta^{(j)} = (\theta_1^{(j)}, \dots, \theta_k^{(j)})$ .



# Gibbs sampling, cont.

- Gibbs draws  $\theta^{(1)}, \dots, \theta^{(N)}$  are **dependent**, but

$$\bar{\theta} = \frac{1}{N} \sum_{t=1}^N \theta_j^{(t)} \rightarrow E(\theta_j)$$

$$\bar{g}(\theta) = \frac{1}{N} \sum_{t=1}^N g(\theta^{(t)}) \rightarrow E[g(\theta)]$$

- $\theta^{(1)}, \dots, \theta^{(N)}$  **converges in distribution** to the target  $p(\theta)$ .
- $\theta_j^{(1)}, \dots, \theta_j^{(N)}$  converges to the marginal distribution of  $\theta_j$ .
- **Dependent draws**  $\rightarrow$  **less efficient** than iid sampling.
- **IID samples**:  $\theta^{(1)}, \dots, \theta^{(N)}$ :  $\text{Var}(\bar{\theta}) = \frac{\sigma^2}{N}$ .
- **Autocorrelated samples**:  $\text{Var}(\bar{\theta}) = \frac{\sigma^2}{N} (1 + 2 \sum_{k=1}^{\infty} \rho_k)$ , where  $\rho_k$  is the autocorrelation at lag  $k$ .
- **Inefficiency factor**:  $1 + 2 \sum_{k=1}^{\infty} \rho_k$ .

# Gibbs sampling bivariate normal

## ■ Joint distribution

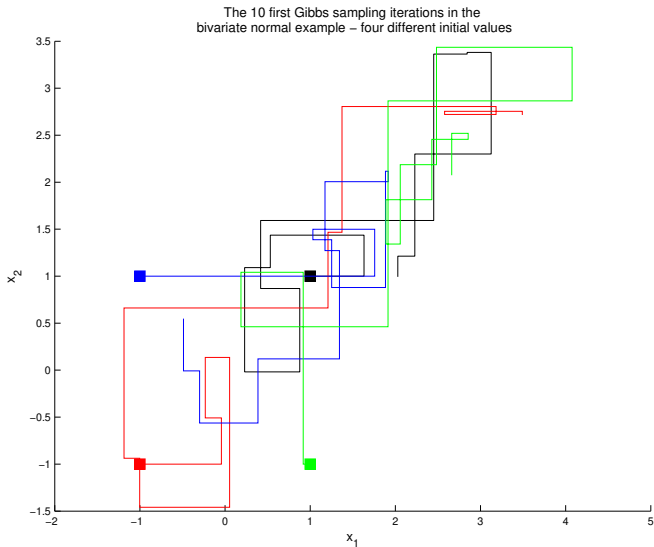
$$\begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix} \sim N_2 \left[ \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix} \right]$$

## ■ Full conditional posteriors

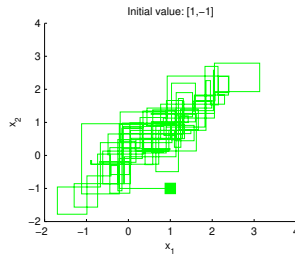
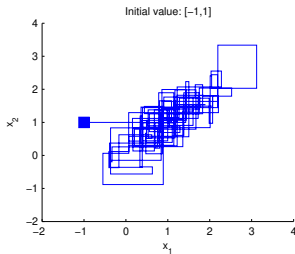
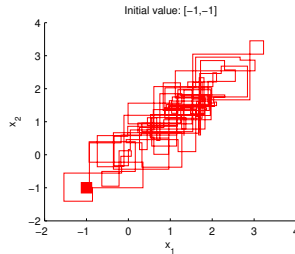
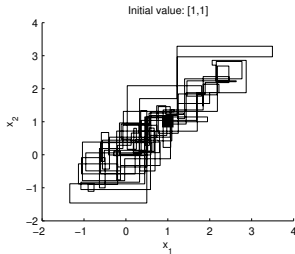
$$\theta_1 | \theta_2 \sim N[\mu_1 + \rho(\theta_2 - \mu_2), 1 - \rho^2]$$

$$\theta_2 | \theta_1 \sim N[\mu_2 + \rho(\theta_1 - \mu_1), 1 - \rho^2]$$

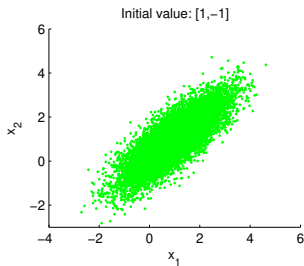
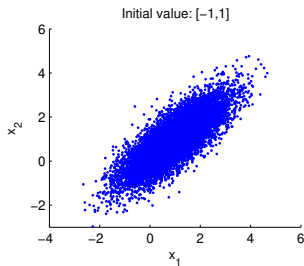
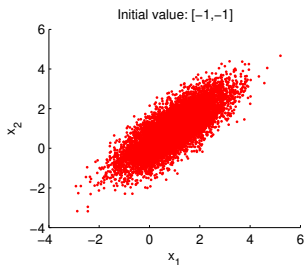
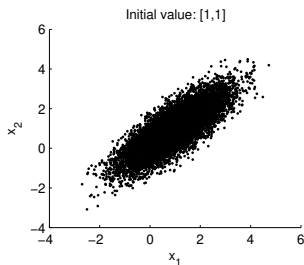
# Gibbs sampling - Bivariate normal



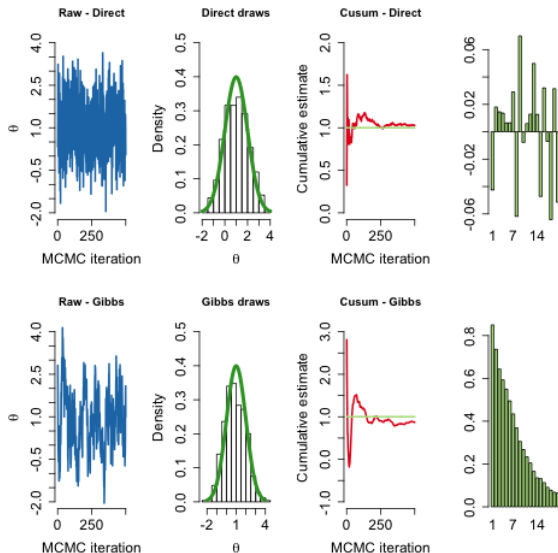
# Gibbs sampling - Bivariate normal



# Gibbs sampling - Bivariate normal



# Direct sampling vs Gibbs sampling

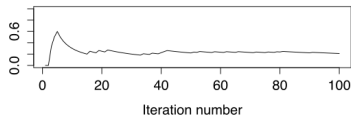


# Estimating $Pr(\theta_1 > 0, \theta_2 > 0)$

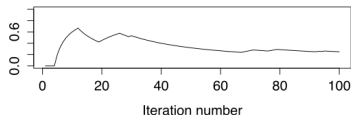
- Joint probability by counting:

$$Pr(\theta_1 > 0, \theta_2 > 0) \approx N^{-1} \sum_{i=1}^N 1(\theta_1^{(i)} > 0, \theta_2^{(i)} > 0)$$

**Direct draws**



**Gibbs draws**



# Normal model with conditionally conjugate prior

## ■ Normal model with conditionally conjugate prior

$$\begin{aligned}\mu &\sim N(\mu_0, \tau_0^2) \\ \sigma^2 &\sim \text{Inv} - \chi^2(\nu_0, \sigma_0^2)\end{aligned}$$

## ■ Full conditional posteriors

$$\begin{aligned}\mu | \sigma^2, x &\sim N(\mu_n, \tau_n^2) \\ \sigma^2 | \mu, x &\sim \text{Inv} - \chi^2\left(\nu_n, \frac{\nu_0 \sigma_0^2 + \sum_{i=1}^n (x_i - \mu)^2}{n + \nu_0}\right)\end{aligned}$$

with  $\mu_n$  and  $\tau_n^2$  defined the same as when  $\sigma^2$  is known.



# Gibbs sampling for AR processes

## ■ AR( $p$ ) process

$$x_t = \mu + \phi_1(x_{t-1} - \mu) + \dots + \phi_p(x_{t-p} - \mu) + \varepsilon_t, \quad \varepsilon_t \stackrel{iid}{\sim} N(0, \sigma^2).$$

■ Let  $\phi = (\phi_1, \dots, \phi_p)'$ .

## ■ Prior:

- ▶  $\mu \sim \text{Normal}$
- ▶  $\phi \sim \text{Multivariate Normal}$
- ▶  $\sigma^2 \sim \text{Scaled Inverse } \chi^2$ .

■ The **posterior** can be simulated by **Gibbs sampling**<sup>1</sup>:

- ▶  $\mu | \phi, \sigma^2, x \sim \text{Normal}$
- ▶  $\phi | \mu, \sigma^2, x \sim \text{Multivariate Normal}$
- ▶  $\sigma^2 | \mu, \phi, x \sim \text{Scaled Inverse } \chi^2$

---

<sup>1</sup>Villani (2009). Steady State Priors for Vector Autoregressions. *Journal of Applied Econometrics*.

# Data augmentation - Mixture distributions

■ Let  $\phi(x|\mu, \sigma^2)$  denote the **PDF** of  $x \sim N(\mu, \sigma^2)$ .

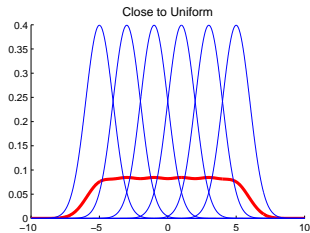
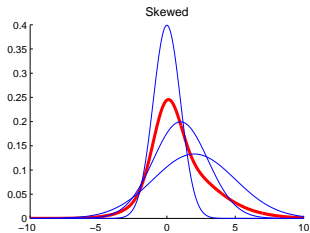
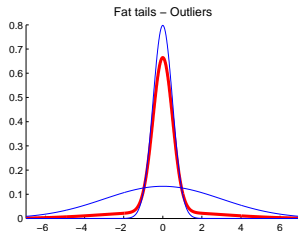
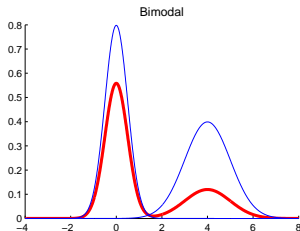
■ Two-component **mixture of normals** [MN(2)]

$$p(x) = \pi \cdot \phi(x|\mu_1, \sigma_1^2) + (1 - \pi) \cdot \phi(x|\mu_2, \sigma_2^2)$$

■ **Simulate** from a MN(2):

- ▶ Simulate a **membership indicator**  $I \in \{1, 2\}$ :  $I \sim \text{Bern}(\pi)$ .
- ▶ If  $I = 1$ , simulate  $x$  from  $N(\mu_1, \sigma_1^2)$
- ▶ If  $I = 2$ , simulate  $x$  from  $N(\mu_2, \sigma_2^2)$ .

# Illustration of mixture distributions



# Mixture distributions, cont.

- The **likelihood** is a product of sums. **Messy** to work with.
- **Assume** that we know where each observation comes from

$$l_i = \begin{cases} 1 & \text{if } x_i \text{ came from Density 1} \\ 2 & \text{if } x_i \text{ came from Density 2} \end{cases}.$$

- Given  $l_1, \dots, l_n$  it is easy to estimate  $\pi, \mu_1, \sigma_1^2, \mu_2, \sigma_2^2$  by separating the sample according to the  $l$ 's.
- But we do **not** know  $l_1, \dots, l_n$ !
- **Data augmentation**: add  $l_1, \dots, l_n$  as unknown parameters.
- **Gibbs sampling**:
  - ▶ Sample  $\pi, \mu_1, \sigma_1^2, \mu_2, \sigma_2^2$  **given**  $l_1, \dots, l_n$
  - ▶ Sample  $l_1, \dots, l_n$  **given**  $\pi, \mu_1, \sigma_1^2, \mu_2, \sigma_2^2$

# Gibbs sampling for mixture distributions

- Prior:  $\pi \sim \text{Beta}(\alpha_1, \alpha_2)$ . Conjugate prior for  $(\mu_j, \sigma_j^2)$ , see L5.
- Define:  $n_1 = \sum_{i=1}^n (l_i = 1)$  and  $n_2 = n - n_1$ .
- **Gibbs sampling:**
  - ▶  $\pi \mid \mathbf{l}, \mathbf{x} \sim \text{Beta}(\alpha_1 + n_1, \alpha_2 + n_2)$
  - ▶  $\sigma_1^2 \mid \mathbf{l}, \mathbf{x} \sim \text{Inv-}\chi^2(\nu_{n_1}, \sigma_{n_1}^2)$  and  $\mu_1 \mid \mathbf{l}, \sigma_1^2, \mathbf{x} \sim N(\mu_{n_1}, \tau_{n_1}^2)$
  - ▶  $\sigma_2^2 \mid \mathbf{l}, \mathbf{x} \sim \text{Inv-}\chi^2(\nu_{n_2}, \sigma_{n_2}^2)$  and  $\mu_2 \mid \mathbf{l}, \sigma_2^2, \mathbf{x} \sim N(\mu_{n_2}, \tau_{n_2}^2)$
  - ▶  $l_i \mid \pi, \mu_1, \sigma_1^2, \mu_2, \sigma_2^2, \mathbf{x} \sim \text{Bern}(\theta_i), i = 1, \dots, n,$

$$\theta_i = \frac{(1 - \pi)\phi(x_i; \mu_2, \sigma_2^2)}{\pi\phi(x_i; \mu_1, \sigma_1^2) + (1 - \pi)\phi(x_i; \mu_2, \sigma_2^2)}.$$

# Gibbs sampling for mixture distributions

## ■ $K$ -component mixture of normals

$$p(x) = \sum_{k=1}^K \pi_k \phi(x; \mu_k, \sigma_k^2)$$

■ **Multi-class indicators:**  $I_i = k$  if  $x_i$  comes from component  $k$ .

## ■ Gibbs sampling

- ▶  $(\pi_1, \dots, \pi_K) \mid \mathbf{I}, \mathbf{x} \sim \text{Dirichlet}(\alpha_1 + n_1, \alpha_2 + n_2, \dots, \alpha_K + n_K)$
- ▶  $\sigma_k^2 \mid \mathbf{I}, \mathbf{x} \sim \text{Inv-}\chi^2$  and  $\mu_k \mid \mathbf{I}, \sigma_k^2, \mathbf{x} \sim \text{Normal}$ , for  $k = 1, \dots, K$ ,
- ▶  $I_i \mid \pi, \mu, \sigma^2, \mathbf{x} \sim \text{Multinomial}(\theta_{i1}, \dots, \theta_{iK})$ , for  $i = 1, \dots, n$ ,

$$\theta_{ij} = \frac{\pi_j \phi(x_i; \mu_j, \sigma_j^2)}{\sum_{r=1}^K \pi_r \phi(x_i; \mu_r, \sigma_r^2)}.$$

■ Gibbs sampling is very powerful for **missing data** problems.

■ **Semi-supervised learning.**

# Data augmentation - Probit regression

## ■ Probit regression:

$$\Pr(y_i = 1 \mid x_i) = \Phi(x_i^T \beta)$$

## ■ Random utility formulation:

$$\begin{aligned} u_i &\sim N(x_i^T \beta, 1) \\ y_i &= \begin{cases} 1 & \text{if } u_i > 0 \\ 0 & \text{if } u_i \leq 0 \end{cases} \end{aligned}$$

- Check:  $\Pr(y_i = 1 \mid x_i) = \Pr(u_i > 0) = 1 - \Pr(u_i \leq 0) = 1 - \Pr(u_i - x_i^T \beta < -x_i^T \beta) = 1 - \Phi(-x_i^T \beta) = \Phi(x_i^T \beta)$ .
- Given  $u = (u_1, \dots, u_n)$ ,  $\beta$  can be analyzed by linear regression.
- $u$  is **not observed**. Gibbs sampling to the rescue!<sup>2</sup>

---

<sup>2</sup>Albert and Chib (1993). Bayesian Analysis of Binary and Polychotomous Response Data. *JASA*.

# Gibbs sampling for the Probit regression

- Simulate from **joint posterior**  $p(u, \beta|y)$  by iterating between
  - ▶  $p(\beta|u, y)$  is multivariate normal (linear regression)
  - ▶  $p(u_i|\beta, y)$ ,  $i = 1, \dots, n$ .
- The **full conditional** posterior distribution of  $u_i$

$$\begin{aligned} p(u_i|\beta, y) &\propto p(y_i|\beta, u_i)p(u_i|\beta) \\ &= \begin{cases} N(u_i|x_i'\beta, 1) & \text{truncated to } u_i \in (-\infty, 0] \text{ if } y_i = 0 \\ N(u_i|x_i'\beta, 1) & \text{truncated to } u_i \in (0, \infty) \text{ if } y_i = 1 \end{cases} \end{aligned}$$

- Histogram of  $\beta$ -draws approximates the marginal posterior of  $\beta$

$$p(\beta|y) = \int p(u, \beta|y) du$$



# Gibbs sampling for Regularized regression

- Recap: The joint posterior of  $\beta$ ,  $\sigma^2$  and  $\lambda$  is

$$\beta|\sigma^2, \lambda, \mathbf{y}, \mathbf{X} \sim N(\mu_n, \Omega_n^{-1})$$

$$\sigma^2|\lambda, \mathbf{y}, \mathbf{X} \sim \text{Inv} - \chi^2(\nu_n, \sigma_n^2)$$

$$p(\lambda|\mathbf{y}, \mathbf{X}) \propto \sqrt{\frac{|\Omega_0|}{|\mathbf{X}'\mathbf{X} + \Omega_0|}} \left(\frac{\nu_n \sigma_n^2}{2}\right)^{-\nu_n/2} \cdot p(\lambda)$$

- This is the **conditional-marginal decomposition**

$$p(\beta, \sigma^2, \lambda|\mathbf{y}, \mathbf{X}) = p(\beta|\sigma^2, \lambda, \mathbf{y}, \mathbf{X})p(\sigma^2|\lambda, \mathbf{y}, \mathbf{X})p(\lambda|\mathbf{y}, \mathbf{X})$$

- **Gibbs sampling** can instead be used:

- ▶ Sample  $\beta|\sigma^2, \lambda, \mathbf{y}, \mathbf{X}$  from Normal
- ▶ Sample  $\sigma^2|\beta, \lambda, \mathbf{y}, \mathbf{X}$  from  $\text{Inv} - \chi^2$
- ▶ Sample  $\lambda|\beta, \sigma^2, \mathbf{y}, \mathbf{X}$  from Gamma

- $\lambda$  is **easy** to simulate **conditional on**  $\beta$  and  $\sigma^2$ .

# Gibbs sampling for Regularized regression

- Assume a Gamma prior for  $\lambda$  (same as  $\lambda^{-1} \sim \text{Inv} - \chi^2$ )

$$\lambda \sim \text{Gamma} \left( \frac{\eta_0}{2}, \frac{\eta_0}{2\lambda_0} \right).$$

- $\mathbb{E}(\lambda) = \frac{\eta_0/2}{\eta_0/(2\lambda_0)} = \lambda_0$  and  $\mathbb{V}(\lambda) = \frac{\eta_0/2}{(\eta_0/(2\lambda_0))^2} = \frac{1}{2\eta_0\lambda_0^2}.$

- Using Bayes' theorem twice:

$$\begin{aligned} p(\lambda | \beta, \sigma^2, \mathbf{y}) &\propto p(\mathbf{y} | \beta, \sigma^2, \lambda) p(\lambda | \beta, \sigma^2) \\ &\propto p(\beta | \sigma^2, \lambda) p(\lambda | \sigma^2) \\ &\propto p(\beta | \sigma^2, \lambda) p(\lambda) \end{aligned}$$

- Note:

- ▶ likelihood  $p(\mathbf{y} | \beta, \sigma^2, \lambda)$  does not depend on  $\lambda$ .
- ▶ prior  $p(\lambda | \sigma^2)$  is assumed to not depend on  $\sigma^2$ .

# Gibbs sampling for Regularized regression

## ■ Full conditional posterior

$$\begin{aligned} p(\lambda | \beta, \sigma^2, \mathbf{y}) &\propto p(\beta | \sigma^2, \lambda) p(\lambda) \\ &\propto \prod_{i=1}^m \frac{1}{\sqrt{2\pi\sigma^2/\lambda}} \exp\left(-\frac{\beta_i^2}{2\sigma^2/\lambda}\right) \cdot \lambda^{\eta_0/2-1} \exp\left(-\lambda \frac{\eta_0}{2\lambda_0}\right) \\ &\propto \lambda^{m/2} \exp\left(-\frac{\lambda}{2\sigma^2} \sum_{i=1}^m \beta_i^2\right) \cdot \lambda^{\eta_0/2-1} \exp\left(-\lambda \frac{\eta_0}{2\lambda_0}\right) \\ &\propto \lambda^{(m+\eta_0)/2-1} \exp\left(-\lambda \left(\frac{\sigma^{-2} \sum_{i=1}^m \beta_i^2 + \eta_0/\lambda_0}{2}\right)\right) \end{aligned}$$

## ■ This shows that

$$\lambda | \beta, \sigma^2, \mathbf{y} \sim \text{Gamma}\left(\frac{m + \eta_0}{2}, \frac{\sigma^{-2} \sum_{i=1}^m \beta_i^2 + \eta_0/\lambda_0}{2}\right).$$

## ■ $\mathbb{E}(\lambda | \beta, \sigma^2, \mathbf{y}) = \frac{m+\eta_0}{\sigma^{-2} \sum_{i=1}^m \beta_i^2 + \eta_0/\lambda_0}$ , so $\lambda$ is learned from variability of the $\beta_i$ . Large $m$ helps!

# Improving the efficiency of the Gibbs sampler

- **Efficient blocking.** Correlated parameters should ideally be included in the same updating block.
- **Reparametrization.** Convergence can improve dramatically in alternative parametrizations.
- **Data augmentation.**
  - ▶ Augment with latent variables to make **full conditional posteriors more easily sampled** (Probit, Mixture models).
  - ▶ But typically **increases the autocorrelation** between draws.