Bayesian Linear Regression

Guest lecture at KTH 2020

Mattias Villani

Department of Statistics
Stockholm University

Department of Computer and Information Science
Linköping University

Lecture overview

- **Bayesian inference**
- The **normal model** with known variance
- **Linear regression**
- **Regularization priors**

Slides at: https://mattiasvillani.com/news

Likelihood function - normal data regression

- **Normal data** with **known variance**:

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

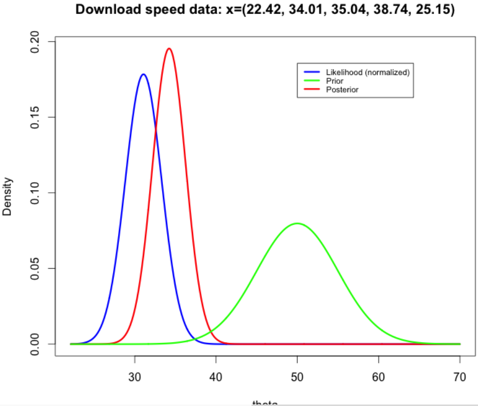
- **Likelihood** from independent observations: $x_1, ..., x_n$

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

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

Am I really getting my 50Mbit/sec?

- My broadband provider promises me at least 50Mbit/sec.
- **Data**: $x = (22.42, 34.01, 35.04, 38.74, 25.15)$ Mbit/sec.
- **Measurement errors**: $\sigma = 5$ ($\pm$10Mbit with 95% probability)
- The likelihood function is proportional to $\mathrm{N}(\bar{x}, \sigma^2/n)$ density.



The likelihood function

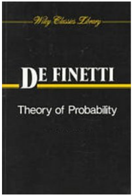
- The mantra:

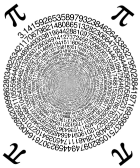
  The likelihood function is
  the probability of the observed data
  considered as a function of the parameter.

- Likelihood function is **NOT** a probability distribution for θ.
- Statements like $\Pr(\theta \geq 50 | \text{data})$ makes no sense.
- Unless ...

Uncertainty and subjective probability

- $\Pr(\theta \geq 50 | \text{data})$ 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** $\mathbf{x}$ and form the **likelihood** function $p(\mathbf{x}|\theta)$.
  - **combine** prior knowledge $p(\theta)$ with data information $p(\mathbf{x}|\theta)$.
- **How to combine** the two sources of information?

Bayes' theorem

Learning from data - Bayes' theorem

- How to **update** from **prior** $p(\theta)$ to **posterior** $p(\theta|Data)$?
- **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

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

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

Great theorems make great tattoos

- Bayes theorem

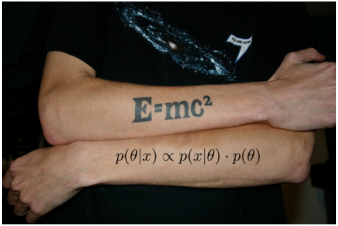
$$p(\theta|Data) = \frac{p(Data|\theta)p(\theta)}{p(Data)}$$

- All you need to know:

$$p(\theta|Data) \propto p(Data|\theta)p(\theta)$$

or

$$\text{Posterior} \propto \text{Likelihood} \cdot \text{Prior}$$



Normal data, known variance - uniform prior

- **Model**

$$x_1, ..., x_n | \theta, \sigma^2 \stackrel{iid}{\sim} N(\theta, \sigma^2).$$

- **Prior**

$$p(\theta) \propto c \text{ (a constant)}$$

- **Likelihood**

$$p(x_1, ..., x_n | \theta, \sigma^2) \quad = \quad \exp\left[-\frac{1}{2(\sigma^2/n)}(\theta - \bar{x})^2\right]$$

- **Posterior**

$$\theta | x_1, ..., x_n \sim N(\bar{x}, \sigma^2/n)$$

Normal data, known variance - normal prior

- **Prior**

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

- **Posterior**

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

where

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

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

and

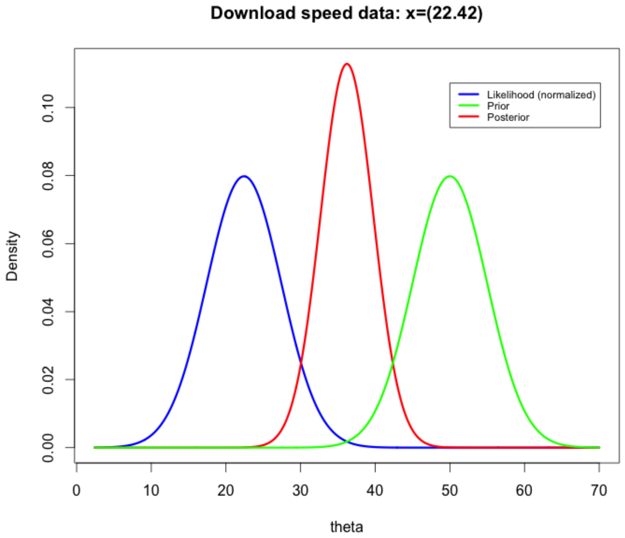
$$w = \frac{\frac{n}{\sigma^2}}{\frac{n}{\sigma^2} + \frac{1}{\tau_0^2}}.$$

- Proof: complete the squares in the exponential.

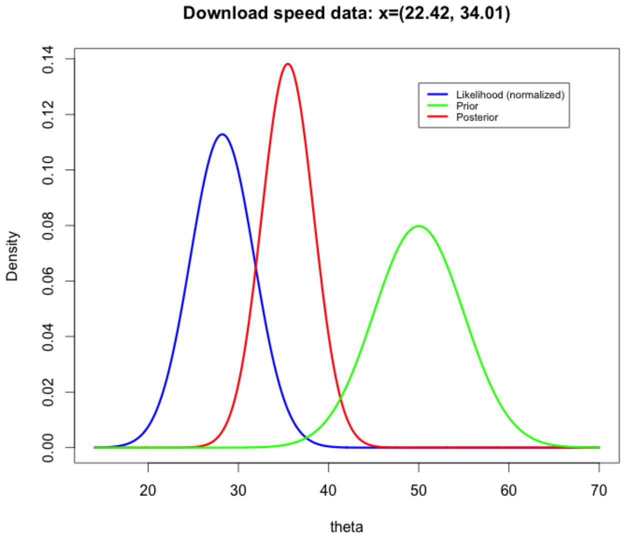
Download speed

- **Data**: $x = (22.42, 34.01, 35.04, 38.74, 25.15)$ Mbit/sec.
- **Model**: $X_1, ..., X_5 \sim N(\theta, \sigma^2)$.
- Assume $\sigma = 5$ (measurements can vary $\pm$10MBit with 95% probability)
- My **prior**: $\theta \sim N(50, 5^2)$.

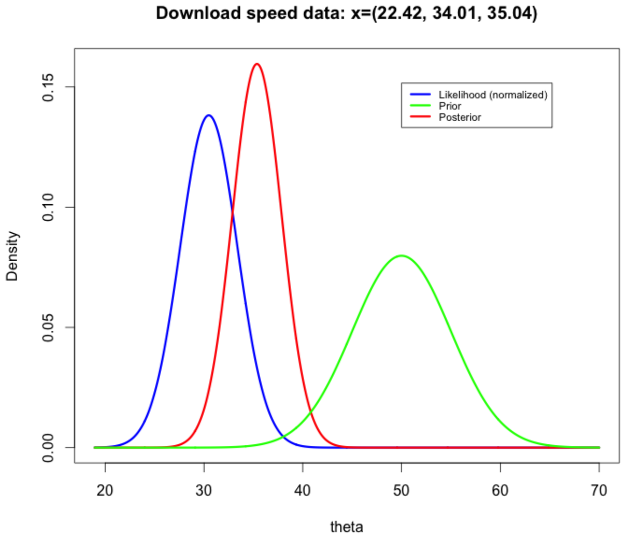
Download speed n=1



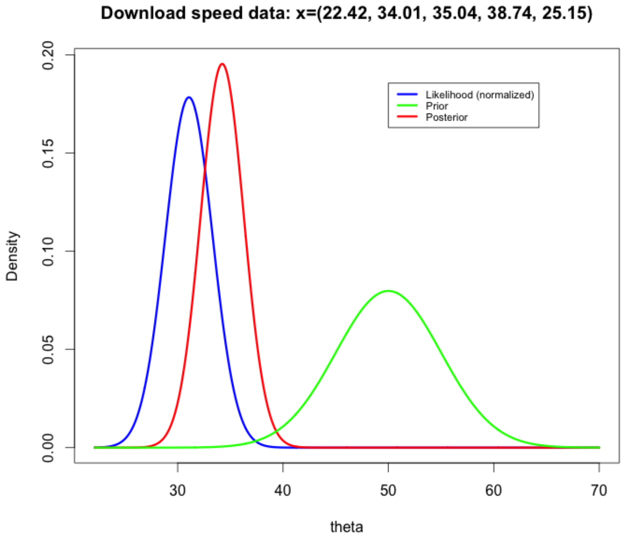
Download speed n=2



Download speed n=3



Download speed n=5



Linear regression

- The linear regression model in matrix form

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

- Usually first column of $\mathbf{X}$ is the unit vector and β_1 is the intercept.

- Normal errors: $\varepsilon_i \overset{iid}{\sim} N(0, \sigma^2)$, so $\varepsilon \sim N(0, \sigma^2 I_n)$.

- Likelihood

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

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:

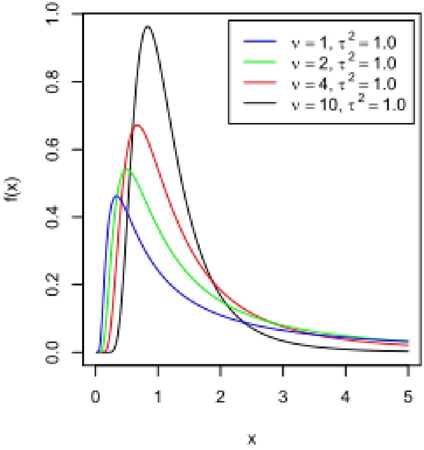
$$\begin{aligned} \beta|\sigma^2, \mathbf{y} &\sim N\left[\hat{\beta}, \sigma^2(\mathbf{X}'\mathbf{X})^{-1}\right] \\ \sigma^2|\mathbf{y} &\sim Inv\text{-}\chi^2(n-k, s^2) \end{aligned}$$

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

- Simulate from the joint posterior by simulating from
  - $p(\sigma^2|\mathbf{y})$
  - $p(\beta|\sigma^2, \mathbf{y})$
- Marginal posterior of β :

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

Scaled inverse χ^2 distribution



- **Inverse gamma** distribution.

Linear regression - conjugate prior

- **Joint prior** for β and σ^2

$$\beta|\sigma^2 \sim N\left(\mu_0, \sigma^2\Omega_0^{-1}\right)$$
$$\sigma^2 \sim Inv-\chi^2\left(\nu_0, \sigma_0^2\right)$$

- **Posterior**

$$\beta|\sigma^2, \mathbf{y} \sim N\left[\mu_n, \sigma^2\Omega_n^{-1}\right]$$
$$\sigma^2|\mathbf{y} \sim Inv-\chi^2\left(\nu_n, \sigma_n^2\right)$$

$$\begin{aligned}
\mu_n &= \left(\mathbf{X}'\mathbf{X} + \Omega_0\right)^{-1}\left(\mathbf{X}'\mathbf{X}\hat{\beta} + \Omega_0\mu_0\right) \\
\Omega_n &= \mathbf{X}'\mathbf{X} + \Omega_0 \\
\nu_n &= \nu_0 + n \\
\nu_n\sigma_n^2 &= \nu_0\sigma_0^2 + \left(\mathbf{y}'\mathbf{y} + \mu_0'\Omega_0\mu_0 - \mu_n'\Omega_n\mu_n\right)
\end{aligned}$$

Ridge regression = normal prior

- Problem: too many covariates leads to over-fitting.
- Smoothness/shrinkage/regularization **prior**

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

- Larger λ gives smoother fit. Note: $\Omega_0 = \lambda I$.
- Equivalent to penalized likelihood:

$$-2 \cdot \log p(\beta|\sigma^2, \mathbf{y}, \mathbf{X}) \propto (y - X\beta)^T (y - X\beta) + \lambda\beta'\beta$$

- Posterior mean gives ridge regression estimator

$$\tilde{\beta} = \left(\mathbf{X}'\mathbf{X} + \lambda I\right)^{-1} \mathbf{X}'\mathbf{y}$$

- Shrinkage toward zero

$$\text{As } \lambda \to \infty,\ \tilde{\beta} \to 0$$

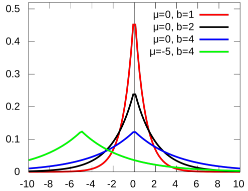
- When $\mathbf{X}'\mathbf{X} = I$

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

Lasso regression = Laplace prior

- **Lasso** is equivalent to posterior mode under Laplace prior

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



- **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.

Estimating the shrinkage

- Cross-validation is often used to determine the degree of smoothness, λ.
- Bayesian: λ is **unknown** $\Rightarrow$ **use a prior** for λ.
- $\lambda \sim \mathit{Inv}\text{-}\chi^2(\eta_0, \lambda_0)$. The user specifies η_0 and λ_0.
- Hierarchical setup:

$$
\begin{aligned}
\mathbf{y}|\beta, \mathbf{X} &\sim N(\mathbf{X}\beta, \sigma^2 I_n) \\
\beta|\sigma^2, \lambda &\sim N\left(0, \sigma^2\lambda^{-1} I_m\right) \\
\sigma^2 &\sim \mathit{Inv} - \chi^2(\nu_0, \sigma_0^2) \\
\lambda &\sim \mathit{Inv}\text{-}\chi^2(\eta_0, \lambda_0)
\end{aligned}
$$

Regression with estimated shrinkage

- The joint posterior of β, σ^2 and λ is

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

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

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

- $\Omega_0(\lambda) = \lambda I_m$, and $p(\lambda)$ is the prior for λ.

Polynomial regression

- **Polynomial regression**

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

$$\mathbf{y} = \mathbf{X}\beta + \varepsilon,$$

where

$$\mathbf{X} = (1, x, x^2, ..., x^k).$$

- Problem: higher order polynomials can overfit the data.

- Solution: shrink higher order coefficients harder:

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

Finding the time for maximum

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

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

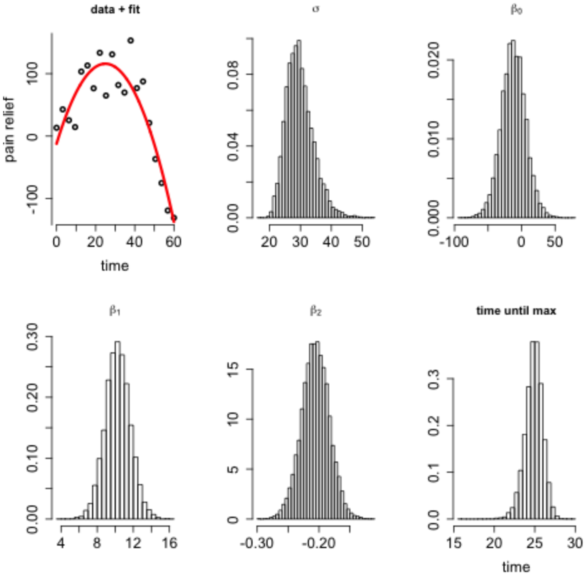
- At what time x_{max} is there **maximal pain relief**?

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

.

- Posterior distribution of x_{max} can be obtained by change of variable. Cauchy-like.
- Easy to obtain marginal posterior $p(x_{max}|\mathbf{y}, \mathbf{X})$ by **simulation**:
  - Simulate N coefficient vectors from the posterior $\beta, \sigma^2|\mathbf{y}, \mathbf{X}$
  - For each simulated β, compute $x_{max} = -\beta_1/2\beta_2$.
  - Plot a histogram. Converges to $p(x_{max}|\mathbf{y}, \mathbf{X})$ as $N \to \infty$.

Finding the time for maximum



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

- Ongoing research on making Bayes more scalable to large data. My own contributions: https://mattiasvillani.com/research

- Probabilistic programming languages (**Stan**) makes Bayes easy.

- Bayesian Learning course at SU: https://github.com/mattiasvillani/BayesLearnCourse