Bayesian Linear Regression

Felipe José Bravo Márquez

July 12, 2021

Bayesian Linear Regression

- In this class, which is mostly based on chapter 4 of [McElreath, 2020], we are going to revisit the linear regression model from a Bayesian point of view.
- The idea is the same: to model the relationship of a numerical dependent variable $\mathbf{y}$ with n independent variables $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$ from a dataset d.
- The response vaiable $\mathbf{y}$ is again modeled with a Gaussian distribution: $y_i \sim N(\mu_i, \sigma^2)$.
- We also mantain the assumption that each attribute has a linear relationship to the mean of the outcome.

$$\mu_i = \beta_0 + \beta_1 x_i + \ldots \beta_n x_n$$

- However, we are not going to use least squares or maximum likelihood to obtain point estimates of the parameters.
- Instead, we are going to estimate the joint posterior distribution of all the parameters of the model:

$$f(\theta|d) = f(\beta_0, \beta_1, \ldots, \beta_n, \sigma|d)$$

Bayesian Linear Models

- The Bayesian linear regresion is more flexible than least squares as it allows incorporating prior information.
- It also allows to interpret the uncertainty of the model in a clearer way.
- Notice that the the parameters of the model are $\beta_0, \beta_1, \ldots, \beta_b$ and σ but not μ_i.
- This is because μ_i it is determined deterministically from the linear model's coefficients.
- In order to build our posterior we need to define a likelihood function:

$$f(d|\beta_0, \beta_1, \cdots, \beta_n, \sigma) = \prod_{i=1}^{m} f(d_i|\beta_0, \beta_1, \cdots, \beta_n, \sigma)$$

- Where d_i corresponds to each data point in the dataset containing values for y and $x_1, \ldots, x_n$ (IID assumption).
- The likelihood of each point is modeled with a Gaussian distribution:

$$f(d_i|\beta_0, \beta_1, \cdots, \beta_n, \sigma) = N(\mu_i, \sigma^2)$$

Bayesian Linear Models

- Now we need a joint prior density:

$$f(\theta) = f(\beta_0, \beta_1, \ldots, \beta_n, \sigma)$$

- And the posterior gets specified as follows:

$$f(\theta|d) = \frac{\prod_{i=1}^{m} f(d_i|\beta_0, \beta_1, \cdots, \beta_n, \sigma) * f(\beta_0, \beta_1, \ldots, \beta_n, \sigma)}{f(d)}$$

- The evidence is expressed by a multiple integral:

$$f(d) = \int \int \cdots \int \prod_{i=1}^{m} f(d_i|\beta_0, \beta_1, \cdots, \beta_n, \sigma) * f(\beta_0, \beta_1, \ldots, \beta_n, \sigma) d\beta_0 d\beta_1 \cdots d\beta_n d\sigma$$

- In most cases, the priors are specified independently for each parameter, which is equivalent to assuming:

$$f(\beta_0, \beta_1, \cdots, \beta_b, \sigma) = f(\beta_0) * f(\beta_1) * \cdots * f(\beta_n) * f(\sigma).$$

A model of height revisited

- To understand this more concretely, we will rebuild the linear model relating the height and weight of the !Kung San people using a Bayesian approach.
- We will refer to each person's height and weight as y_i and x_i respectively.
- Our probabilistic model specifying all components of a Bayesian model is defined as follows:

$$\begin{aligned}
y_i &\sim N(\mu_i, \sigma) && \text{[likelihood]} \\
\mu_i &= \beta_0 + \beta_1 x_i && \text{[linear model]} \\
\beta_0 &\sim N(150, 50) && [\beta_0 \text{ prior}] \\
\beta_1 &\sim N(0, 1) && [\beta_1 \text{ prior}] \\
\sigma &\sim \text{Uniform}(0, 50) && [\sigma \text{ prior}]
\end{aligned}$$

- Parameters β_0 and β_1 are the intercept and the slope of our linear model.
- The parameter σ is the standard deviation of all the heights.
- Note that we are setting the same σ for all observations, which is equivalent to the Homoscedasticity property of the standard linear regression.

A model of height revisited

- Our priors were set independently for each parameter which implies that the joint prior density $f(\beta_0, \beta_1, \sigma)$ can be expressed as $f(\beta_0) * f(\beta_1) * f(\sigma)$.
- It should be kept in mind that the choice of priors is subjective and should be evaluated accordingly.
- Let's try to justify our choice a bit:
  1. The Gaussian prior for β_0 (intercept), centered on 150cm with a standard variation of 50, covers a huge range of plausible mean heights for human populations while giving very little chance for negative heights.
  2. The Gaussian prior for β_1 (slope), centered on 0 with a standard variation of 1, acts as a **regularizer** to prevent the model from **overfitting** the data by assigning extreme values to β_1.[1]
  3. The uniform prior for the standard deviation σ between 0 and 50 prohibits obtaining negative standard deviations. The upper bound (50 cm) would imply that 95% of individual heights lie within 100cm of the average height. That's a very large range.

[1] Regularization and overfitting will be discussed later in the course.

Fitting the Model

- Now we need to fit the model to the data to build the posterior distribution.
- Grid approximation is not a valid option, as setting up a grid for 3 parameters would be too computationally expensive.
- We will use Laplace approximation instead.
- In this approach we obtain the MAP estimates for each parameter using a hill-climbing **optimization** method.
- Then we fit a **multivariate Gaussian distribution** centered on these values.
- This distribution is the multidimensional extension to the standard Gaussian.

The multivariate Gaussian distribution

- The multivariate Gaussian distribution in d-dimensions defined by the following density function (PDF):

$$f_x = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(\vec{x}-\vec{\mu})^T\Sigma^{-1}(\vec{x}-\mu)\right)$$

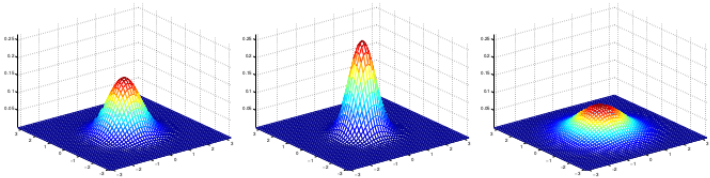
- This density function allows working with a d-dimensional vector of random variables $\vec{X}$.
- The first parameter of this distributions is a mean vector $\vec{\mu} \in \mathcal{R}^d$ with the mean value of each dimension.
- The second parameter is a covariance matrix $\Sigma \in R^{d \times d}$,
- This matrix contains the variance of each variable in the diagonal and the covariance of variables X_i and X_j in the other cells $\Sigma_{i,j}$:

$$Cov(X) = \Sigma$$

- The matrix Σ is symmetric and positive semi-definite.
- The multivariate Gaussian $N(\vec{\mu}, \Sigma)$ is a very convenient distribution for modeling multidimensional random variables.

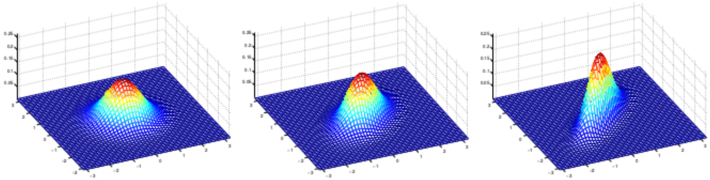
The multivariate Gaussian distribution

- Here are some examples taken from [Ng, 2008] of what the density of a multivariate Gaussian distribution looks like:



- The left-most figure shows a Gaussian with mean zero (that is, the 2x1 zero-vector) and covariance matrix $\Sigma = I$ (the 2×2 identity matrix).
- A Gaussian with zero mean and identity covariance is also called the standard normal distribution.
- The middle figure shows the density of a Gaussian with zero mean and $\Sigma = 0.6I$.
- The rightmost figure shows one with , $\Sigma = 2I$.
- We see that as Σ becomes larger, the Gaussian becomes more “spread-out”, and as it becomes smaller, the distribution becomes more “compressed”.

The multivariate Gaussian distribution

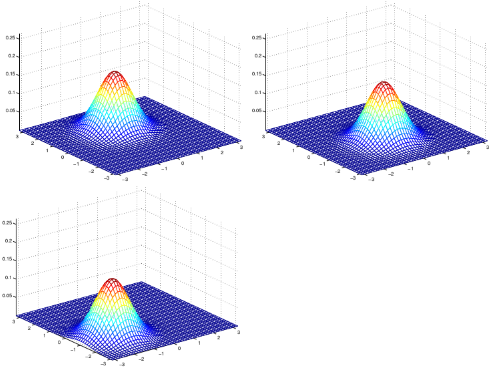


- The figures above show Gaussians with mean 0, and with covariance matrices respectively

$$\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}; \; \Sigma = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}; \; \Sigma = \begin{bmatrix} 1 & 0.8 \\ 0.8 & 1 \end{bmatrix}.$$

- The leftmost figure shows the familiar standard normal distribution, and we see that as we increase the off-diagonal entry in Σ, the density becomes more "compressed" towards the 45· line (given by $x_1 = x_2$).

- As our last set of examples, fixing $\Sigma = I$, by varying $\vec{\mu}$ we can also move the mean of the density around.



- The figures above were generated using $\Sigma = I$, and respectively

$$\mu = \begin{bmatrix} 1 \\ 0 \end{bmatrix}; \;\; \mu = \begin{bmatrix} \text{-}0.5 \\ 0 \end{bmatrix}; \;\; \mu = \begin{bmatrix} \text{-}1 \\ \text{-}1.5 \end{bmatrix}.$$

Laplace approximation

- In Laplace approximation we assume that the joint posterior follows a multivariate Gaussian distribution $f(\theta_1, \ldots, \theta_n) = N(\vec{\mu}, \Sigma)$.
- According to [Gelman et al., 2013], this approximation is convenient for unimodal and roughly symmetric posterior distributions.
- There are also a theoretical asymptotic argument in favor of this approximation: "if the dataset is large enough, a posterior distribution can be approximated by a Gaussian" [Gelman et al., 2013].
- The values of $\vec{\mu}$ are obtained from the posterior mode of each parameter (MAP):

$$\vec{\mu} = \vec{\theta}_{MAP}$$

- Which are obtained using numeric optimization techniques.

Laplace approximation

- The values of Σ are obtained from the curvature near these values, which are obtained from the second derivatives of the posterior:

$$\Sigma = [I(\theta_{MAP})]^{-1}$$

where

$$I(\theta) = -\frac{d^2}{d\theta^2} \log f(\theta|d)$$

- Notice that both $\vec{\mu}$ and Σ can be calculated from the unnormalized posterior $f(d|\theta) * f(\theta)$
- We don't need to calculate the evidence $f(d)$ to perform Laplace approximation [**?**].

Fitting the Model

- Laplace approximation is implemented in the **quap** function from the **rethinking** package.
- The model for height defined above can be implemented as follows:

```
library(rethinking)
data(Howell1)
d <- Howell1
d2 <- d[ d$age >= 18 , ]

b.reg1 <- quap(
  alist(
    height ~ dnorm( b0 + b1*weight, sigma ),
    b0 ~ dnorm( 150 , 50 ) ,
    b1 ~ dnorm( 0 , 1) ,
    sigma ~ dunif( 0 , 50 )
  ) , data=d2 )
```

Fitting the Model

- We can summarize this posterior with the command **precis**:

```
> precis( b.reg1, prob=0.95 )
        mean   sd   2.5%  97.5%
b0    113.99 1.90 110.26 117.71
b1      0.90 0.04   0.82   0.98
sigma   5.07 0.19   4.70   5.45
```

- These numbers provide Gaussian approximations for each parameter's **marginal** posterior distribution.
- The marginal of multivariate distribution is the univariate distribution of a single parameter θ_i after integrating (averaging) over all the other parameters θ_j $\forall j \neq i$:

$$f(\theta_i|d) = \int \cdots \int f(\theta_1, \ldots, \theta_n|d) d\theta_1, \ldots, d\theta_{i-1}, d\theta_{i+1}, \ldots, d\theta_n$$

Fitting the Model

- In a multivariate Gaussian $N(\vec{\mu}, \Sigma)$, the marginal distribution of θ_i is a univariate Gaussian $N(\vec{\mu}_i, \Sigma_{ii})$.
- So, the marginal distribution of β_1 is a Gaussian distribution with $\mu = 0.9$ and $\sigma = 1.9$.
- The last two columns of the summary table show the lower and upper limits of a 95% equal-tailed credible interval for each parameter.
- For the case of β_1, 95% of the posterior probability lies between 0.82 and 0.98.
- That suggests that β_1 values close to zero or greatly above one are highly incompatible with these data and this model.

Fitting the Model

- Let's compare the mean and standard deviation of β_0 and β_1 with what we obtained using least squares in previous lecture:

```
> summary(reg1)
...

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 113.87939    1.91107   59.59   <2e-16 ***
weight        0.90503    0.04205   21.52   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5.086 on 350 degrees of freedom
....
```

- These values are almost the same!
- Recall that maximum likelihood or least squares estimators are identical to MAP estimators with uniform priors.
- This indicates that our priors did not have a significant impact in the resulting posterior.

Covariance Matrix

- We can also get the covariance matrix Σ of our multivariate Gaussian obtained with Laplace approximation:

```
> vcov( b.reg1 )
                b0            b1         sigma
b0     3.620141920 -7.884254e-02  1.765417e-03
b1    -0.078842537  1.752477e-03 -3.830179e-05
sigma  0.001765417 -3.830179e-05  3.654305e-02
```

- The matrix tell us how each parameter relates to every other parameter in the posterior distribution.
- It is better to turn them into correlations for easier interpretation:

```
> cov2cor( vcov( b.reg1 ) )
                b0           b1        sigma
b0     1.000000000 -0.989855066  0.004853802
b1    -0.989855066  1.000000000 -0.004786196
sigma  0.004853802 -0.004786196  1.000000000
```

- Notice that β_0 and β_1 are al- most perfectly negatively correlated.
- This means that these two parameters carry the same information: as we change the slope of the line, the best intercept changes to match it.

Centering

- In more complex models, strong correlations like this can make it difficult to fit the model to the data.
- A useful trick to avoid this problem is centering: subtract the mean of a variable from each value.

```
d2$weight.c <- d2$weight - mean(d2$weight)
b.reg2 <- quap(
  alist(
    height ~ dnorm( b0 + b1*weight.c, sigma ),
    b0 ~ dnorm( 150 , 50 ) ,
    b1 ~ dnorm( 0 , 1) ,
    sigma ~ dunif( 0 , 50 )
  ) , data=d2 )

> cov2cor( vcov( b.reg2 ) )
                 b0            b1         sigma
b0     1.000000e+00  6.901851e-08 -2.435168e-05
b1     6.901851e-08  1.000000e+00 -2.860675e-03
sigma -2.435168e-05 -2.860675e-03  1.000000e+00
```

- The correlations among parameters are now all very close to zero.

Sampling from the Posterior

- In the remainder of this class we will work with samples from our multi-dimensional posterior.
- The rethinking package provides the **extract.samples** function to obtain them:

```
> post <- extract.samples( b.reg1, n= 1e4 )
> head(post)
        b0        b1    sigma
1 112.6949 0.9241772 5.379502
2 114.0312 0.9049003 5.213418
3 114.5932 0.8867978 5.057051
4 111.9088 0.9522044 5.021026
5 111.7631 0.9496914 4.886909
6 116.6987 0.8367835 5.256882
```

- Each sample is a different line relating height and weight.
- Bear in mind that each line is sampled in proportion to the posterior probabilities, which represent our model's beliefs.
- Since our posterior was approximated with Laplace approximation we could get equivalent samples from a multi-variate Gaussian:

```
> library(MASS)
> post2 <- mvrnorm(n=1e4 , mu=coef(b.reg1),
Sigma=vcov(b.reg1))
```

Conclusions

- Blabla

References I

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., and Rubin, D. B. (2013).
Bayesian data analysis.
CRC press.

McElreath, R. (2020).
Statistical rethinking: A Bayesian course with examples in R and Stan.
CRC press.

Ng, A. (2008).
Generative learning algorithms.
Stanford Lecture Notes, 5(4).