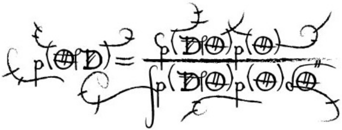


Bayesian Learning 732A46: Lecture 12

Matias Quiroz[1,2]

[1]Division of Statistics and Machine Learning, Linköping University

[2]Research Division, Sveriges Riksbank

May 2016

Lecture overview

- Hierarchical models
- MCMC with RStan

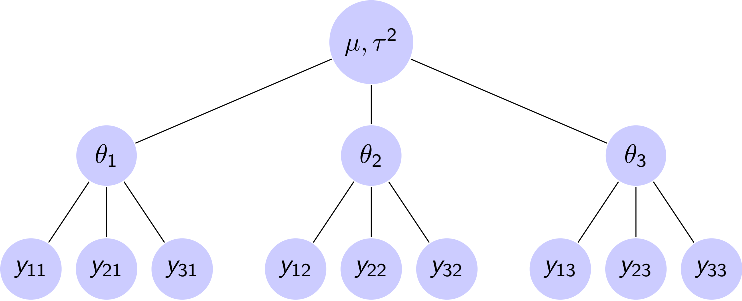
The normal Hierarchical model

- **The Bayesian hierarchical** normal model (μ and τ^2 are also **random**!)

$$\begin{aligned} y_{ij}|\theta_j, \sigma^2 &\sim \mathcal{N}(\theta_j, \sigma^2) \\ \theta_j|\mu, \tau^2 &\sim \mathcal{N}(\mu, \tau^2) \quad \text{and} \quad \sigma^2 \sim p(\sigma^2), \quad \mu, \tau^2 \sim p(\mu, \tau^2), \end{aligned}$$

  where $i = 1, \ldots, N$ (observations) and $j = 1, \ldots, J$ (groups). Let n_j be the **number of observations** in **group** j.

- **Example**: $N = 3$, $J = 3$ and σ^2 known



Some remarks on a hierarchical model

- **Note**: the (unconditional/marginal) prior for θ is

$$p(\theta) = p(\theta_1, \ldots, \theta_J) = \int \left(\prod_{j=1}^{N} p(\theta_j | \mu, \tau^2) \right) p(\mu, \tau^2) d\mu d\tau^2.$$

- $\theta_1, \ldots, \theta_J$ **are not** independent because

$$p(\theta_1, \theta_2, \ldots, \theta_J) \neq p(\theta_1)p(\theta_2) \cdots p(\theta_J),\ \text{but they are } \textbf{exchangeable}.$$

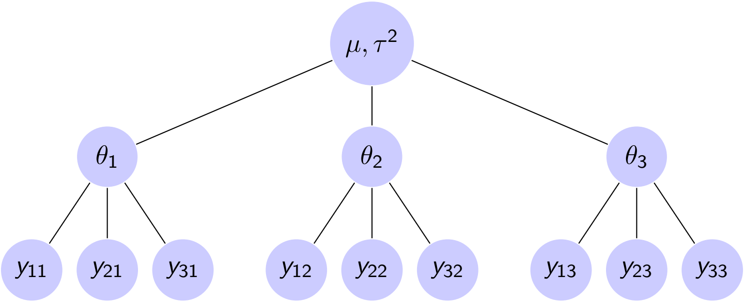
- **Exchangeable**: $p(\theta)$ invariant to **permutation of indices**. **Weaker** than independence.
- Hyper-parameters set to **sensible values** earlier. Modelling them now!

Bayesian core philosophy

Regard unknown quantities as **random variables** and **learn from data**.

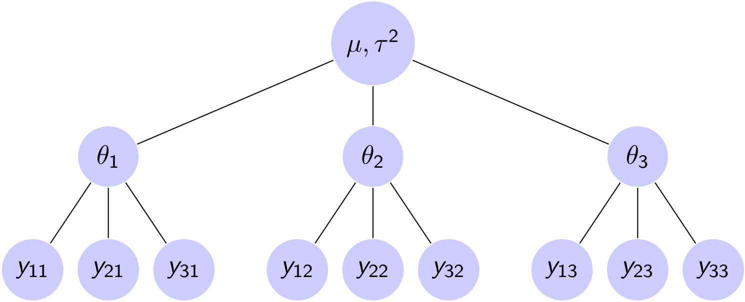
- Hierarchical models are **full probability models**... makes a Bayesian go ☺
- **Practical advantages**? Shrinkage (=pooling in hierarchical terminology).

The power of pooling (shrinking)



- If $\tau^2 \approx 0$ the θ_j's are close to each other ($\approx \mu$). The opposite for large τ^2.
- To estimate the θ_j's the Frequentist performs a **one-way ANOVA**.
- $H_0 =$ *The means are equal vs* $H_1 =$ *The means are not equal*. **F-test**.
- H_0: *All data to estimate the common mean.* $\neg H_0$: *Estimate each separately.*
- **Bayesian**: Why black or white? "To shrink completely or to not shrink at all"

The power of pooling (shrinking), cont.



- **The Bayesian way: The data** decides the amount of pooling: $p(\tau^2|y)$.
- **Extreme cases** give the Frequentist solution

$$H_0 : \tau^2 = 0 \qquad \text{[Shrink completely to } \mu\text{]}$$
$$H_1 : \tau^2 = \infty \qquad \text{[Don't shrink at all]}$$

- Groups with **few** y_{ij}'s: H_1 gives **high variance** on group mean estimates.
- **Pooling to the rescue**: the estimates **borrow strength** from each other by **sharing hyper-parameters** (estimated using **all data** y)

Estimation of the hierarchical normal model

- **Blocks** of parameters: $\theta = \left\{(\theta_1, \ldots, \theta_J), \sigma^2, \mu, \tau^2\right\}$.
- The **joint posterior**

$$
\begin{aligned}
\pi(\theta) &\propto p(y|\theta_1, \ldots, \theta_J, \sigma^2, \mu, \tau^2) p(\theta_1, \ldots, \theta_J, \sigma^2, \mu, \tau^2) \\
&= p(y|\theta_1, \ldots, \theta_J, \sigma^2) p(\theta_1, \ldots, \theta_J|\sigma^2, \mu, \tau^2) p(\sigma^2, \mu, \tau^2) \\
&= p(y|\theta_1, \ldots, \theta_J, \sigma^2) p(\theta_1, \ldots, \theta_J|\mu, \tau^2) p(\sigma^2, \mu, \tau^2) \\
&= \left(\prod_{j=1}^{J} \prod_{i=1}^{n_j} \mathcal{N}(y_{ij}|\theta_j, \sigma^2)\right) \left(\prod_{j=1}^{J} \mathcal{N}(\theta_j|\mu, \tau^2)\right) p(\sigma^2, \mu, \tau^2)
\end{aligned}
$$

is a **nightmare**... **But** assuming $p(\sigma^2, \mu, \tau^2) = \underbrace{p(\sigma^2)}_{\text{Inv-}\chi^2} \underbrace{p(\mu)}_{\mathcal{N}} \underbrace{p(\tau^2)}_{\text{Inv-}\chi^2}$

1. $\theta_j \,|\mathrm{rest}, y \sim \mathcal{N}$, $j = 1, \ldots, J$
2. $\sigma^2|\mathrm{rest}, y \sim \text{Inv-}\chi^2$
3. $\mu \,|\mathrm{rest}, y \sim \mathcal{N}$
4. $\tau^2|\mathrm{rest}, y \sim \text{Inv-}\chi^2$.

- **Gibbs sampling**!

More complex hierarchical models

- We are (of course) **not limited** to just 2 layers.
  - ***L*-layers with params $\gamma_1, \ldots \gamma_L$**: Just crank the **Bayesian machine**

$$p(\gamma_1, \ldots, \gamma_L | y) \propto p(y | \gamma_1, \ldots, \gamma_L) p(\gamma_1, \ldots, \gamma_L)$$

    and **factorize the prior** with the formula we have used more than 1000 times

$$p(\gamma_1, \ldots, \gamma_L) = p(\gamma_L |, \gamma_{L-1} \ldots, \gamma_2, \gamma_1) p(\gamma_2 | \gamma_1) p(\gamma_1).$$

  - Derive **full conditionals** $\gamma_l | \mathrm{rest}, y$ by choosing (if possible) a **conjugate prior**.
  - **Estimation**: **Gibbs sampling**. Is any γ_k of unknown form? **Metropolis-Hastings** within Gibbs (Lecture 9)!

- We are (of course) **not limited** to normal distributions for **the data** y.
  - We have **conjugate priors** for some other models...
  - ... and if we don't: **M-H within Gibbs** saves us!

More complex hierarchical models, cont.

- Can easily **be generalized** to a regression setting.
- Make γ_l a function of specific covariates in the lth layer. **Example**:

$$\gamma_l = g_l(x_l'\beta_l) \quad [g_l(x_l'\beta_l) = x_l'\beta_l \text{ if linear regression}]$$

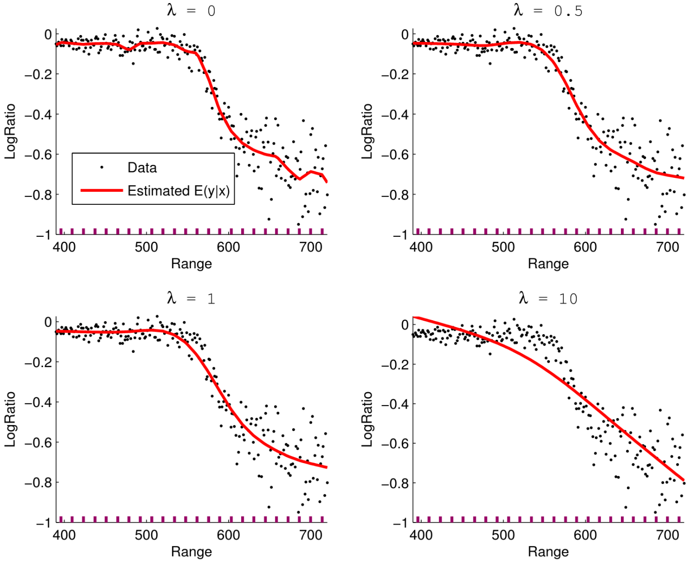
- **Estimate** β_l. If normal model: **Bayesian linear regression** updates.
- **Example**: **Analyzing performance of students**.

  Hierarchies: **students** within **classes** within **schools** within **states**.

  Data: 10 tests (y = score) for each student during a year. Possible x's
  - **Student**: Male/female, junior/senior, education of parents, etc.
  - **Class**: Years of experience of teacher, number of students in class, etc.
  - **School**: Private/public, measures on geographical level, e.g. crimes, unemployment, etc.
  - **State**: Welfare policies, e.g. investments in schools, social securities, etc.
- We are (of course) **not limited** to a univariate response.

  Example: For student i, y_i = (math score, english score, history score)

Revisiting regularization in the Bayesian spline model



Estimating the shrinkage parameter λ by direct sampling

- **Model**: $y|\beta, \sigma^2 \sim \mathcal{N}(X\beta, \sigma^2 I)$
- The **joint posterior** factorizes

$$p(\beta, \sigma^2, \lambda|y) = p(\beta|\sigma^2, \lambda, y)p(\sigma^2|\lambda, y)p(\lambda|y),$$

where

$$\begin{array}{rcl} \textbf{Prior} & \rightarrow & \textbf{Posterior} \\ \beta|\sigma^2, \lambda \sim \mathcal{N}(0, \sigma^2\Omega_0^{-1}) & \rightarrow & \beta|\sigma^2, \lambda, y \sim \mathcal{N}(\beta_n, \sigma^2\Omega_n^{-1}) \\ \sigma^2 \sim \text{Inv-}\chi^2(\nu_0, s_0^2) & \rightarrow & \sigma^2|\lambda, y \sim \text{Inv-}\chi^2(\nu_n, s_n^2) \\ \lambda \sim p(\lambda) & \rightarrow & \lambda|y \sim \sqrt{\frac{|\Omega_0|}{|\Omega_n|}} \left(\frac{\nu_n s_n^2}{2}\right)^{-\nu_n/2} p(\lambda) \end{array}$$

and

$$\begin{array}{ll} \beta_n = (X'X + \Omega_0)^{-1}X'y & \Omega_n = X'X + \Omega_0 \\ \nu_n = \nu_0 + n & \nu_n s_n^2 = \nu_0 s_0^2 + y'y - \beta_n'\Omega_n\beta_n \end{array}$$

- **Note**: β and σ^2 dependent apriori to **achieve conjugacy**.

Alternatively: make your life easy by a hierarchical setup

- **Model**:

$$y|\beta, \sigma^2 \sim \mathcal{N}(X\beta, \sigma^2 I)$$
$$\beta|\lambda \sim \mathcal{N}(0, \Omega_0^{-1}) \quad \text{and } \sigma^2 \sim p(\sigma^2), \lambda \sim p(\lambda)$$

  and take (for example) $\Omega_0 = \lambda I$.

- **Draw** the **hierarchical structure** (white board)!

- Assuming $p(\sigma^2) = \text{Inv-}\chi^2$ and $p(\lambda) = \text{Inv-}\chi^2$ [**semi-conjugate prior**]
  1. $\beta\ |\mathrm{rest}, y \sim \mathcal{N}$
  2. $\sigma^2|\mathrm{rest}, y \sim \text{Inv-}\chi^2$
  3. $\lambda\ |\mathrm{rest}, y \sim \text{Inv-}\chi^2$.

- **That's easy**!...

- ... **But**: recall that Gibbs is **never as efficient** as **direct sampling**.

RStan - a short demonstration

- Why **Stan** (mc-stan.org)
  - **Easy** to install (see here).
  - **Easy** to use.
  - **Efficient** MCMC. **Hamiltonian Monte Carlo**.
  - Integrates nice with **RStudio**.
  - Wrappers from **Python**, **R**, **Matlab**, **Stata**, **Julia**.
  - Good documentation.
- Alternatives to **Stan** (**Stan**islaw Ulam)
  - Do it yourself ☺
  - **BUGS** (**B**ayesian inference **U**sing **G**ibbs **S**ampling)
  - **JAGS** (**J**ust **A**nother **G**ibbs **S**ampler)
- More examples found on the **course web page** and the **GitHub-repo**...
- ... and using your friend **Google**.

The parts of a model in Stan

- **Six parts** in a **Stan** model:
  - `data`
  - `transformed data`
  - `parameters`
  - `transformed parameters`
  - `model`
  - `generated quantities`

Example: Poisson regression

- **Poisson regression** for the **Number of roaches caught in buildings**.
- **Covariates**
  - Exposure
  - Treatment (yes/no)
  - Senior building (yes/no).
- **Non-conjugate** model.
- **Model**:

$$
\begin{aligned}
y_i|\beta &\sim \text{Poisson}(\lambda_i) \\
\log(\lambda_i) &= \log(\text{exposure}_i) + \beta_1 + \beta_2 \cdot \text{treatment}_i + \beta_3 \cdot \text{senior}_i \\
\beta &\sim \mathcal{N}(0, 1000)
\end{aligned}
$$

Model in Stan: `data`

- Read in **data** (done once)
  - Variable declarations
  - A lot of different data types, e.g. `int`, `real`, `vector`, `matrix`.

Example: **`Data block`**

```
data {
   int<lower=0> N; # The number of observations
   int<lower=0> y;
   vector[N] exposure2;
   vector[N] senior;
   vector[N] treatment;
}
```

Model in Stan: `transformed data`

- **Variable declarations** and **statements** (done once)
- See Chapter V in the documentation for all functions that can be used.

Example: **`Transformed data block`**

```
transformed data {
   vector[N] log_expo;
   log_expo <- log(exposure2);
}
```

Model in Stan: parameters

- **Parameters** that should be sampled.
- **Parameter declarations** only.

Example: **Parameters block**

```
parameters {
   vector[3] beta;
}
```

Model in Stan: `transformed parameters`

- **Note**: Make sure that you know **which parametrization is used**. **One of the best advices I can ever give you**.
- **Parameter declarations** and **statements**.

Example: **`Transformed parameter block`** [not our example]

```
transformed parameters {
    real<lower=0> sigma;
    sigma <- 1.0 / sqrt(tau);
}
```

- Declare the **priors** and **model for data** with "sampling statement" symbol $\sim$.
- **Distributions** can be found in Chapter VI and VII in the documentation.
- **Again**: Make sure that you know **which parametrization is used**.

Example: Model block

```
model {
   # Priors
   beta ~ normal(0.0, 1000.0);
   # Model
   y ~ poisson_log(log_expo + beta[1] +
   beta[2] * treatment + beta[3]*senior);
}
```

Model in Stan: `generated quantities`

- Post sampling computations. **Examples**:
  - Model checking
  - Posterior predictive distribution
  - Applying full Bayesian decision theory
  - Transforming parameters for reporting.

Example: **`Generated quantities block`**

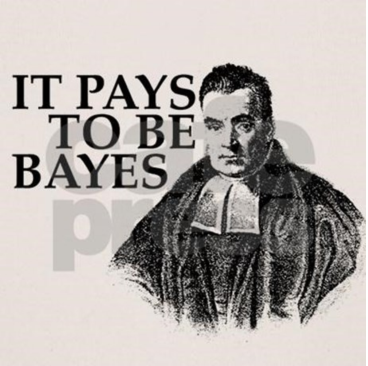
```
generated quantities {
   int<lower=0> pred_treat;
   int<lower=0> pred_notreat;
   vector[3] exp_beta;

   exp_beta <- exp(beta);
   pred_treat <- poisson_rng(exp_beta[1]*exp_beta[2]);
   pred_notreat <- poisson_rng(exp_beta[1]);
}
```

Demonstration

```
Demonstration on my computer
```

This is the End...



- .. **of my lectures**..
- ... **but the Beginning** of **your new life as a Bayesian**.
- **Thank you**, it has been a pleasure.