Statistical Methods for Data Science

Semester 1 2021

DATA7202

Bayesian Models

Slava Vaisman

The University of Queensland

r.vaisman@uq.edu.au

1. Bayesian inference
2. Conjugate priors
3. Sampling from the posterior
4. Markov Chain Monte Carlo
5. Bayesian regression
    - Normal Linear Regression
    - Linear Regression with t errors

A different approach to inference (1)

- Given a data $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)$, we would like to build a function $g(\boldsymbol{x})$ that predicts the response variable y. Suppose that $y \in \mathbb{R}$.
- Numerous functions are of course possible, but for now, let us consider a (restricted) class of linear functions, namely:

$$\mathcal{H} = \{\boldsymbol{x}^\top \boldsymbol{\beta}\}.$$

- We already know what is the reason for introducing a restricted class of functions (inductive bias).
- In addition, such model is highly interpretable as we saw earlier.

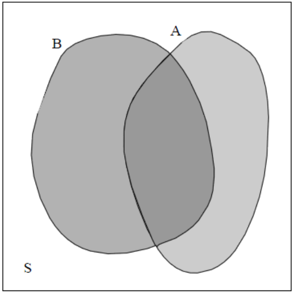
A different approach to inference (2)

- The class $\mathcal{H}$ is very structured.
- Specifically, we need to estimate the set of parameters $\boldsymbol{\beta} = (\beta_0, \ldots, \beta_d)$ (suppose that $\boldsymbol{x}$ is d-dimensional).
- Make inference about $\boldsymbol{\beta}$ (confidence intervals), and perform hypothesis testing ($\beta_i = 0$?).

- Recall that we used maximum likelihood methods to estimate parameters.
- The major question that we are going to answer is: Can a parameter space (such as $\boldsymbol{\beta}$), be estimated from the data, using different techniques?

The Bayes' rule (1)

Let A and B be events and consider how probabilities change when we know some event $B \subset S$ has occurred.



1. Suppose B has occurred. Thus, we know that the outcome lies in B.
2. Then, A will occur if and only if $A \cap B$ occurs, and the relative chance of A occurring is therefore $\mathbb{P}(A \cap B)/\mathbb{P}(B)$.

The Bayes' rule (2)

Definition

This leads to the definition of the conditional probability of A given B:

$$\mathbb{P}(A \mid B) = \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(B)}. \tag{1}$$

Of course, we can also write:

$$\mathbb{P}(B \mid A) = \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(A)}. \tag{2}$$

From (1) and (2) we arrive at the Bayes' rule. Namely, the $\mathbb{P}(A \mid B)\, \mathbb{P}(B) = \mathbb{P}(B \mid A)\, \mathbb{P}(A)$ leads to

$$\mathbb{P}(B \mid A) = \frac{\mathbb{P}(A \mid B)\, \mathbb{P}(B)}{\mathbb{P}(A)}. \tag{3}$$

Major ideas (1)

- It is convenient to think about a regression model.
- Let y be a data matrix (or vector) and θ be a vector which contains the parameters for a model that we are going to use in order to explain the data y.
- Clearly, we are interested in learning about θ based on the observed data, y. By replace B by θ and A by y in (3) to obtain:

$$\mathbb{P}(\theta \mid y) = \frac{\mathbb{P}(y \mid \theta)\, \mathbb{P}(\theta)}{\mathbb{P}(y)}. \tag{4}$$

Major ideas (2)

- The $\mathbb{P}(\theta \mid y)$ term is the *posterior* density, the p.d.f., is of fundamental interest. This is the conditional distribution of the random variable θ given the data y. It answers the question: "Given the data, what do we know about θ?"
- The $\mathbb{P}(y \mid \theta)$ term is the *likelihood function* ($L(\theta)$). It is the density of the data y conditional on the parameters of the model θ.
- The *prior*, $\mathbb{P}(\theta)$ term contains any non-data information available about θ. Note that the prior does not depend upon the data. The prior summarizes our believes about the model parameters before we observe the data.
- The $\mathbb{P}(y)$ term is the *marginal density* (or *marginal likelihood*) of the observed data. Note that

$$\mathbb{P}(y) = \int_{\Theta} \mathbb{P}(\theta)\, L(\theta)\, d\theta,$$

Nuisance parameters

Definition (Nuisance parameters)

A *nuisance parameter* is any parameter which is not of immediate interest but which must be accounted for in the analysis of those parameters which are of interest.

In many practical situations, not all elements of the vector θ are of direct interest. Suppose that $\theta_1 \in \Theta_1$ and $\theta_2 \in \Theta_2$ are the mean and the variance of a sampling distribution, respectively. Then $\theta = [\theta_1, \theta_2] \in \Theta_1 \times \Theta_2$. The Bayesian approach allows to eliminate nuisance parameters from the problem. We can just integrate the nuisance parameter out of the posterior density. This will, yield the marginal posterior density for the parameters of interest. Namely, with the posterior $\mathbb{P}([\theta_1, \theta_2] \mid y)$ in hand, we get

$$\mathbb{P}(\theta_1 \mid y) = \int_{\Theta_2} \mathbb{P}([\theta_1, \theta_2] \mid y)\, d\theta_2.$$

Conjugate Bernoulli analysis example (1)

- Given the parameter θ, where $0 < \theta < 1$, consider n iid Bernoulli random variables Y_i, $(i = 1, \ldots, n)$, each with p.m.f.

$$\mathbb{P}(y \mid \theta) = \begin{cases} \theta & y = 1 \\ 1 - \theta & y = 0. \end{cases}$$

- Suppose that the prior for θ is given by Beta distribution with known parameters α and β.
- That is $\mathbb{P}(\theta) \sim \text{Beta}(\alpha, \beta)$.
- Given that we recorded m successes (m out of n random variables came out to be 1), derive the posterior distribution.

Conjugate Bernoulli analysis example (2)

The following holds.

1. $\mathbb{P}(\theta) = B(\alpha, \beta)^{-1} \, \theta^{\alpha-1} \, (1-\theta)^{\beta-1}$.
2. The likelihood is given by

$$L(\theta) = \prod_{i=1}^{n} \mathbb{P}(y_i \mid \theta) = \theta^{m} \, (1-\theta)^{n-m}.$$

3. Therefore, the posterior is

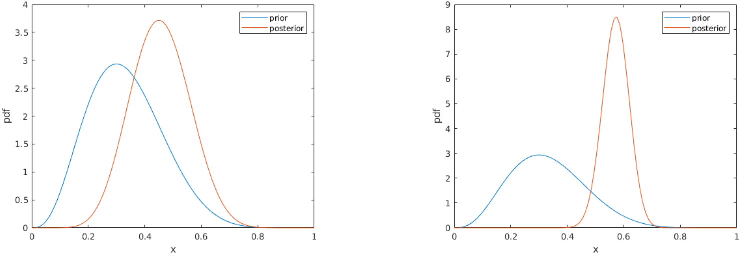
$$\begin{aligned} \mathbb{P}(\theta \mid y) \propto L(\theta) \, \mathbb{P}(\theta) &= \theta^{m} \, (1-\theta)^{n-m} \, \theta^{\alpha-1} \, (1-\theta)^{\beta-1} \\ &= \theta^{(m+\alpha)-1} \, (1-\theta)^{(n-m+\beta)-1}. \end{aligned}$$

It is not very hard to recognize that

$$\mathbb{P}(\theta \mid y) \sim \text{Beta}(\alpha + m, n - m + \beta).$$

Conjugate Bernoulli analysis example (3)

Figure 1 illustrates the prior and the posterior of the Bernoulli example for different sizes of the data. Note that as the sample size of y increases, the posterior peak concentrates around the value of 0.6.



(a) $n = 10, m = 6$

(b) $n = 100, m = 60$

Figure 1: Conjugate Bernoulli analysis. Figures (a) and (b) show the prior and the posterior densities of the Bernoulli experiment with $m, n = (10, 6)$ and $m, n = (100, 60)$, respectively. In both cases, we take $\alpha = 4$ and $\beta = 8$ as a parameters of the prior Beta distribution.

Inference (1)

In general, we would like to calculate various quantities of interest of the form:

$$\mathbb{E}[g(\boldsymbol{\theta}) \mid \boldsymbol{y}] = \int g(\boldsymbol{\theta})\, \mathbb{P}(\boldsymbol{\theta} \mid \boldsymbol{y})\, d\boldsymbol{\theta}.$$

As soon as $\mathbb{P}(\boldsymbol{\theta} \mid \boldsymbol{y})$ is available - we can do anything!!!

- We would like to obtain a point estimate for $\theta_i \in \boldsymbol{\theta}$. In this case, set $g(\boldsymbol{\theta}) = \theta_i$ and calculate:

$$\mathbb{E}[\theta_i \mid \boldsymbol{y}] = \int \theta_i\, \mathbb{P}(\boldsymbol{\theta} \mid \boldsymbol{y})\, d\boldsymbol{\theta}.$$

Inference (2)

- In order to express the uncertainty about $\theta_i \in \boldsymbol{\theta}$, we might want to report the posterior standard deviations of the parameters. Namely, report the standard deviation $\sqrt{\mathrm{Var}(\theta_i \mid \boldsymbol{y})}$. The variance can be calculated via

$$\mathbb{E}[\theta_i^2 \mid \boldsymbol{y}] - (\mathbb{E}[\theta_i \mid \boldsymbol{y}])^2 = \int \theta_i^2 \, \mathbb{P}(\boldsymbol{\theta} \mid \boldsymbol{y}) \, d\boldsymbol{\theta} - \left(\int \theta_i \, \mathbb{P}(\boldsymbol{\theta} \mid \boldsymbol{y}) \, d\boldsymbol{\theta} \right)^2 .$$

- One can show that the prediction of the future value y^* given the past data $\boldsymbol{y}$ can be written in a convenient form via the posterior, namely

$$\mathbb{P}(y^* \mid \boldsymbol{y}) = \int \mathbb{P}(y^* \mid \boldsymbol{y}, \boldsymbol{\theta}) \, \mathbb{P}(\boldsymbol{\theta} \mid \boldsymbol{y}) \, d\boldsymbol{\theta}.$$

- Finally, the Bayesian approach can be used for model selection and hypothesis testing.

Conjugate priors

If the posterior distributions $p(\theta|y)$ are in the same probability distribution family as the prior probability distribution $p(\theta)$, the prior and posterior are then called conjugate distributions, and the prior is called a conjugate prior for the likelihood function $p(y|\theta)$.

Example: Consider n iid Exponential random variables Y_i, $(i = 1, \dots, n)$, each with p.d.f.

$$\mathbb{P}(y \mid \theta) = \frac{1}{\theta} e^{-\frac{1}{\theta} y}.$$

Suppose that the prior for θ is the inverse gamma distribution $\mathrm{IG}(\alpha_0, \beta_0)$. Namely

$$p(\theta) = \frac{\beta_0^{\alpha_0}}{\Gamma(\alpha_0)} \theta^{-(\alpha_0+1)} e^{-\frac{\beta_0}{\theta}}.$$

Derive the posterior distribution of θ.

Conjugate priors - Exponential random variables

1. The prior is
$$\mathbb{P}(\theta) \propto \theta^{-(\alpha_0+1)} \mathrm{e}^{-\frac{\beta_0}{\theta}}.$$
2. The likelihood is
$$\mathbb{P}(\boldsymbol{y} \mid \theta) = \theta^{-n} \mathrm{e}^{-\frac{n\overline{y}}{\theta}}$$
3. The posterior is therefore
$$\mathbb{P}(\theta \mid \boldsymbol{y}) \propto \theta^{-(\alpha_0+1)} \mathrm{e}^{-\frac{\beta_0}{\theta}} \theta^{-n} \mathrm{e}^{\frac{n\overline{y}}{\theta}} \propto \theta^{-(n+\alpha_0+1)} \mathrm{e}^{-\frac{\beta_0+n\overline{y}}{\theta}}$$
That is
$$(\theta \mid \boldsymbol{y}) \sim \mathsf{IG}(n + \alpha_0, \beta_0 + n\overline{y}).$$

Conjugate priors - Normal (with **known** variance)

- Conjugate Normal analysis (with **known** variance): Given the parameter σ^2, consider n iid Normal random variables $Y_i, \ (i = 1, \dots, n)$, each with p.d.f.

$$\mathbb{P}(y \mid \mu, \sigma^2) = (2\pi\sigma^2)^{-\frac{1}{2}} \exp\left\{ -\frac{1}{2\sigma^2}(y_i - \mu)^2 \right\}.$$

  Suppose that the prior for μ is also a normal distribution with parameters μ_0 and σ_0^2. Our objective is to derive the posterior distribution of μ.

- It is possible to show (practical), that if the prior is Normal with parameters μ_0 and σ_0, the posterior satisfies:

$$(\mu \mid \boldsymbol{y}) \sim \mathrm{N}\left(\left(\frac{n\overline{y}}{\sigma^2} + \frac{\mu_0}{\sigma_0^2} \right) \left(\frac{n}{\sigma^2} + \frac{1}{\sigma_0^2} \right)^{-1}, \left(\frac{n}{\sigma^2} + \frac{1}{\sigma_0^2} \right)^{-1} \right).$$

Conjugate priors - Normal Model with Unknown Mean and Variance (1)

- Consider n iid Normal random variables Y_i, $(i = 1, \ldots, n)$. The model is given by:

  $$(y_i \mid \mu, \sigma^2) \sim \mathsf{N}(\mu, \sigma^2), \quad i = 1, \ldots, n.$$

  Here. both μ and σ^2 are unknown.
- We assume the following priors:

  $$(\mu \mid \mu_0, \sigma_0^2) \sim \mathsf{N}(\mu_0, \sigma_0^2), \quad (\sigma^2 \mid \alpha_0, \beta_0) \sim \mathsf{IG}(\alpha_0, \beta_0).$$

Conjugate priors - Normal Model with Unknown Mean and Variance (2)

- The likelihood is:

$$L(\mu) = \mathbb{P}(\boldsymbol{y} \mid \mu, \sigma^2) = \prod_{i=1}^{n} \mathbb{P}(y_i \mid \mu, \sigma^2) = \prod_{i=1}^{n} (2\pi\sigma^2)^{-\frac{1}{2}} \exp\left\{-\frac{1}{2\sigma^2}(y_i - \mu)^2\right\}$$

$$\propto (\sigma^2)^{-\frac{1}{2}n} \exp\left\{\sum_{i=1}^{n} -\frac{1}{2\sigma^2}(y_i - \mu)^2\right\}$$

- Finally, the posterior is;

$$\mathbb{P}(\mu, \sigma^2 \mid \boldsymbol{y}) \propto \mathbb{P}(\mu, \sigma^2, \boldsymbol{y}) \propto \mathbb{P}(\mu)\, \mathbb{P}(\sigma^2)\, \mathbb{P}(\boldsymbol{y} \mid \mu, \sigma^2) \tag{5}$$

$$\propto e^{-\frac{1}{2\sigma_0^2}(\mu-\mu_0)^2} (\sigma^2)^{-(\alpha_0+1)} e^{-\frac{\beta_0}{\sigma^2}} (\sigma^2)^{-\frac{1}{2}n} e^{-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i-\mu)^2}.$$

A very "nice" distribution... Can you work with this one?

Sampling from the posterior

- In some cases, we managed to obtain an analytical expression for the posterior distribution.
- In general, however, the latter is rarely available.
- Consequently, we would like to sample from the posterior distribution. If such sampling is available, we can obtain estimates for various quantities of interest.
- This can be accomplished via Monte Carlo methods. In general, we want to determine the expectation

$$\ell = \mathbb{E}[g(X)] = \int g(x)p(x)\mathrm{d}x.$$

The unbiased estimator of ℓ — the sample mean:

$$\hat{\ell} = \frac{1}{N}\sum_{i=1}^{N} g(X_i),$$

where $X_1, \ldots, X_N \overset{\text{iid}}{\sim} p(x)$, and $p(x)$ is the posterior distribution.

Sampling accuracy — a reminder

- Recall that by the central limit theorem, $\hat{\ell}$ has an approximately normal $\mathsf{N}(\ell, \sigma^2/N)$ distribution.
- Despite that the σ^2 is generally unavailable, it can be estimated from Monte Carlo simulation via a calculation of a sample variance:

$$S^2 = \frac{1}{N-1} \sum_{i=1}^{N} \left(g(X_i) - \hat{\ell} \right)^2 .$$

As $N \to \infty$, $S^2 \to \sigma^2$.

- Consequently, for large N, $\hat{\ell}$ has an approximately $\mathsf{N}(\ell, S^2/N)$ distribution. Thus,

$$\mathbb{P}\left(\hat{\ell} - z_{1-\alpha/2} \frac{S}{\sqrt{N}} \leq \ell \leq \hat{\ell} + z_{1-\alpha/2} \frac{S}{\sqrt{N}} \right) \approx 1 - \alpha,$$

where z_γ denotes the γ quantile of the standard normal distribution. (For example, $z_{0.975} \approx 1.96$.) In particular, we constructed an approximate $(1-\alpha)100\%$ interval for ℓ, and it is equal to:

$$\left(\hat{\ell} \pm z_{1-\alpha/2} \frac{S}{\sqrt{N}} \right)$$

Random number generation

In general, a random number generator has the following structure.

Algorithm 1: Pseudo-random number generator

```
input : An initial number $X_0 \in \mathcal{S}$ called the seed, $f : \mathcal{S} \to \mathcal{S}$,
        $g : \mathcal{S} \to (0,1)$.
output: A stream $U_1, U_2, \ldots,$ of pseudo-random numbers $\sim \mathsf{U}(0,1)$.
for $t = 1$ to $\cdots$ do
    $X_t \leftarrow f(X_{t-1})$.
    $U_t \leftarrow g(X_t)$.
end
```

Random number generation — important questions

First of all, we must answer the important question of our expectations from a good random number generator.

1. It should be robust and reliable.
   1. It should pass *statistical tests*, namely, it should output a stream of uniform random numbers that is indistinguishable from a genuine uniform iid sequence.
   2. It should have a theoretical support, so that rigorous properties of the generator could be analyzed.
2. It should be *fast*.
3. It should be *reproducible*. That is, one should be able to recover the stream without storing it in the memory. This property is important for testing.
4. The *period* of the generator is the smallest number of steps taken before entering a previously visited state. A good generator should have a large period.
5. It should be cheap in the sense that it will not require an additional or an expansive equipment.
6. It should not produce 0 or 1, to avoid the division by 0.
7. It should be application dependent. For example, in Monte Carlo simulation, the distributional properties are very important. In the other hand, in cryptography, it is crucial that the generated sequence will be hard to predict.

So why the uniform number is sufficient?

We are interested in a generation of a random variable X with commutative distribution function (cdf) $F(x)$. Then, the inverse-transform method is

$$X = F^{-1}(U), \quad U \sim \mathrm{U}(0,1).$$

The procedure is given in the Algorithm below.

Algorithm 2: Inverse-Transform Method

```
input  : Commutative distribution function $F$
output: $X \sim F$
Generate $U \sim \mathrm{U}(0,1)$.
$X \leftarrow F^{-1}(U)$.
return $X$.
```

This is called the inverse-transform method.

The inverse-transform method

We next show why this is true. By definition:

$$F^{-1}(y) = \inf\{x : F(x) \geq y\}, \quad 0 \leq y \leq 1.$$

To show that $X \sim F$, note that

$$\mathbb{P}(X \leq x) = \mathbb{P}\left(F^{-1}(U) \leq x\right) = \mathbb{P}\left(U \leq F(x)\right) = F(x),$$

where the second and the third equalities follow from the fact that F is invertible and that $\mathbb{P}(U \leq \alpha) = \alpha$ for any $0 \leq \alpha \leq 1$, respectively.

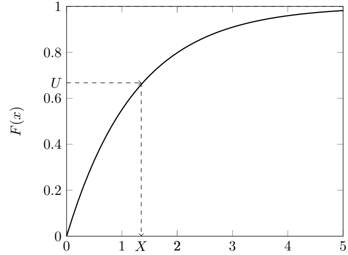


Figure 2: Inverse-transform Method

Example

Exponential distribution

Consider the Exponential distribution with probability density function (pdf)

$$f(x) = \lambda\,\mathrm{e}^{-\lambda\,x}.$$

The cdf is given by:

$$F(x) = \mathbb{P}(X \leq x) = \int_0^x \lambda\,\mathrm{e}^{-\lambda\,u}\mathrm{d}u = 1 - \mathrm{e}^{-\lambda\,x}.$$

We complete the solution by writing:

$$X = F^{-1}(U) \;\Rightarrow\; F(X) = U \;\Rightarrow\; 1 - \mathrm{e}^{-\lambda\,X} \;\Rightarrow\; X = -\frac{1}{\lambda}\ln U.$$

Inverse-Transform Method in Discrete Case

The inverse-transform method can also be used for generating from a discrete distribution. The setting is as follows.

- Suppose that X is a discrete random variable, taking values in the set $\{x_1, \dots, x_n\}$.
- It holds that $\mathbb{P}(X = x_i) = p_i$ and $\sum_{i=1}^{n} p_i = 1$ for $i = 1, \dots, n$.

Suppose without loss of generality that $x_1 < x_2 < \cdots < x_n$. Then, the cdf is given by

$$F(x) = \sum_{i=1}^{x_i < x} p_i.$$

Inverse-Transform Method in Discrete Case

Algorithm 3: Inverse-Transform Method for a Discrete Distribution

```
input  : Discrete commutative distribution function $F$
output: $X \sim F$
Generate $U \sim \mathrm{U}(0,1)$.
Let $j$ be the smallest integer such that $U \le F(x_j)$.
$X \leftarrow x_j$.
return $X$.
```

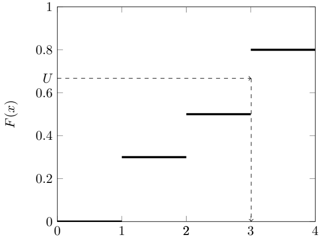


Figure 3: Discrete Inverse-transform Method

Example — Bernoulli Distribution

The pdf is given by

$$f(x;p) = p^x(1-p)^{1-x} \quad x \in \{0,1\}.$$

Here, $0 \le p \le 1$ stands for the success parameter. The distribution is written as $\text{Ber}(p)$.

Algorithm 4: Bernoulli random variable generation

```
input  : The success parameter $p$.
output: $X \sim \text{Ber}(p)$.
Generate $U \sim \text{Uni}(0,1)$.
if $U \le p$ then
    $X \leftarrow 1$.
end
else
    $X \leftarrow 0$.
end
return $X$.
```

Example — Binomial Distribution

Can you propose a method to generate $X \sim \mathrm{Bin}(n, p)$?

Poisson Process

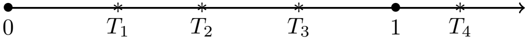


Figure 4: Poisson arrivals. That is, the time between 0 and T_1, T_1 and T_2, T_2 and T_3, T_3 and T_4 is exponentially distributed with parameter λ.

Algorithm 5: The Poisson Process Generation

```
input  : The parameter $\lambda$.
output: A random vector $(T_1, \dots, T_n)$ of arrivals.
Set $T_0 \leftarrow 0$.
for $i = 1$ to $n$ do
    Generate $X_i \sim \text{Exp}(\lambda)$.
    Set $T_i \leftarrow T_{i-1} + X_i$.
end
return $(T_1, \dots, T_n)$.
```

Random Walk on Integers

Consider a walk that starts at $x = 0$, and,

$$\mathbb{P}(i, i+1) = p, \quad \mathbb{P}(i, i-1) = 1 - p = q.$$

The generation idea is simple. We just draw from $\text{Ber}(p)$ and make the corresponding step.

Markov Chains

Let $\{X_t\ ,\ t \geq 0\}$ be a discrete time Markov Chain. Recall that the transition probabilities are given by a matrix P. Given the transition probabilities, the Markov Chain generation is straight-forward.

Algorithm 6: Markov Chain Generation

```
input  : An initial state $X_0$.
output: A random vector $(X_1, \dots, X_n)$ of MC states.
for $i = 1$ to $n$ do
    Generate $X_{t+1}$ from the discrete distribution corresponding to the
      $X_t$-th row of $P$.
end
return $\boldsymbol{X} = (X_1, \dots, X_n)$.
```

Random walks on graphs (1)

1. A graph is described by a set of vertices V, and a set of edges E.
2. An adjacency matrix $A(u, v)$, which is 1 if u and v are "neighbors" and 0 otherwise.
3. The degree of a vertex u is equal to the number of neighbors it has:

$$d(u) = \sum_{v} A(u, v).$$

Motivation: Google page rank; more important nodes have a higher degree.

Random walks on graphs (2)

- Let X_n be a Markov chain on the graph vertices with transition probabilities defined as follows.
- Suppose that we are located in $u \in V$. Choose a random neighbor and go there. In this case:

$$P(u,v) = \frac{A(u,v)}{d(u)}.$$

- It can be shown analytically that

$$\pi(v) = \frac{d(v)}{2|E|}, \quad \forall v \in V.$$

Random walks on graphs (3)

Consider an example (draw a graph associated with A):

$$A = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix} \Rightarrow P = \begin{pmatrix} 0 & 1/2 & 1/2 & 0 & 0 \\ 1/3 & 0 & 1/3 & 1/3 & 0 \\ 1/4 & 1/4 & 0 & 1/4 & 1/4 \\ 0 & 1/2 & 1/2 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}.$$

- In this example, we have 6 edges and the vertex degrees are $(2, 3, 4, 2, 1)$. That is

$$\pi = \left(\frac{2}{12}, \frac{3}{12}, \frac{4}{12}, \frac{2}{12}, \frac{1}{12}\right) \approx$$

$$\approx (0.1667,\ 0.2500,\ 0.3333,\ 0.1667,\ 0.0833).$$

- Now, we can compare this with the MC program's output.

Markov Chain Monte Carlo (1)

- Recall the posterior distribution from the normal model with the unknown mean and the unknown variance:

$$\mathbb{P}(\mu, \sigma^2 \mid \boldsymbol{y}) \propto \mathbb{P}(\mu, \sigma^2, \boldsymbol{y}) \propto \mathbb{P}(\mu)\, \mathbb{P}(\sigma^2)\, \mathbb{P}(\boldsymbol{y} \mid \mu, \sigma^2)$$
$$\propto \mathrm{e}^{-\frac{1}{2\sigma_0^2}(\mu-\mu_0)^2} \left(\sigma^2\right)^{-(\alpha_0+1)} \mathrm{e}^{-\frac{\beta_0}{\sigma^2}} \left(\sigma^2\right)^{-\frac{1}{2}n} \mathrm{e}^{-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i-\mu)^2}.$$

- If we could obtain samples from this distribution, we could (easily) make inference about the model parameters.
- A general method for such sampling from complex distribution is *Markov Chain Monte Carlo* (MCMC).

Markov Chain Monte Carlo (2)

- *Markov chain Monte Carlo* (MCMC) is a Monte Carlo sampling technique for (approximately) generating samples from an arbitrary distribution — often referred to as the *target* distribution.
- The basic idea is to run a Markov chain long enough such that its limiting distribution is close to the target distribution.
- Often such a Markov chain is constructed to be reversible, so that the detailed balance equations can be used.
- Depending on the starting position of the Markov chain, the initial random variables in the Markov chain may have a distribution that is significantly different from the target (limiting) distribution. The random variables that are generated during this *burn-in period* are often discarded.
- The remaining random variables form an *approximate* and *dependent* sample from the target distribution.

Metropolis–Hastings Sampler

- The Metropolis–Hastings sampler simulates a trial state, which is then accepted or rejected according to some random mechanism.
- Specifically, suppose we wish to sample from a target pdf $f(\boldsymbol{x})$, where $\boldsymbol{x}$ takes values in some d-dimensional set.
- The aim is to construct a Markov chain $\{\boldsymbol{X}_t, t = 0, 1, \ldots\}$ in such a way that its limiting pdf is f.
- Suppose the Markov chain is in state $\boldsymbol{x}$ at time t. A transition of the Markov chain from state $\boldsymbol{x}$ is carried out in two phases.
- First a *proposal* state $\boldsymbol{Y}$ is drawn from a transition density $q(\cdot \mid \boldsymbol{x})$. This state is accepted as the new state, with *acceptance probability*

$$\alpha(\boldsymbol{x}, \boldsymbol{y}) = \min\left\{\frac{f(\boldsymbol{y})\, q(\boldsymbol{x} \mid \boldsymbol{y})}{f(\boldsymbol{x})\, q(\boldsymbol{y} \mid \boldsymbol{x})}, 1\right\}, \tag{6}$$

  or rejected otherwise.
- In the latter case the chain remains in state $\boldsymbol{x}$. The algorithm just described can be summarized as follows.

Metropolis–Hastings Sampler Algorithm

Algorithm 7: Metropolis–Hastings Sampler

```
input: Initial state $\boldsymbol{X}_0$, sample size $N$, target pdf $f(\boldsymbol{x})$, proposal
       function $q(\boldsymbol{x}, \boldsymbol{y})$.
output: $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_N$ (dependent), approximately distributed according
        to $f(\boldsymbol{x})$.
for $t = 0$ to $N - 1$ do
    Draw $\boldsymbol{Y} \sim q(\boldsymbol{y} \mid \boldsymbol{X}_t)$                // draw a proposal
    $\alpha \leftarrow \alpha(\boldsymbol{X}_t, \boldsymbol{Y})$       // acceptance probability as in (6)
    Draw $U \sim \mathsf{U}(0,1)$
    if $U \leq \alpha$ then $\boldsymbol{X}_{t+1} \leftarrow \boldsymbol{Y}$
    else $\boldsymbol{X}_{t+1} \leftarrow \boldsymbol{X}_t$
end
return $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_N$
```

The Metropolis–Hastings Sampling

The fact that the limiting distribution of the Metropolis–Hastings Markov chain is equal to the target distribution (under general conditions) is a consequence of the following result.

Theorem (Local Balance for the Metropolis–Hastings Sampler)

The transition density of the Metropolis–Hastings Markov chain satisfies the detailed balance equations.

- The efficiency of the algorithm depends of course on the choice of the proposal transition density $q(\boldsymbol{y} \mid \boldsymbol{x})$.
- Ideally, we would like $q(\boldsymbol{y} \mid \boldsymbol{x})$ to be “close” to the target $f(\boldsymbol{y})$, irrespective of $\boldsymbol{x}$. We discuss two common approaches.

Proposals

1. Choose the proposal transition density $q(\boldsymbol{y} \mid \boldsymbol{x})$ independent of $\boldsymbol{x}$; that is, $q(\boldsymbol{y} \mid \boldsymbol{x}) = g(\boldsymbol{y})$ for some pdf $g(\boldsymbol{y})$. An MCMC sampler of this type is called an *independence sampler*. The acceptance probability is thus

$$\alpha(\boldsymbol{x}, \boldsymbol{y}) = \min\left\{\frac{f(\boldsymbol{y})\, g(\boldsymbol{x})}{f(\boldsymbol{x})\, g(\boldsymbol{y})},\ 1\right\}.$$

2. If the proposal transition density is symmetric (that is, $q(\boldsymbol{y} \mid \boldsymbol{x}) = q(\boldsymbol{x} \mid \boldsymbol{y})$), then the acceptance probability has the simple form

$$\alpha(\boldsymbol{x}, \boldsymbol{y}) = \min\left\{\frac{f(\boldsymbol{y})}{f(\boldsymbol{x})},\ 1\right\}, \tag{7}$$

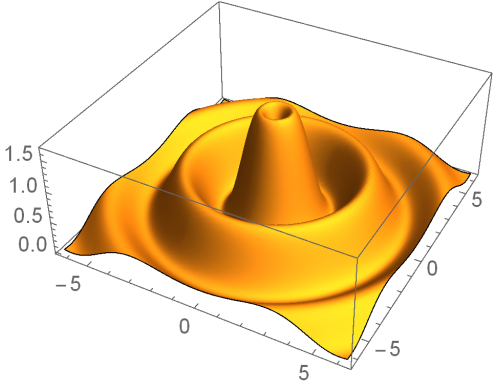
and the MCMC algorithm is called a *random walk sampler*. A typical example is when, for a given current state $\boldsymbol{x}$, the proposal state $\boldsymbol{Y}$ is of the form $\boldsymbol{Y} = \boldsymbol{x} + \boldsymbol{Z}$, where $\boldsymbol{Z}$ is generated from some spherically symmetric distribution, such as $\mathrm{N}(0, \boldsymbol{I})$.

Random Walk Sampler Example (1)

Consider the two-dimensional pdf

$$f(x_1, x_2) = c\,\mathrm{e}^{-\frac{1}{4}\sqrt{x_1^2+x_2^2}}\left(\sin\left(2\sqrt{x_1^2+x_2^2}\right)+1\right), \quad -2\pi < x_1 < 2\pi,\ -2\pi < x_2 < 2\pi,$$

where c is an unknown normalization constant.

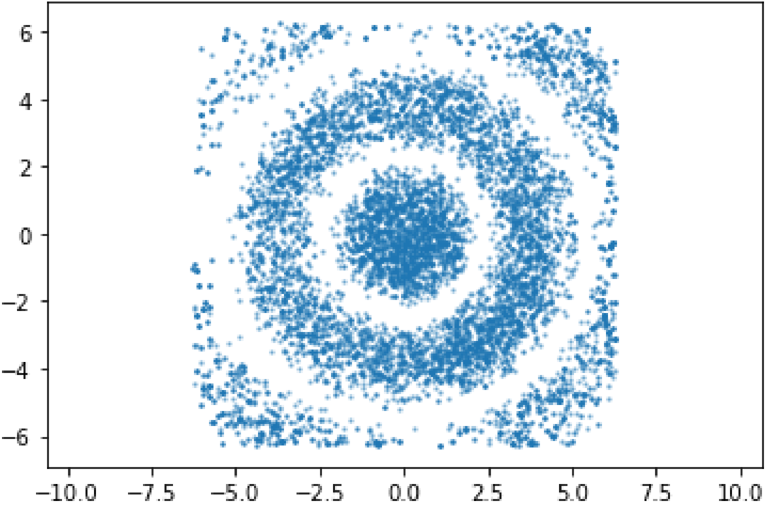


Random Walk Sampler Example (2)

- The following Python program implements a random walk sampler to (approximately) draw $N = 10^4$ dependent samples from the pdf f.
- At each step, given a current state $\boldsymbol{x}$, a proposal $\boldsymbol{Y}$ is drawn from the $\mathrm{N}(\boldsymbol{x}, \boldsymbol{I})$ distribution.
- That is, $\boldsymbol{Y} = \boldsymbol{x} + \boldsymbol{Z}$, with $\boldsymbol{Z}$ bivariate standard normal.
- . The starting point for the Markov chain is chosen as $(0, 0)$.
- Note that the normalization constant c is never required to be specified in the program.

```
# MCMC1.py
import numpy as np
import matplotlib.pyplot as plt
from numpy import pi, exp, sqrt, sin
from numpy.random import rand, randn
N = 10000
a = lambda x: -2*pi < x
b = lambda x: x < 2*pi
f = lambda x1, x2: exp(-sqrt(x1**2+x2**2)/4)*(
        sin(2*sqrt(x1**2+x2**2))+1)*a(x1)*b(x1)*a(x2)*b(x2)
xx = np.zeros((N,2))
x = np.zeros((1,2))
for i in range(1,N):
    y = x + randn(1,2)
    alpha = np.amin((f(y[0][0],y[0][1])/f(x[0][0],x[0][1]),1))
    r = rand() < alpha
    x = r*y + (1-r)*x
    xx[i,:] = x
plt.scatter(xx[:,0], xx[:,1], alpha =0.4,s =2)
plt.axis('equal')
plt.show()
```

Random Walk Sampler Example (3)



The Gibbs Sampler

- The *Gibbs sampler* uses a somewhat different methodology from the Metropolis–Hastings algorithm and is particularly useful for generating n-dimensional random vectors.
- The key idea of the Gibbs sampler is to update the components of the random vector one at a time, by sampling them from *conditional* pdfs.
- Thus, Gibbs sampling can be advantageous if it is easier to sample from the conditional distributions than from the joint distribution.
- Specifically, suppose that we wish to sample a random vector $\boldsymbol{X} = [X_1, \ldots, X_n]^\top$ according to a target pdf $f(\boldsymbol{x})$. Let $f(x_i \mid x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$ represent the conditional pdf of the i-th component, X_i, given the other components $x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n$.
- The Gibbs sampling algorithm is as follows.

The Gibbs Sampling Algorithm

Algorithm 8: Gibbs Sampler

```
input: Initial point $\boldsymbol{X}_0$, sample size $N$, and target pdf $f$.
output: $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_N$ approximately distributed according to $f$.
for $t = 0$ to $N - 1$ do
    Draw $Y_1$ from the conditional pdf $f(y_1 \mid X_{t,2}, \ldots, X_{t,n})$.
    for $i = 2$ to $n$ do
        Draw $Y_i$ from the conditional pdf
          $f(y_i \mid Y_1, \ldots, Y_{i-1}, X_{t,i+1}, \ldots, X_{t,n})$.
    end
    $\boldsymbol{X}_{t+1} \leftarrow \boldsymbol{Y}$
end
return $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_N$
```

The Gibbs Sampling (1)

- There exist many variants of the Gibbs sampler, depending on the steps required to update $\boldsymbol{X}_t$ to $\boldsymbol{X}_{t+1}$ — called the *cycle* of the Gibbs algorithm.
- In the algorithm above, the cycle consists of Steps 2–5, in which the components are updated in a fixed order $1 \to 2 \to \cdots \to n$. For this reason Algorithm 8 is also called the *systematic Gibbs sampler*.
- In the *random-order Gibbs sampler*, the order in which the components are updated in each cycle is a random permutation of $\{1, \ldots, n\}$.
- Other modifications are to update the components in blocks (i.e., several at the same time), or to update only a random selection of components.
- The variant where in each cycle only a single random component is updated is called the *random Gibbs sampler*.

The Gibbs Sampling (2)

- In the *reversible Gibbs sampler* a cycle consists of the coordinate-wise updating

$$1 \to 2 \to \cdots \to n-1 \to n \to n-1 \to \cdots \to 2 \to 1.$$

- In all cases, except for the systematic Gibbs sampler, the resulting Markov chain $\{\boldsymbol{X}_t, t = 1, 2, \ldots\}$ is *reversible* and hence its limiting distribution is precisely $f(\boldsymbol{x})$.

Normal Model with Unknown Mean and Variance (1)

Consider n iid Normal random variables Y_i, $(i = 1, \ldots, n)$. The model is given by:

$$(y_i \mid \mu, \sigma^2) \sim \mathsf{N}(\mu, \sigma^2), \quad i = 1, \ldots, n.$$

Here. both μ and σ^2 are unknown.

1. We assume the following priors:

$$(\mu \mid \mu_0, \sigma_0^2) \sim \mathsf{N}(\mu_0, \sigma_0^2), \quad (\sigma^2 \mid \alpha_0, \beta_0) \sim \mathsf{IG}(\alpha_0, \beta_0).$$

2. The likelihood is:

$$\begin{aligned} L(\mu) = \mathbb{P}(\boldsymbol{y} \mid \mu, \sigma^2) &= \prod_{i=1}^{n} \mathbb{P}(y_i \mid \mu, \sigma^2) = \prod_{i=1}^{n} (2\pi\sigma^2)^{-\frac{1}{2}} \exp\left\{-\frac{1}{2\sigma^2}(y_i - \mu)^2\right\} \\ &\propto (\sigma^2)^{-\frac{1}{2}n} \exp\left\{\sum_{i=1}^{n} -\frac{1}{2\sigma^2}(y_i - \mu)^2\right\} \end{aligned}$$

3. Finally, the posterior is;

$$\begin{aligned} \mathbb{P}(\mu, \sigma^2 \mid \boldsymbol{y}) \propto \mathbb{P}(\mu, \sigma^2, \boldsymbol{y}) &\propto \mathbb{P}(\mu)\, \mathbb{P}(\sigma^2)\, \mathbb{P}(\boldsymbol{y} \mid \mu, \sigma^2) \qquad (8) \\ &\propto \mathrm{e}^{-\frac{1}{2\sigma_0^2}(\mu-\mu_0)^2} (\sigma^2)^{-(\alpha_0+1)} \mathrm{e}^{-\frac{\beta_0}{\sigma^2}} (\sigma^2)^{-\frac{1}{2}n} \mathrm{e}^{-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i-\mu)^2}. \end{aligned}$$

Normal Model with Unknown Mean and Variance (2)

In order to implement the Gibbs sampler for $\mathbb{P}(\mu, \sigma^2 \mid \boldsymbol{y})$, we need to derive the following conditional distributions: $\mathbb{P}(\mu \mid \sigma^2, \boldsymbol{y})$, and $\mathbb{P}(\sigma^2 \mid, \mu, \boldsymbol{y})$. With these distributions, the Gibbs sampler is as follows. Pick initial values: $\mu^{(0)}$ and $(\sigma^2)^{(0)}$

1. Draw $\mu^{(r)} \sim \mathbb{P}(\mu \mid (\sigma^2)^{(r-1)}, \boldsymbol{y})$.
2. Draw $(\sigma^2)^{(r)} \sim \mathbb{P}(\sigma^2 \mid \mu^{(r)}, \boldsymbol{y})$.

Apparently, it is not hard to derive these conditional distributions. Noting that:

$$\begin{aligned}\mathbb{P}(\mu \mid \sigma^2, \boldsymbol{y}) &\propto \mathrm{e}^{-\frac{1}{2\sigma_0^2}(\mu-\mu_0)^2} (\sigma^2)^{-(\alpha_0+1)} \mathrm{e}^{-\frac{\beta_0}{\sigma^2}} (\sigma^2)^{-\frac{1}{2}n} \mathrm{e}^{-\frac{1}{2\sigma^2}\sum_{i=1}^n (y_i-\mu)^2} \\ &\propto \mathrm{e}^{-\frac{1}{2\sigma_0^2}(\mu-\mu_0)^2} \mathrm{e}^{-\frac{1}{2\sigma^2}\sum_{i=1}^n (y_i-\mu)^2},\end{aligned}$$

that is, $(\mu \mid \sigma^2, \boldsymbol{y}) \sim \mathsf{N}\left(\left(\frac{n\overline{y}}{\sigma^2} + \frac{\mu_0}{\sigma_0^2}\right)\left(\frac{n}{\sigma^2} + \frac{1}{\sigma_0^2}\right)^{-1}, \left(\frac{n}{\sigma^2} + \frac{1}{\sigma_0^2}\right)^{-1}\right)$.

Normal Model with Unknown Mean and Variance (3)

In addition, it holds that:

$$\begin{aligned}
\mathbb{P}(\sigma^2 \mid \mu, \boldsymbol{y}) &\propto \mathrm{e}^{-\frac{1}{2\sigma_0^2}(\mu-\mu_0)^2} \, (\sigma^2)^{-(\alpha_0+1)} \mathrm{e}^{-\frac{\beta_0}{\sigma^2}} (\sigma^2)^{-\frac{1}{2}n} \mathrm{e}^{-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i-\mu)^2} \\
&\propto (\sigma^2)^{-\left(\frac{1}{2}n+\alpha_0+1\right)} \mathrm{e}^{-\frac{\beta_0}{\sigma^2}} \mathrm{e}^{-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i-\mu)^2} \\
&\propto (\sigma^2)^{-\left(\frac{1}{2}n+\alpha_0+1\right)} \mathrm{e}^{\frac{\beta_0+\frac{1}{2}\sum_{i=1}^{n}(y_i-\mu)^2}{\sigma^2}}.
\end{aligned}$$

Consequently, $(\sigma^2 \mid \mu, \boldsymbol{y}) \sim \mathrm{IG}\left(\frac{1}{2}n + \alpha_0,\ \beta_0 + \frac{1}{2}\sum_{i=1}^{n}(y_i - \mu)^2\right)$.

Normal Model with Unknown Mean and Variance (4)

- The following code generates a sample of 100 points from $\mathsf{N}(3, \sqrt{0.15}^2)$, set the corresponding priors and runs the Gibbs sampler.
- After a burnin period of 1000 samples and simulation of 10000 samples, the estimated values for the posterior mean of μ and σ^2 are 3.0759 and 0.1535, respectively.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% NormalModelMCMC.m                                                       %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
rng(12345);
% generae dataset
n = 100;
mu = 3;
sig2 = 0.15;
y = randn(n,1)*sig2^0.5 + mu;

% set prior
mu0 = -1; sig20 = 100;      % normal
alpha0 = 3; beta0 = 0.5;    % inverse gamma

% simulation parameters
N = 10000;
burnin = 1000;
% array for theta recording
ell = zeros(N,2);
% initial values
mu = 0; sigma2 = 10;
```

```
% Main MCMC loop
for i = 1:N + burnin
    % sample mu
    sigma_post = (1/sig20 + n/sigma2)^-1;
    mu_post = (n*mean(y)/sigma2 + mu0/sig20)*sigma_post;
    mu = mvnormalrnd(mu_post,sigma_post);

    % sample sigmna2
    alpha = 0.5*n + alpha0;
    betta =  (beta0 + 0.5*sum((y-mu).^2)) ;
    sigma2 = igrnd(alpha,betta);

    if(i>burnin)
        ell(i-burnin,:) = [mu,sigma2];
    end
end
fprintf("posterior (mu,sigma^2) is (%d, %d) \n",mean(ell));

>> NormalModelMCMC
posterior (mu,sigma^2) is (3.075905e+00, 1.535462e-01)
```

We used the following helper methods.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% mvnormalrnd.m                                                     %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function X = mvnormalrnd(mu, Sigma)
    C = chol(Sigma,'lower');
    X = mu +C*randn(length(Sigma),1);
end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% igrnd.m                                                           %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function X = igrnd(alpha,beta)
    X = 1/gamrnd(alpha,1/beta);
end
```

Statistical modeling of disasters (1)

- Bayesian inference is so powerful, that we would like to consider an additional example.
- We consider the dataset, which is a time series of recorded coal mining disasters in the UK from 1851 to 1962.

```
[ 4, 5, 4, 0, 1, 4, 3, 4, 0, 6, 3, 3, 4, 0, 2, 6,
  3, 3, 5, 4, 5, 3, 1, 4, 4, 1, 5, 5, 3, 4, 2, 5,
  2, 2, 3, 4, 2, 1, 3, 2, 2, 1, 1, 1, 1, 3, 0, 0,
  1, 0, 1, 1, 0, 0, 3, 1, 0, 3, 2, 2, 0, 1, 1, 1,
  0, 1, 0, 1, 0, 0, 0, 2, 1, 0, 0, 0, 1, 1, 0, 2,
  3, 3, 1, 1, 2, 1, 1, 1, 1, 2, 4, 2, 0, 0, 1, 4,
  0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1]
```

Statistical modeling of disasters (2)

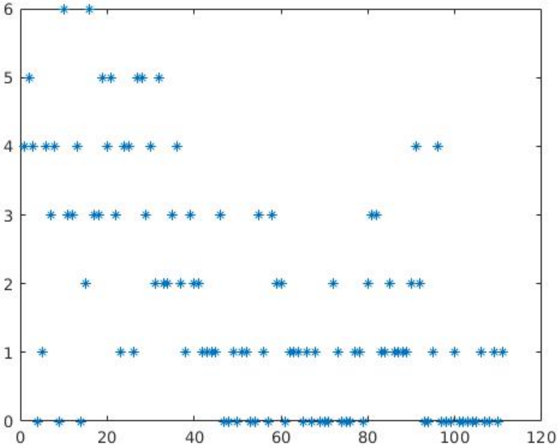


Figure 5: Recorded coal mining disasters in the UK.

Statistical modeling of disasters (3)

- Occurrences of disasters in the time series is thought to be derived from a Poisson process with a large rate parameter in the early part of the time series, and from one with a smaller rate in the later part.
- We are interested in locating the change point in the series, which perhaps is related to changes in mining safety regulations. The formal model is summarized below.

$$
\begin{aligned}
s &\sim \mathrm{DiscreteUniform}[1, 111] \quad \text{(prior)} \\
r_1 &\sim \mathrm{Exp}(1) \quad \text{(prior)} \\
r_2 &\sim \mathrm{Exp}(1) \quad \text{(prior)} \\
(D_t \mid s, r_1, r_2) &\sim \begin{cases} \mathrm{Poisson}(r_1) & t \leq s \\ \mathrm{Poisson}(r_2) & t < s. \end{cases}
\end{aligned}
$$

Statistical modeling of disasters (4)

Our objective is to make inference about s (the regime switch point); here, r_1 and r_2 are Poisson rates (before s and after s, respectively).

First, we note that (for $n = 111$, the number of observations),

$$p(D \mid s, r_1, r_2) = \prod_{t=1}^{s-1} \frac{r_1^{D_t} e^{-r_1}}{D_t!} \prod_{t=s}^{n} \frac{r_2^{D_t} e^{-r_2}}{D_t!} \propto r_1^{\sum_{t=1}^{s-1} D_t} e^{-\sum_{t=1}^{s-1} r_1} r_2^{\sum_{t=s}^{n} D_t} e^{-\sum_{t=s}^{n} r_2}.$$

In addition, it holds that

$$p(s) = \frac{1}{111},\ p(r_1) = e^{-r_1},\ p(r_2) = e^{-r_2}.$$

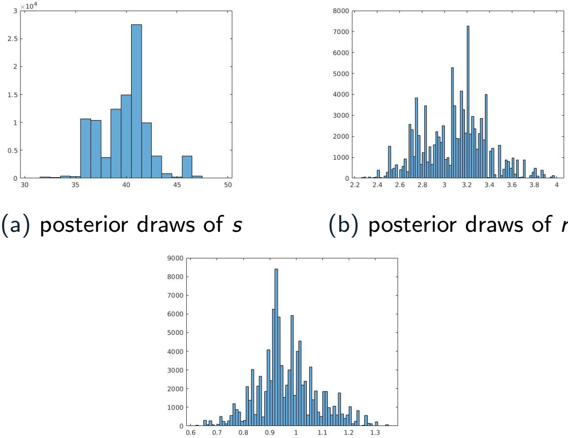
Statistical modeling of disasters (5)

Applying the Bayesian theorem, we have:

$$
\begin{aligned}
p(s, r_1, r_2 \mid D) \propto p(D \mid s, r_1, r_2)\, p(s, r_1, r_2) \propto (D \mid s, r_1, r_2)\, p(s)\, p(r_1)\, p(r_2) \\
\propto r_1^{\sum_{t=1}^{s-1} D_t} \mathrm{e}^{-\sum_{t=1}^{s-1} r_1} r_2^{\sum_{t=s}^{n} D_t} \mathrm{e}^{-\sum_{t=s}^{n} r_2} \frac{1}{111}\, \mathrm{e}^{-r_1}\, \mathrm{e}^{-r_2} \\
\propto r_1^{\sum_{t=1}^{s-1} D_t} \mathrm{e}^{-\left(1+\sum_{t=1}^{s-1} r_1\right)} r_2^{\sum_{t=s}^{n} D_t} \mathrm{e}^{-\left(1+\sum_{t=s}^{n} r_2\right)}.
\end{aligned}
$$

The Figure on the next slide depicts the posterior draws of the variables s, r_1, r_2. The output of the bellow MCMC program is:

```
>> disaster_model
s =  3.991510e+01, 5 percent  CI = (36,46)
r_1 =  3.094196e+00, 5 percent  CI = (2.513411e+00,3.713855e+00)
r_2 =  9.665869e-01, 5 percent  CI = (7.587527e-01,1.207083e+00)
```



(a) posterior draws of s

(b) posterior draws of r_1

(c) posterior draws of r_2

Figure 6: Posterior draws.

```
% disaster_model.m
disastersarray = [4, 5, 4, 0, 1, 4, 3, 4, 0, 6, 3, 3, 4, 0, 2, 6, ...
         3, 3, 5, 4, 5, 3, 1, 4, 4, 1, 5, 5, 3, 4, 2, 5, ...
         2, 2, 3, 4, 2, 1, 3, 2, 2, 1, 1, 1, 1, 3, 0, 0, ...
         1, 0, 1, 1, 0, 0, 3, 1, 0, 3, 2, 2, 0, 1, 1, 1, ...
         0, 1, 0, 1, 0, 0, 0, 2, 1, 0, 0, 0, 1, 1, 0, 2, ...
         3, 3, 1, 1, 2, 1, 1, 1, 1, 2, 4, 2, 0, 0, 1, 4, ...
         0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1];

n = length(disastersarray);

%plot(disastersarray,’*’)
%return
N = 100000;
burnin = 10000;
% init chain
s  = 2;
r1 = 4;
r2 = 4;
```

```
ac_rate = 0;
ell = zeros(N,3);
%MCMC
for i=1:N+burnin
    % sample
    s_ = ceil(rand*n);
    r1_ = rand*4;
    r2_ = rand*2;
    accept = min(1, ...
      exp(target(s_,r1_,r2_,disastersarray)-target(s,r1,r2,disastersarray)));
    if(rand<=accept)
        s = s_;
        r1 = r1_;
        r2 = r2_;
        ac_rate = ac_rate +1;
    end
    if(i>burnin)
       ell(i- burnin,:) = [s,r1,r2];
    end
end
```

```
%fprintf("acceptence rate = %d\n",ac_rate/(N+burnin));
fprintf("s =  %d, 5 percent  CI = (%d,%d) \n",mean(ell(:,1)), ...
                    quantile(ell(:,1),[0.025,0.975]));

fprintf("r_1 =  %d, 5 percent  CI = (%d,%d) \n",mean(ell(:,2)), ...
                    quantile(ell(:,2),[0.025,0.975]));

fprintf("r_2 =  %d, 5 percent  CI = (%d,%d) \n",mean(ell(:,3)), ...
                    quantile(ell(:,3),[0.025,0.975]));


figure(1)
histogram(ell(:,1))
figure(2)
histogram(ell(:,2))
figure(3)
histogram(ell(:,3))
```

```
function logres = target(s,r1,r2,disastersarray)
    [d1,d2] = DataPartition(s,disastersarray);
    s1 = length(d1);
    s2 = length(d2);
    logres = sum(d1)*log(r1) + sum(d2)*log(r2) - (s1+1)*r1 - (s2+1)*r2;
end

function [d1,d2] = DataPartition(s,disastersarray)
    n = length(disastersarray);
    if(s==1)
       d1 = [];
       d2 = disastersarray;
    elseif(s==n)
        d1 = disastersarray;
        d2 = [];
    else
        d1 = disastersarray(1:s);
        d2 = disastersarray(s:length(disastersarray));
    end
end
```

Normal Linear Regression (1)

- We are now ready to consider a simple normal linear regression model. Our setting is follows. Let

$$y_t = \beta_0 x_{t,0} + \beta_1 \times x_{t,1} + \beta_2 \times x_{t,2} + \cdots + \beta_k \times x_{t,k} + \epsilon_t,$$

  where $\{x_{t,j}\}_{j=0}^{k}$ are explanatory variables, $\beta_0, \dots, \beta_k$ are regression coefficients, and $\epsilon_t \sim \mathrm{N}(0, \sigma^2)$, for $t = 1, \dots, n$.

- That is we are given n data point tuples $(\boldsymbol{x}_t, y_t)_{t=1}^{n}$. It is convenient to write the model via a matrix notation. Define $\boldsymbol{y} = (y_1, \dots, y_n)$, $\boldsymbol{\beta} = (\beta_0, \dots, \beta_k)$, and X to be the design matrix, namely

$$X = \begin{pmatrix} 1 & x_{1,1} & x_{12} & \dots & x_{1k} \\ 1 & x_{2,1} & x_{2,2} & \dots & x_{2,k} \\ \vdots & & & & \\ 1 & x_{n,2} & x_{n,3} & \dots & x_{n,k} \end{pmatrix}.$$

Normal Linear Regression (2)

- Then, the regression model in the matrix notation becomes:

$$\boldsymbol{y} = X\boldsymbol{\beta} + \boldsymbol{\epsilon}. \tag{9}$$

  Note that $\epsilon \sim \mathrm{MVN}(0, \sigma^2 \boldsymbol{I}_n)$, where $\boldsymbol{I}_n$ is the $n \times n$ identity matrix.

- In this case, the likelihood function is the joint density of the data given the parameters. Specifically, $L(\boldsymbol{\beta}, \sigma^2) = p(\boldsymbol{y} \mid \boldsymbol{\beta}, \sigma^2)$. Equation (9) is an affine transformation, that is,

$$(\boldsymbol{y} \mid \boldsymbol{\beta}, \sigma^2) \sim \mathrm{N}(X\boldsymbol{\beta}, \sigma^2 \boldsymbol{I}_n).$$

  The likelihood function is therefore:

$$\begin{aligned} p(\boldsymbol{y} \mid \boldsymbol{\beta}, \sigma^2) &= |2\pi\sigma^2 \boldsymbol{I}_n|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(\boldsymbol{y} - X\boldsymbol{\beta})^\top \left(\sigma^2 \boldsymbol{I}_n\right)^{-1} (\boldsymbol{y} - X\boldsymbol{\beta})\right\} \\ &= \left(2\pi\sigma^2\right)^{-\frac{n}{2}} \exp\left\{-\frac{1}{2\sigma^2}(\boldsymbol{y} - X\boldsymbol{\beta})^\top (\boldsymbol{y} - X\boldsymbol{\beta})\right\}. \end{aligned}$$

Normal Linear Regression (3)

- Note that the model parameters are $\boldsymbol{\beta}$ and σ^2. In this chapter, we assumes prior independence between$\boldsymbol{\beta}$ and σ^2, that is, we take $p(\boldsymbol{\beta}, \sigma^2) = p(\boldsymbol{\beta})p(\sigma^2)$. Specifically, we take the following priors:

$$\boldsymbol{\beta} \sim \mathsf{N}(\boldsymbol{\mu}_0, \Sigma_0), \quad \sigma^2 \sim \mathsf{IG}(\alpha_0, \beta_0).$$

- Namely

$$p(\boldsymbol{\beta}) = |2\pi\Sigma_0|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(\boldsymbol{\beta} - \boldsymbol{\mu}_0)^\top \Sigma_0^{-1}(\boldsymbol{\beta} - \boldsymbol{\mu}_0)\right\},$$

$$p(\sigma^2) = \frac{\beta_0^{\alpha_0}}{\Gamma(\alpha_0)} \left(\sigma^2\right)^{-(\alpha_0+1)} \mathrm{e}^{-\frac{\beta_0}{\sigma^2}}.$$

- We would like to derive the Gibbs sampler for the linear model. To do so, we need to derive the conditional densities $p(\sigma^2 \mid \boldsymbol{\beta}, \boldsymbol{y})$, and $p(\boldsymbol{\beta} \mid \sigma^2, \boldsymbol{y})$.

Conditional densities

$$
\begin{aligned}
p(\sigma^2 \mid \boldsymbol{\beta}, \boldsymbol{y}) &\propto p(\boldsymbol{y} \mid \boldsymbol{\beta}, \sigma^2) p(\boldsymbol{\beta}) \propto p(\boldsymbol{y} \mid \boldsymbol{\beta}, \sigma^2) p(\sigma^2) \\
&\propto \left(\sigma^2\right)^{-\frac{n}{2}} \exp\left\{-\frac{1}{2\sigma^2}(\boldsymbol{y} - X\boldsymbol{\beta})^\top (\boldsymbol{y} - X\boldsymbol{\beta})\right\} \times \\
&\qquad \times \left(\sigma^2\right)^{-(\alpha_0+1)} \mathrm{e}^{-\frac{\beta_0}{\sigma^2}} \\
&\propto \left(\sigma^2\right)^{-\left(\frac{n}{2}+\alpha_0+1\right)} \exp\left\{-\frac{1}{\sigma^2}\left[\beta_0 + \frac{1}{2}(\boldsymbol{y} - X\boldsymbol{\beta})^\top (\boldsymbol{y} - X\boldsymbol{\beta})\right]\right\}.
\end{aligned}
$$

That is,

$$
(\sigma^2 \mid \boldsymbol{\beta}, \boldsymbol{y}) \sim \mathsf{IG}\left(\frac{n}{2} + \alpha_0,\ \beta_0 + \frac{1}{2}(\boldsymbol{y} - X\boldsymbol{\beta})^\top (\boldsymbol{y} - X\boldsymbol{\beta})\right). \qquad (10)
$$

The development of $p(\boldsymbol{\beta} \mid \sigma^2, \boldsymbol{y})$ is a bit more involved, but one can show that

$$
(\boldsymbol{\beta}, \mid \sigma^2, \boldsymbol{y}) \sim \mathsf{N}\left(\left[\frac{1}{\sigma^2}X^\top X + \Sigma_0^{-1}\right]^{-1}\left[\frac{1}{\sigma^2}X^\top \boldsymbol{y} + \Sigma_0^{-1}\boldsymbol{\mu}_0\right],\ \left[\frac{1}{\sigma^2}X^\top X + \Sigma_0^{-1}\right]^{-1}\right). \qquad (11)
$$

The Gibbs sampling algorithm

The Gibbs sampler for the linear regression model as summarized in Algorithm 9.

Algorithm 9: Gibbs Sampler for the Linear Regression Model

```
Pick initial $\boldsymbol{\beta}^{(0)}$ and $\sigma^{2\,(0)}$
for $r = 1$ to $R$ do
    Draw $(\sigma^{2\,(r)} \mid \boldsymbol{\beta}^{(r-1)}, \boldsymbol{y})$ via (10)
    Draw $(\boldsymbol{\beta}^{(r)} \mid \sigma^{2\,(r)}, \boldsymbol{y})$ via (11)
end
```

Normal Linear Regression exercise (1)

- Generates a sample of $n = 500$ observations from a normal linear regression model.
- Implement the Gibbs sampler from Algorithm 9, where the sampler is initialized using the least squares estimate.
- Calculate the posterior means of the model parameters and the corresponding 95% credible intervals.

```
% NormalLinearModelMCMC.m
rng(12345);
% artificial data generation
n = 500;
dim = 2;
X = [ones(n,1)];
for i=1:dim
    X = [X rand(n,1)];
end
beta_real = [3;4;-5];
sigma2_real = 2;
y = X*beta_real + randn(n,1).*sqrt(sigma2_real);

% prior
alpha0 = 3; beta0 = 2;
mu0 = zeros(dim+1,1);
sigma0_inv = eye(dim+1)/100;

N = 10000;
burnin = 1000;
```

```
% initialize the chain
beta = X\y;
sig2 = var(y-X*beta);
ell = zeros(N,dim+1+1);
for i=1:N+burnin
    % sample sig2 from inverse gamma
    alpha1 = alpha0 + n/2;
    tmp = y-X*beta;
    beta1 = beta0 + 0.5*tmp’*tmp;
    sig2 =  igrnd(alpha1,beta1);

    % sample beta from multi var norm
    Sigma1 = inv(sig2^-1 * X’*X + sigma0_inv);
    mu1 = Sigma1 * (sig2^-1 * (X’*y)  + sigma0_inv*mu0);

    beta = mvnormalrnd(mu1,Sigma1);

    if(i>burnin)
        ell(i-burnin,:) = [beta’,sig2];
    end
end
```

```
fprintf("beta_hat, sigma_hat, %d,%d,%d, | %d \n",  mean(ell));
% credible intervalse
for i=1:size(ell,2)
    arr = ell(:,i);
    fprintf("(%d,%d)\n", quantile(arr,0.05/2), quantile(arr,1-(0.05/2)));
end
```

```
>> NormalLinearModelMCMC
beta_hat, sigma_hat, 2.974711e+00,4.224208e+00,-5.267312e+00, | 1.852649e+00
(2.651523e+00,3.293831e+00)
(3.807479e+00,4.638206e+00)
(-5.686864e+00,-4.847202e+00)
(1.638106e+00,2.091179e+00)
```

Linear Regression with t errors (1)

- Many studies demonstrated that models with heavier tails than those of normal distributions generally fit financial and macroeconomic data better.
- A random variable X follows a Student's t distribution if its density function is given by

$$f(x) = \frac{\Gamma((v+1)/2)}{\sqrt{v\pi\sigma^2}\Gamma(v/2)} \left(1 + \frac{(x-\mu)^2}{v\sigma^2}\right)^{-\frac{v+1}{2}}$$

- We say that $X \sim \mathcal{T}_v(\mu, \sigma^2)$. However, if we assume that the errors $\{\epsilon_t\}$ are iid $\mathcal{T}_v(\mu, \sigma^2)$, the posterior distribution of $\boldsymbol{\beta}$ is not normal anymore.
- Instead of working with the t distribution directly, we work with an important idea of latent variables such that given these latent variables, standard estimation methods can be used.

Linear Regression with t errors (2)

Theorem

Let $(X \mid \lambda) \sim \mathsf{N}(\mu, \lambda\sigma^2)$, *and* $\lambda \sim \mathsf{IG}(v/2, v/2)$. *Then, the distribution of* X *unconditionally of* λ *is* $\mathcal{T}_v(\mu, \sigma^2)$.

Proof:

Note that

$$\begin{aligned} f(x,\lambda) = f(x \mid \lambda) f(\lambda) &= (2\pi\lambda\sigma^2)^{-\frac{1}{2}} \exp\left\{-\frac{1}{2\lambda\sigma^2}(x-\mu)^2\right\} \times \\ &\times \frac{(v/2)^{(v/2)}}{\Gamma(v/2)} \lambda^{-((v/2)+1)} e^{-\frac{(v/2)}{\lambda}} \\ &= (2\pi\sigma^2)^{-\frac{1}{2}} \frac{(v/2)^{v/2}}{\Gamma(v/2)} \lambda^{-\frac{v+1}{2}+1} e^{-\frac{v}{2\lambda}\left(1+\frac{(x-\mu)^2}{v\sigma^2}\right)}. \end{aligned}$$

Linear Regression with t errors (3)

The completion of the proof follows from the identity

$$\int_0^\infty z^{-(\alpha+1)} e^{-\frac{\beta}{z}} \mathrm{d}z = \beta^{-\alpha}\Gamma(\alpha).$$

The marginal density of X id given by:

$$\begin{aligned} f(x) &= \int_0^\infty f(x,\lambda)\mathrm{d}\lambda = (2\pi\sigma^2)^{-\frac{1}{2}} \frac{(v/2)^{v/2}}{\Gamma(v/2)} \int_0^\infty \lambda^{-\frac{v+1}{2}+1} e^{-\frac{v}{2\lambda}\left(1+\frac{(x-\mu)^2}{v\sigma^2}\right)} \mathrm{d}\lambda \\ &= (2\pi\sigma^2)^{-\frac{1}{2}} \frac{(v/2)^{v/2}}{\Gamma(v/2)} \left[\left(\frac{v}{2}\right)^{-\frac{v+1}{2}} \left(1+\frac{(x-\mu)^2}{v\sigma^2}\right)^{-\frac{v+1}{2}} \Gamma((v+1)/2) \right] \\ &= \frac{\Gamma((v+1)/2)}{\sqrt{v\pi\sigma^2}\,\Gamma(v/2)} \left(1+\frac{(x-\mu)^2}{v\sigma^2}\right)^{-\frac{v+1}{2}}. \end{aligned}$$

We can exploit Theorem 5 to generate t-distributed random variables. The example code is given below.

Generating t-distributed random variables

```
%latent_t_dist.m
% generate t-distributioned numbers
rng(12345);
N = 1000;
mu = 8;
sigma = 2;
v = 8;

Z = trnd(8,N,1);
T = Z*sigma + mu;

% latent variable
lambda = 1./gamrnd((v/2),1/(v/2),N,1);
T2 = normrnd(mu,(lambda.^0.5)*sigma);

cdfplot(T)
hold on
cdfplot(T2)
```

Linear Regression with t errors (v) is known (1)

- We considered the regression model in the matrix notation, namely,

$$\boldsymbol{y} = X\boldsymbol{\beta} + \boldsymbol{\epsilon}.$$

- We suppose now that

$$\epsilon \sim \mathrm{MVN}(0, \Sigma_{\boldsymbol{\theta}}),$$

  where $\Sigma_{\boldsymbol{\theta}}$ is a general $n \times n$ Covariance Matrix.
- In this section, $(\boldsymbol{\epsilon} \mid \Lambda) \sim \mathrm{MVN}(0, \sigma^2\Lambda)$, where $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$, and $\lambda_i \sim \mathrm{IG}(v/2, v/2)$ for $i = 1, \ldots, n$. That is, $\Sigma_{\boldsymbol{\theta}} = \sigma^2\Lambda$.
- The likelihood function (where $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_n)^\top$) is therefore

$$\begin{aligned} p(\boldsymbol{y} \mid \boldsymbol{\beta}, \sigma^2, \boldsymbol{\lambda}, v) &= |2\pi\sigma^2\Lambda|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(\boldsymbol{y} - X\boldsymbol{\beta})^\top (\sigma^2\Lambda)^{-1}(\boldsymbol{y} - X\boldsymbol{\beta})\right\} \\ &= \left(2\pi\sigma^2\right)^{-\frac{n}{2}} |\Lambda|^{-1/2} \exp\left\{-\frac{1}{2}(\boldsymbol{y} - X\boldsymbol{\beta})^\top (\sigma^2\Lambda)^{-1}(\boldsymbol{y} - X\boldsymbol{\beta})\right\}. \end{aligned}$$

Linear Regression with t errors (v) is known (2)

- We assume the independent priors $p(\boldsymbol{\beta}, \sigma^2, \boldsymbol{\lambda}) = p(\boldsymbol{\beta})p(\sigma^2)p(\boldsymbol{\lambda})$ with $\boldsymbol{\beta} \sim \mathsf{N}(\boldsymbol{\mu}_0, \Sigma_0)$ as before.
- Using our regular analysis (see (11)), we can show that:

$$(\boldsymbol{\beta}, \mid \boldsymbol{y}, \boldsymbol{\lambda}, \sigma^2) \sim \mathsf{N}\Bigg(\left[X^\top (\sigma^2 \Lambda)^{-1} X + \Sigma_0^{-1}\right]^{-1} \left[X^\top (\sigma^2 \Lambda)^{-1} \boldsymbol{y} + \Sigma_0^{-1} \boldsymbol{\mu}_0\right], \\ \left[X^\top (\sigma^2 \Lambda)^{-1} X + \Sigma_0^{-1}\right]^{-1} \Bigg). \tag{12}$$

- Please note that

$$(\sigma^2 \Lambda)^{-1} = \frac{1}{\sigma^2} \mathrm{diag}(\lambda_1^{-1}, \ldots, \lambda_n^{-1}).$$

- To complete the Gibbs sampler, we need to sequentially draw from the following three conditional densities: $p(\boldsymbol{\beta} \mid \boldsymbol{y}, \boldsymbol{\lambda}, \sigma^2)$ (done), $p(\boldsymbol{\lambda} \mid \boldsymbol{y}, \boldsymbol{\beta}, \sigma^2)$, and $p(\sigma^2 \mid \boldsymbol{y}, \boldsymbol{\beta}, \boldsymbol{\lambda})$.

Linear Regression with t errors (v) is known (3)

We omit the details, but note that one can derive the following.

$$(\sigma^2 \mid \boldsymbol{y}, \boldsymbol{\beta}, \boldsymbol{\lambda}, v) \sim \mathsf{IG}\left(\alpha_0 + \frac{n}{2},\ \beta_0 + \frac{1}{2}(\boldsymbol{y} - X\boldsymbol{\beta})^\top \Lambda^{-1} (\boldsymbol{y} - X\boldsymbol{\beta})\right). \quad (13)$$

$$(\boldsymbol{\lambda} \mid \boldsymbol{y}, \sigma^2, \boldsymbol{\beta}, v) \sim \mathsf{IG}\left(\frac{v+1}{2},\ \frac{1}{2}\left(\frac{(y_i - \boldsymbol{x}_i \boldsymbol{\beta})^2}{\sigma^2} + v\right)\right). \quad (14)$$

The Gibbs sampler

Algorithm 10: Gibbs Sampler for the Linear Regression Model with t errors

```
Pick initial $\boldsymbol{\beta}^{(0)}$ and $\sigma^{2\,(0)}$. Assume that the degree of freedom $v$ is known.
**for** $r = 1$ **to** $R$ **do**
    Draw $\lambda_t^{(r)} \sim p(\lambda_i \mid \boldsymbol{y}, \sigma^{2\,(r-1)}, \boldsymbol{\beta}^{(r-1)}, v)$ from (14) (inverse gamma) for $t = 1, \ldots, n$.
    Draw $\boldsymbol{\beta}^{(r)} \sim p(\boldsymbol{\beta} \mid \boldsymbol{y}, \sigma^{2\,(r-1)}, \boldsymbol{\lambda}^{(r)}, v)$ via (12) (multivariate normal)
    Draw $\sigma^{2\,(r)} \sim p(\sigma^2 \mid \boldsymbol{y}, \boldsymbol{\beta}^{(r)}, \boldsymbol{\lambda}^{(r)}, v)$ via (13) (inverse gamma)
**end**
```

Linear Regression with t errors (v) is known example (1)

- Generate a hand-made data-set (500 samples) with 3 explanatory variables generated from uniform distribution ($\mathrm{U}(-1,1)$) and $\boldsymbol{\beta} = (1, 1.5, -3, 2)$ and using $\sigma^2 = 0.2$ and $v = 5$ degrees of freedom.
- Write a program that estimates the model parameters.

```
% t_lin_reg1.m
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% generate custom dataset
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
n = 500;
sigma_sqr = 0.2;
v = 5;

beta_original = [1,1.5,-3,2]';
X = [ones(n,1) -1 + 2*rand(n,3)];
y = X*beta_original + trnd(v,n,1)*sigma_sqr^0.5;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% prior for sigma
alpha0 = 3;
beta0 = 2;

% prior for betta
mu0 = zeros(4,1);
sigma0_inv = eye(4)/100;

N = 10000;
burnin = 5000;
%  array to store \beta and sigma squared
ell = zeros(N,5);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

beta = (X'*X)\(X'*y);
sigma_sq = var(X*beta - y);
```

```
for r=1:N+burnin
    %  sample lambda from inverse gamma
    igparams = ((X*beta - y)).^2/sigma_sq;
    lambda = 1./gamrnd(((v+1)/2),1./(0.5*(v+igparams)));

    %  sample beta from multi-variate normal
    LAMBDA = diag(lambda.^-1);
    sigLAMBDA = LAMBDA/sigma_sq;
    SIG = (X'*sigLAMBDA*X + sigma0_inv)^-1;
    mu = SIG*(X'*sigLAMBDA*y + sigma0_inv*mu0 );
    beta = mvnrnd(mu,SIG)';

    % sample sigma
    sigma_sq = 1./gamrnd(alpha0 + n/2, 1/(beta0 + 0.5*(y-X*beta)'* ...
        LAMBDA*(y-X*beta)));

    if(r>=burnin+1)
        ell(r-burnin,1:4) = beta;
        ell(r-burnin,5) = sigma_sq;
    end
end
```

```
disp("original paraneters")
disp("%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%")
fprintf("beta = [%d,%d,%d,%d], sigma = %d \n",beta_original,sigma_sqr)
disp("%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%")

averages = mean(ell,1);
disp("estimated parameters")
disp("%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%")
fprintf("beta_hat = [%d,%d,%d,%d], sigma_sqr_hat = %d \n",
             averages(1:4),averages(5))
disp("%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%")


disp("credible intervals")
disp("%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%")
ci_1 = quantile(ell,0.05/2,1);
ci_2 = quantile(ell,1-0.05/2,1);

for i=1:5
    fprintf("(%d,%d)\n",ci_1(i),ci_2(i));
end
```

```
>>t_lin_reg1
original paraneters
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
beta = [1,1.500000e+00,-3,2], sigma = 2.000000e-01
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
estimated parameters
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
beta_hat = [1.003525e+00,1.498303e+00,-3.028266e+00,2.040767e+00],
sigma_sqr_hat = 1.789911e-01
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
credible intervals
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
(9.607075e-01,1.046043e+00)
(1.424598e+00,1.571738e+00)
(-3.098807e+00,-2.956701e+00)
(1.967620e+00,2.115815e+00)
(1.534038e-01,2.083897e-01)
```

Summary

- Bayesian statistics is very powerful.
- Priors are an issue. If we have a good “prior” knowledge about the parameters, Bayesian statistics is very appealing.
- We did not cover model comparison and prediction, but these are well defined via the posterior distribution.
- Bayesian statistics require heavy computational resources.
- Therefore, the Bayesian paradigm is rarely used for large datasets. (However, a lot of effort is invested to fix this.)