Predicting Automobile Fuel Efficiency: A Bayesian Inference Approach

Cole Sturza   Alex Book

May 2, 2021

The Problem

- Build a model to predict fuel efficiency of a vehicle using Bayesian inference
- Why is this useful?
    - Allows for less manual analysis of vehicle design (and thus less prototyping and testing)
    - Allows for easier business expense analysis

The Model

- MPG is a count per unit (i.e miles per gallon)
- Poisson distributions model these types of variables well
- Assumptions:
  - The predicted variable is a count per unit of time or space described by a Poisson distribution
  - The observations must be independent of each other
  - The mean and variance of a Poisson random variable must be equal
  - The log of the mean rate, $\log(\lambda)$, must be linear with respect to our feature variables x_i

$$f(k;\lambda) = \Pr(X = k) = \frac{\lambda^k e^{-\lambda}}{k!}$$

The Data

- Seven usable fields
  - Number of engine cylinders (discrete)
  - Displacement (summative volume of vehicle's cylinders; continuous)
  - Engine horsepower (continuous)
  - Vehicle weight (continuous)
  - Acceleration (continuous)
  - Model year (discrete)
  - Origin of the car (US, Europe, Japan; label-encoded, discrete)
- Predicting MPG rating (discrete, necessary for Poisson distribution)

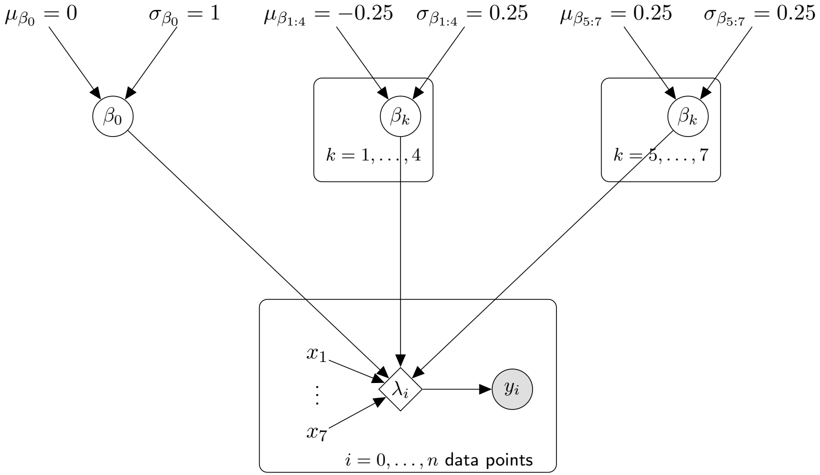
Poisson Regression

$$\log(\lambda_i) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_7 x_7$$

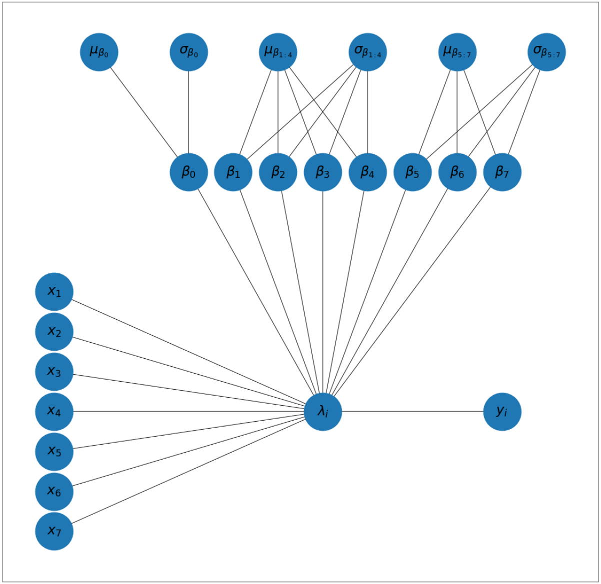
$$y_i \sim \text{ Poisson}(\lambda = \lambda_i)$$

$$\beta_k \sim \mathcal{N}(\mu_{\beta_k}, \sigma^2_{\beta_k}) \quad \text{for } k = 0, ..., 7$$

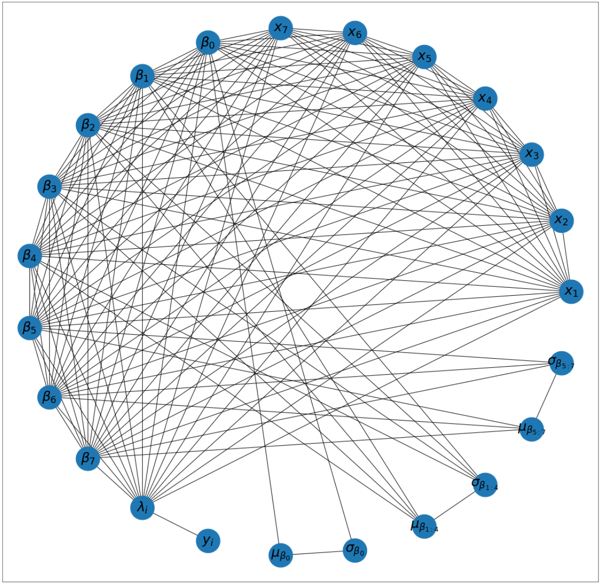
Belief Network (Plate Diagram)



Markov Network and Skeleton



Moralized and Triangulated Network



Joint Prior and Likelihood

Joint Prior:

$$\pi_0(\beta_k) \propto \frac{1}{\sigma^2_{\beta_k}\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{\beta_k - \mu_{\beta_k}}{\sigma^2_{\beta_k}}\right)^2}$$

$$\pi_0(\beta) \propto \prod_{k=0}^{7} \pi_0(\beta_k)$$

Likelihood:

$$\pi(y \mid \beta) \propto \prod_{i=0}^{n} \frac{\lambda_i^{y_i} e^{-\lambda_i}}{y_i!}$$

$$\lambda_i = \exp(\beta_0 + \beta_1 x_1^i + \beta_2 x_2^i + \cdots + \beta_7 x_7^i)$$

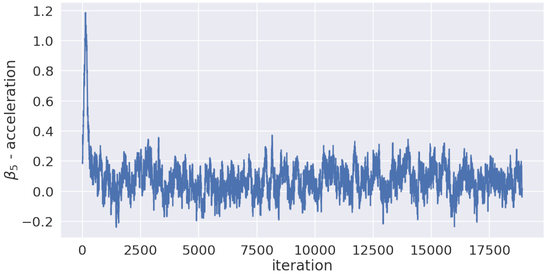
$$\pi(y \mid \beta) \propto \prod_{i=0}^{n} \frac{\exp(\beta_0 + \beta_1 x_1^i + \cdots + \beta_7 x_7^i)^{y_i}}{\exp(\exp(\beta_0 + \beta_1 x_1^i + \cdots + \beta_7 x_7^i))y_i!}$$

Metropolis-Hastings Algorithm (MCMC)

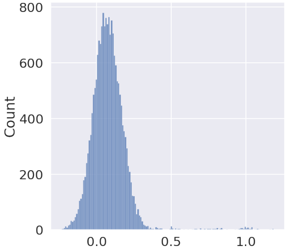
- This algorithm depends on the posterior being correctly proportional to the real posterior
- The assumptions we made earlier may be satisfied given that MPG is truly a Poisson distribution
  - mean = variance
- Possible improvements:
  - Negative binomial likelihood in place of the Poisson to capture dispersion
  - Quasi-likelihood

Metropolis-Hastings Algorithm (MCMC)

- 20/80 test-train split
- Ran for $100,000$ iterations with a burn in of $25,000$.
- Acceptance ratio of 23.317%

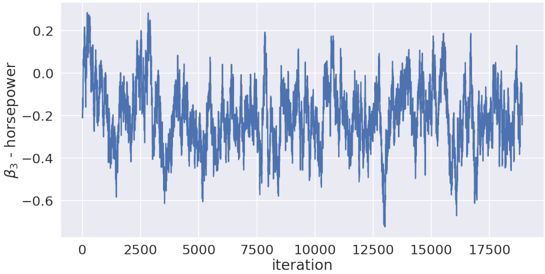


(a) Trace of acceleration parameter β_5.

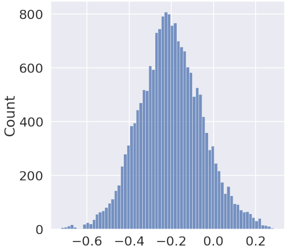


(b) Marginalization of acceleration parameter β_5.

Metropolis-Hastings Algorithm (MCMC)



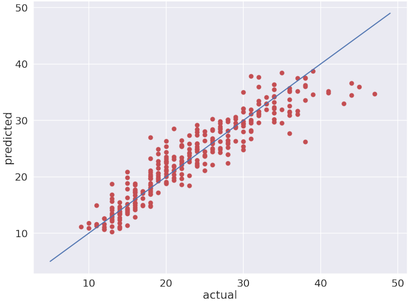
(a) Trace of horsepower parameter β_3.



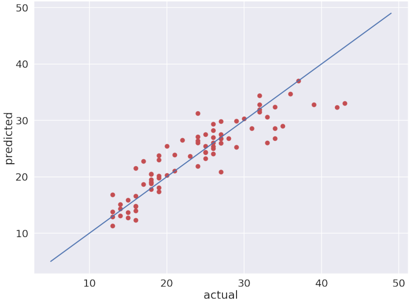
(b) Marginalization of horsepower parameter β_3.

Results

- Train: $R^2 = 0.86758, \text{ MSE} = 8.327$
- Test: $R^2 = 0.806, \text{ MSE} = 10.34$



(a) Error in train set predictions.



(b) Error in test set predictions.

Figure 3: The error in both the train and test sets, points closer the $x = y$ line are more accurate.