

Bayesian Linear Regression

Guest lecture at KTH 2022

Mattias Villani

Department of Statistics
Stockholm University

Department of Computer and Information Science
Linköping University



mattiasvillani.com



@matvil



[@mattiasvillani](https://www.instagram.com/mattiasvillani)

Lecture overview

- Bayesian inference
- The **normal model** with known variance
- Linear regression
- Regularization priors
- Outlook: Bayes in complex problems

Slides on course page and at: <https://mattiasvillani.com/news>

Rough draft book at: <https://github.com/mattiasvillani/BayesianLearningBook>

Likelihood function - normal data

- Normal data with known variance:

$$X_1, \dots, X_n | \theta \stackrel{iid}{\sim} N(\theta, \sigma^2).$$

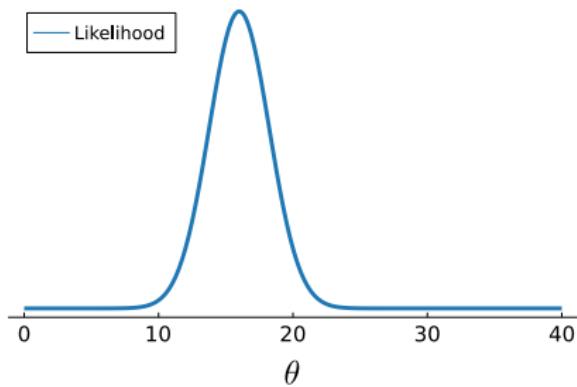
- Likelihood from independent observations: x_1, \dots, x_n

$$\begin{aligned} p(x_1, \dots, x_n | \theta) &= \prod_{i=1}^n p(x_i | \theta) = \frac{1}{(2\pi\sigma^2)^{n/2}} \exp\left(-\frac{1}{2\sigma^2} \sum_{i=1}^n (x_i - \theta)^2\right) \\ &\propto \exp\left(-\frac{1}{2(\sigma^2/n)} (\theta - \bar{x})^2\right) \end{aligned}$$

- Maximum likelihood: $\hat{\theta} = \bar{x}$ maximizes $p(x_1, \dots, x_n | \theta)$.
- Given the data x_1, \dots, x_n , plot $p(x_1, \dots, x_n | \theta)$ as a function of θ .

Am I really getting my 20Mbit/sec?

- I have a 50Mbit/sec internet connection.
- ISP promises at least 20Mbit/sec on average.
- Data:** $x = (15.77, 20.5, 8.26, 14.37, 21.09)$ Mbit/sec.
- Measurement errors:** $\sigma = 5$ (± 10 Mbit with 95% probability)
- The likelihood function is proportional to $N(\bar{x}, \sigma^2 / n)$ density.



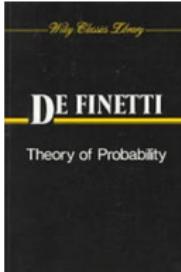
The likelihood function

*The likelihood function is
the probability of the observed data
considered as a function of the parameter.*

- Likelihood function is **NOT** a probability distribution for θ .
- Statements like $\Pr(\theta \geq 20 | \text{data})$ makes no sense.
- Unless ...

Uncertainty and subjective probability

- $\Pr(\theta \geq 20 | \text{data})$ only makes sense if θ is random.
- But θ may be a fixed natural constant?
- **Bayesian: doesn't matter if θ is fixed or random.**
- Do **You** know the value of θ or not?
- $p(\theta)$ reflects Your knowledge/**uncertainty** about θ .
- **Subjective probability**.
- The statement $\Pr(10\text{th decimal of } \pi = 9) = 0.1$ makes sense.

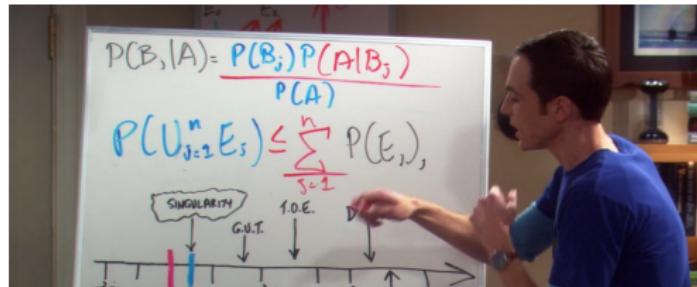


Bayesian learning

- Bayesian learning about a model parameter θ :
 - ▶ state your prior knowledge as a probability distribution $p(\theta)$.
 - ▶ collect data x and form the likelihood function $p(x|\theta)$.
 - ▶ combine prior knowledge $p(\theta)$ with data information $p(x|\theta)$.

- How to combine the two sources of information?

Bayes' theorem



A photograph of Sheldon Cooper from 'The Big Bang Theory' standing in front of a whiteboard. He is wearing a blue t-shirt and has his hands clasped together. On the whiteboard, there are handwritten mathematical equations. The top equation is $P(B_i|A) = \frac{P(B_i)P(A|B_i)}{P(A)}$. Below it is another equation: $P(\cup_{j=1}^n E_j) \leq \sum_{j=1}^n P(E_j)$. At the bottom of the board, there is a diagram with arrows pointing to labels: 'SINGULARITY' (pink arrow), 'G.U.T.' (blue arrow), '1.D.E.' (green arrow), and 'D' (orange arrow). The background shows a living room setting.

$$P(B_i|A) = \frac{P(B_i)P(A|B_i)}{P(A)}$$
$$P(\cup_{j=1}^n E_j) \leq \sum_{j=1}^n P(E_j)$$

Learning from data - Bayes' theorem

- How to **update** from **prior** $p(\theta)$ to **posterior** $p(\theta|\text{Data})$?
- **Bayes' theorem** for events A and B

$$p(A|B) = \frac{p(B|A)p(A)}{p(B)}.$$

- Bayes' Theorem for a model parameter θ

$$p(\theta|\text{Data}) = \frac{p(\text{Data}|\theta)p(\theta)}{p(\text{Data})}.$$

- It is the prior $p(\theta)$ that takes us from $p(\text{Data}|\theta)$ to $p(\theta|\text{Data})$.
- A probability distribution for θ is extremely useful:
 - ▶ **Predictions**
 - ▶ **Decision making**
 - ▶ **Regularization**

Great theorems make great tattoos

- Bayes theorem

$$p(\theta|\text{Data}) = \frac{p(\text{Data}|\theta)p(\theta)}{p(\text{Data})}$$

- All you need to know:

$$p(\theta|\text{Data}) \propto p(\text{Data}|\theta)p(\theta)$$

or

Posterior \propto Likelihood \cdot Prior



$$p(\theta|x) \propto p(x|\theta) \cdot p(\theta)$$

Normal data, known variance - normal prior

■ Prior

$$\theta \sim N(\mu_0, \tau_0^2)$$

■ Posterior

$$\begin{aligned} p(\theta|x_1, \dots, x_n) &\propto p(x_1, \dots, x_n|\theta, \sigma^2)p(\theta) \\ &\propto N(\theta|\mu_n, \tau_n^2), \end{aligned}$$

where the **posterior mean** is

$$\mu_n = w\bar{x} + (1-w)\mu_0$$

$$w = \frac{\frac{n}{\sigma^2}}{\frac{n}{\sigma^2} + \frac{1}{\tau_0^2}}.$$

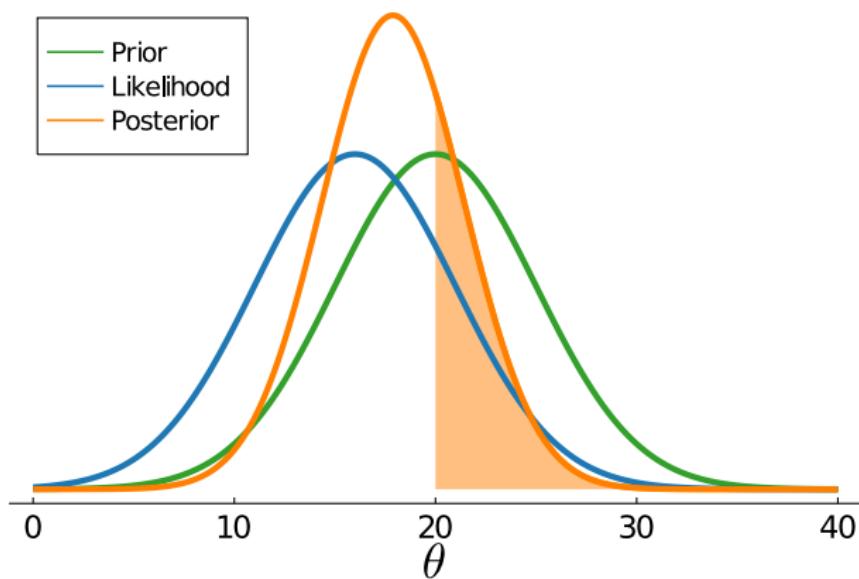
- Define: Precision $\equiv 1/\text{Variance}$.
- Posterior precision = Data precision + Prior precision

$$\frac{1}{\tau_n^2} = \frac{n}{\sigma^2} + \frac{1}{\tau_0^2}$$

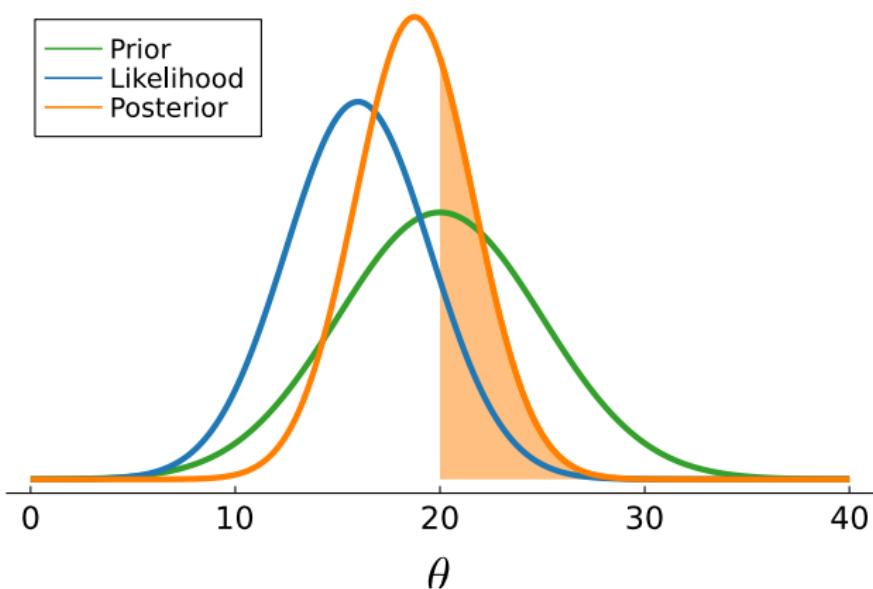
Download speed

- **Data:** $x = (15.77, 20.5, 8.26, 14.37, 21.09)$ Mbit/sec.
- **Model:** $X_1, \dots, X_5 \sim N(\theta, \sigma^2)$.
- Assume $\sigma = 5$ (measurements can vary ± 10 MBit with 95% probability)
- My **prior:** $\theta \sim N(20, 5^2)$.

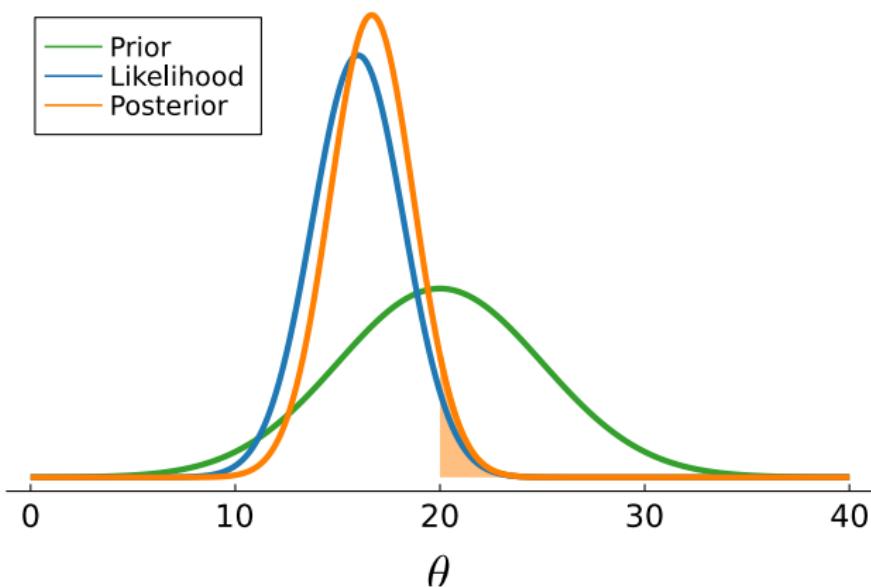
Download speed $n=1$



Download speed $n=2$

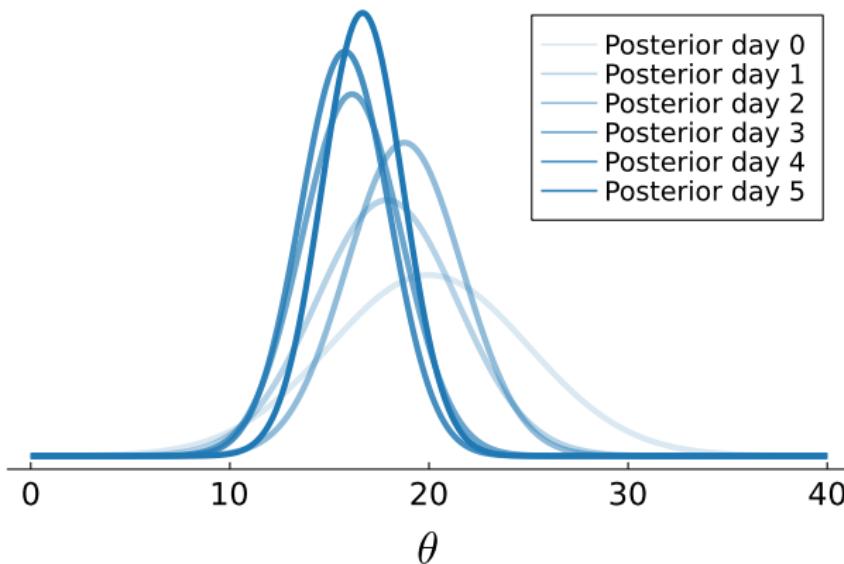


Download speed $n=5$



Bayesian Online learning

- Yesterday's posterior is today's prior.



Bayesian Prediction

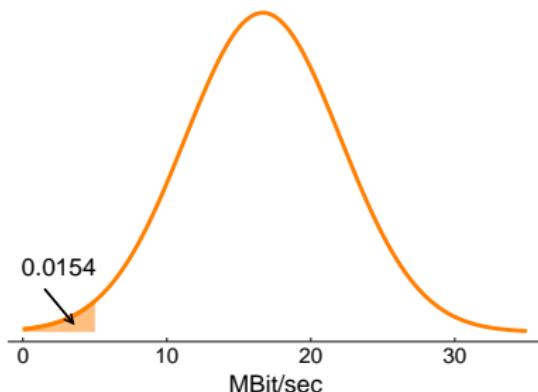
- **Predictive distribution** averages over the unknown parameter

$$\underbrace{p(x_{n+1} | x_{1:n})}_{\text{predictive dist}} = \int \underbrace{p(x_{n+1} | \theta)}_{\text{model}} \underbrace{p(\theta | x_{1:n})}_{\text{posterior}} d\theta$$

- Normal data, normal prior:

$$x_{n+1} | x_{1:n} \sim N(\mu_n, \sigma^2 + \tau_n^2)$$

- My streaming buffers whenever $x < 5$ MBit/Sec.



Linear regression

- The linear regression model in **matrix form**

$$\underset{(n \times 1)}{y} = \underset{(n \times k)(k \times 1)}{X\beta} + \underset{(n \times 1)}{\varepsilon}$$

- First column of X is the unit vector and β_1 is the intercept.
- Normal errors: $\varepsilon_i \stackrel{iid}{\sim} N(0, \sigma^2)$, so $\varepsilon \sim N(0, \sigma^2 I_n)$.
- Likelihood**

$$y | \beta, \sigma^2, X \sim N(X\beta, \sigma^2 I_n)$$

Linear regression - uniform prior

- Standard **non-informative prior**: uniform on $(\beta, \log \sigma^2)$

$$p(\beta, \sigma^2) \propto \sigma^{-2}$$

- **Joint posterior** of β and σ^2 :

$$\begin{aligned}\beta | \sigma^2, y &\sim N\left[\hat{\beta}, \sigma^2(X^\top X)^{-1}\right] \\ \sigma^2 | y &\sim \text{Inv-}\chi^2(n - k, s^2)\end{aligned}$$

where $\hat{\beta} = (X^\top X)^{-1} X^\top y$ and $s^2 = \frac{1}{n-k}(y - X\hat{\beta})^\top(y - X\hat{\beta})$.

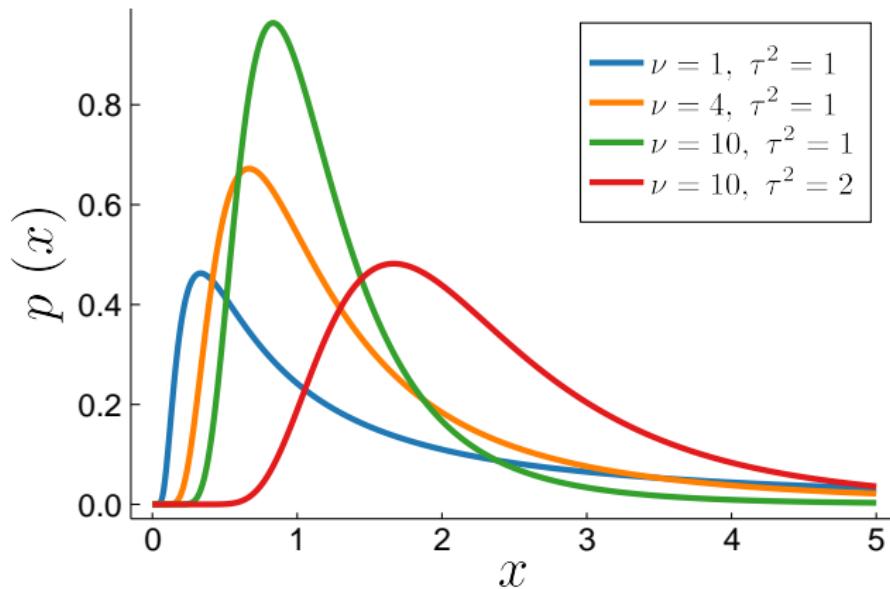
- **Simulate** from the joint posterior by simulating from

- ▶ $p(\sigma^2 | y)$
- ▶ $p(\beta | \sigma^2, y)$

- **Marginal posterior** of β :

$$\beta | y \sim t_{n-k} \left[\hat{\beta}, s^2(X^\top X)^{-1} \right]$$

Scaled inverse χ^2 distribution



■ Inverse gamma distribution.

Linear regression - conjugate prior

■ Joint prior for β and σ^2

$$\begin{aligned}\beta | \sigma^2 &\sim N(\mu_0, \sigma^2 \Omega_0^{-1}) \\ \sigma^2 &\sim \text{Inv-}\chi^2(\nu_0, \sigma_0^2)\end{aligned}$$

■ Posterior

$$\begin{aligned}\beta | \sigma^2, y &\sim N[\mu_n, \sigma^2 \Omega_n^{-1}] \\ \sigma^2 | y &\sim \text{Inv-}\chi^2(\nu_n, \sigma_n^2)\end{aligned}$$

$$\mu_n = (X^\top X + \Omega_0)^{-1} (X^\top y + \Omega_0 \mu_0)$$

$$\Omega_n = X^\top X + \Omega_0$$

$$\nu_n = \nu_0 + n$$

$$\nu_n \sigma_n^2 = \nu_0 \sigma_0^2 + (y^\top y + \mu_0^\top \Omega_0 \mu_0 - \mu_n^\top \Omega_n \mu_n)$$

Bayesian Linear regression in Julia 😍

```
function BayesLinReg(y::Vector, X, μ₀, Ω₀, ν₀, σ²₀, nSim)

    # Compute posterior hyperparameters
    n = length(y)
    p = size(X,2)
    XX = X'*X
    βhat = X \ y
    Ωₙ = Symmetric(XX + Ω₀)
    μₙ = Ωₙ\ (XX*βhat + Ω₀*μ₀)
    νₙ = ν₀ + n
    σ²ₙ = (ν₀*σ²₀ + (y-X*βhat)'*(y-X*βhat) +
            (μₙ-βhat)'*XX*(μₙ-βhat) +
            (μₙ-μ₀)'*Ω₀*(μₙ-μ₀))
    /νₙ

    # Sampling from posterior
    invΩₙ = inv(Ωₙ)
    σ²sim = zeros(nSim)
    βsim = zeros(nSim,p)
    for i ∈ 1:nSim
        # Simulate from p(σ²|y,X)
        σ² = rand(ScaledInverseChiSq(νₙ,σ²ₙ))
        σ²sim[i] = σ²

        # Simulate from p(β|σ²,y,X)
        β = rand(MvNormal(μₙ,σ²*invΩₙ))
        βsim[i,:] = β'

    end
    return μₙ, Ωₙ, νₙ, σ²ₙ, βsim, σ²sim
end
```

Bike share data

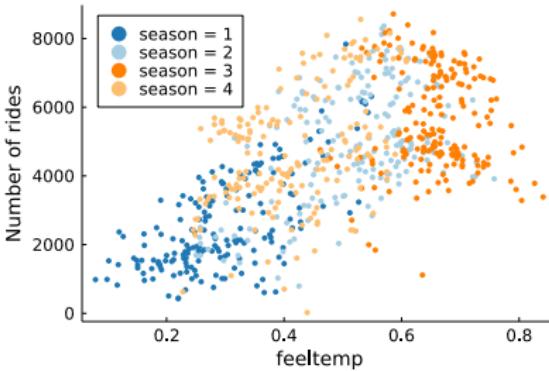
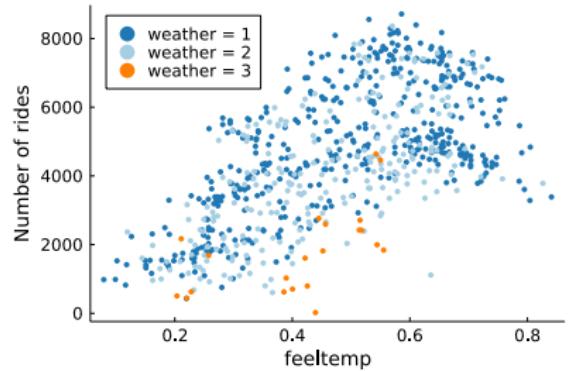
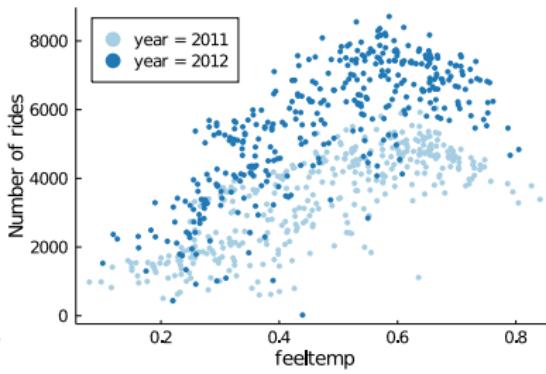
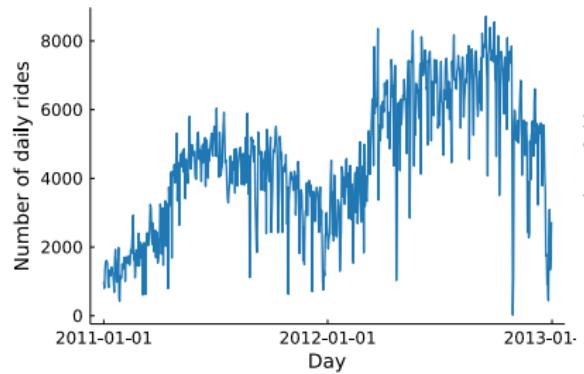
- Bike share data. Predict the number of bike rides.
- Response variable: number of rides on 731 days.

variable	description	data type	values	comment
nrides	number of rides	counts	{0, 1, ...}	min= 22, max= 8714
feeltemp	perceived temp	continuous	[0, 1]	min= 0.07, max= 0.85
hum	humidity	continuous	[0, 1]	min= 0.00, max= 0.98
wind	wind speed	continuous	[0, 1]	min= 0.02, max= 0.51
year	year	binary	{0, 1}	year 2011 = 0
season	season	categorical	{1, 2, 3, 4}	winter → fall
weather	weather	ordinal	{1, 2, 3}	clear → rain/snow
weekday	day of week	categorical	{0, 1, ..., 6}	sunday → saturday
holiday	holiday	binary	{0, 1}	holiday = 1

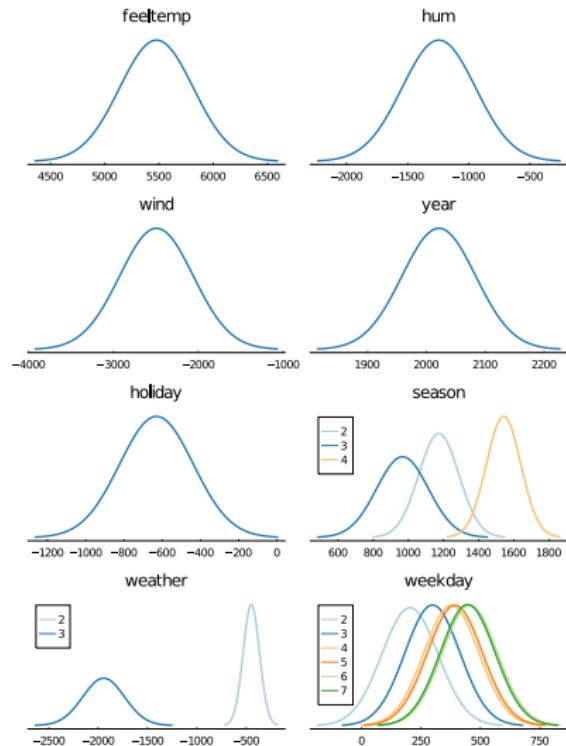
- Prior:

- ▶ $\mu_0 = (1000, 0, \dots, 0)^\top$
- ▶ $\Omega_0 = \frac{1}{n} X^\top X$
- ▶ $\sigma_0^2 = 1000^2$ and $v_0 = 5$.

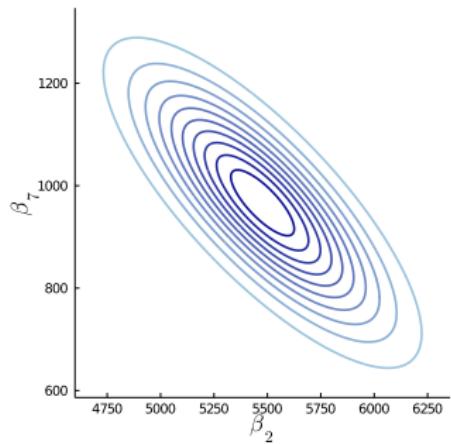
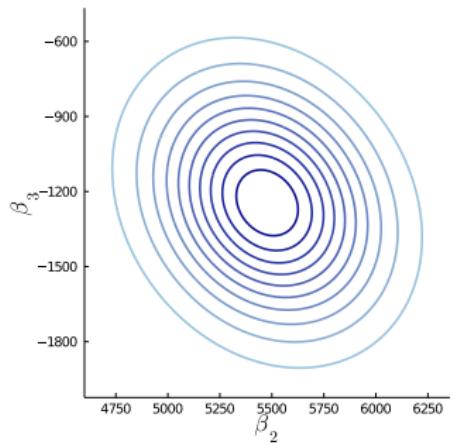
Bike share data



Bike share data - marginal posteriors of β



Bike share data - joint posteriors of β



Ridge regression = normal prior

- Smoothness/shrinkage/regularization prior $[\Omega_0 = \lambda I]$

$$\beta_i | \sigma^2 \stackrel{\text{iid}}{\sim} N\left(0, \frac{\sigma^2}{\lambda}\right)$$

- Posterior mean is the **ridge regression** estimator

$$\mu_n = (X^\top X + \lambda I)^{-1} X^\top y$$

- Shrinkage toward zero

As $\lambda \rightarrow \infty$, $\mu_n \rightarrow 0$

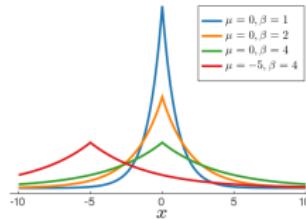
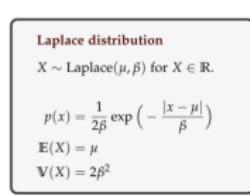
- When $X^\top X = I$

$$\mu_n = \frac{1}{1 + \lambda} \hat{\beta}$$

Lasso regression = Laplace prior

- Lasso is equivalent to posterior mode under Laplace prior

$$\beta_i | \sigma^2 \stackrel{\text{iid}}{\sim} \text{Laplace}\left(0, \frac{\sigma^2}{\lambda}\right)$$



- **Laplace prior:**
 - ▶ heavy tails
 - ▶ many β_i close to zero, but some β_i can be very large.
- **Normal prior:**
 - ▶ light tails
 - ▶ all β_i 's are similar in magnitude and no β_i very large.

Learning the optimal shrinkage

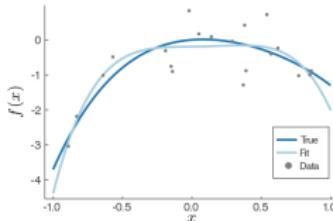
- Cross-validation is often used to determine λ .
- Bayesian: λ is **unknown** \Rightarrow **use a prior** for λ .
- $\lambda \sim \text{Inv-}\chi^2(\eta_0, \lambda_0)$. The user specifies η_0 and λ_0 .
- Joint posterior
$$p(\beta, \sigma^2, \lambda | y, X)$$
- Marginal posterior λ .

Polynomial regression

- Polynomial regression is linear in β :

$$f(x_i) = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \dots + \beta_k x_i^k.$$

$$y = X\beta + \varepsilon, \text{ where } X = (1, x, x^2, \dots, x^k).$$



- Problem: higher order polynomials can **overfit** the data.
- Solution: **shrink** higher order coefficients harder:

$$\beta | \sigma^2 \sim N \left[0, \begin{pmatrix} 100 & 0 & 0 & \cdots & 0 \\ 0 & \frac{1}{\lambda} & 0 & \cdots & 0 \\ 0 & 0 & \frac{1}{2\lambda} & & \\ \vdots & \vdots & & \ddots & \\ 0 & 0 & 0 & \cdots & \frac{1}{k\lambda} \end{pmatrix} \right]$$

How long until maximal pain relief?

- Quadratic relationship between pain relief (y) and time (x)

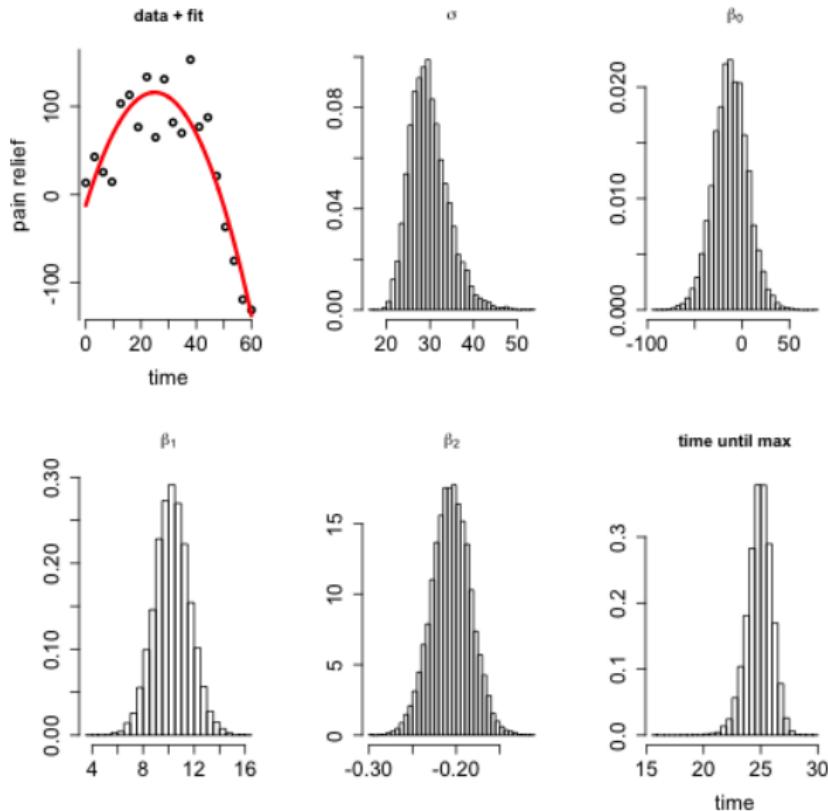
$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \varepsilon.$$

- At what time x_{\max} is there **maximal pain relief**?

$$x_{\max} = -\beta_1 / 2\beta_2$$

- Easy to obtain marginal posterior $p(x_{\max} | y, X)$ by **simulation**:
 - ▶ Simulate N coefficient vectors from the posterior $\beta, \sigma^2 | y, X$
 - ▶ For each simulated β , compute $x_{\max} = -\beta_1 / 2\beta_2$.
 - ▶ Plot a histogram. Converges to $p(x_{\max} | y, X)$ as $N \rightarrow \infty$.

How long until maximal pain relief?



Bayes is easy to use

- Substantially more complex models can be analyzed by
 - ▶ **Markov Chain Monte Carlo** (MCMC) simulation
 - ▶ **Hamiltonian Monte Carlo** (HMC) simulation
 - ▶ **Variational inference** optimization
- **Deep Learning.** Bayes quantifies uncertainty \Rightarrow Probabilistic predictions \Rightarrow Decisions under uncertainty.
- Ongoing research on making Bayes more scalable to large data.
My own contributions: <https://mattiasvillani.com/research>
- Probabilistic programming languages makes Bayes easy:
 - ▶ **Stan (R and more)**
 - ▶ **Turing.jl (Julia)**
 - ▶ **Pyro (Python)**
- Bayesian Learning course at SU (March-April):
<https://github.com/mattiasvillani/BayesLearnCourse>
Engineers welcome!

Poisson regression in Turing.jl (Julia)

■ Poisson regression:

$$y_i | \theta_i \sim \text{Pois}(\exp(\theta_i)), \quad \text{for } i = 1, \dots, n$$

$$\theta_i = x_i^\top \beta$$

$$\beta \sim N(0, \tau_0^2 I)$$

```
# Bayesian poisson regression model in Turing.jl
@model poisson_reg(x, y, τ₀) = begin
    n = length(y)
    β₀ ~ Normal(0, τ₀^2)
    β₁ ~ Normal(0, τ₀^2)
    β₂ ~ Normal(0, τ₀^2)
    β₃ ~ Normal(0, τ₀^2)
    for i = 1:n
        θ = β₀ + β₁*X[i, 1] + β₂*X[i, 2] + β₃*X[i, 3]
        y[i] ~ Poisson(exp(θ))
    end
end

# Simulate from the posterior using HMC with NUTS tuning
sample(poisson_reg(X, y, 10), NUTS(200, 0.65), 2500)
```

■ Deep Neural Net in Turing.jl:

<https://turing.ml/dev/tutorials/03-bayesian-neural-network/>