University of Colorado Boulder

CSCI 5822
Probabilistic Models of Human and Machine Intelligence
Final Project

Predicting Automobile Fuel Efficiency: A Bayesian Inference Approach

Cole Sturza
Alex Book

May 3, 2021

1 Introduction and Background

The fuel efficiency of a car is typically tested manually by car manufacturers using a machine called a dynamometer. The "dyno" simulates the driving environment similar to how a stationary exercise bike simulates cycling. This test requires engineers to adjust the settings of the rollers to account for wind resistance and the vehicle's weight. When estimating the miles per gallon (MPG) rating of a car on the "dyno," a driver runs the vehicle through a series of standardized driving routines to simulate typical trips in the city or on the highway. For vehicles that use carbon-based fuels, a hose is connected to the exhaust pipe that collects the engine exhausts during the various tests. The carbon is then measured by a machine to calculate the amount of fuel burned during the test. This process can be rather lengthy as it requires the manufacturer to build a prototype car, then perform the test. A possible solution to make this process quicker and easier for car manufacturers is the introduction of a model that allows particular feature inputs of the car being designed, yielding an estimated MPG rating before physically building a prototype. This could eliminate potential iterations on a prototype car to achieve a desired MPG rating. Another such area in which a model like this could be useful is businesses that offer catering services and need to pay delivery drivers. The business could use a model to get an estimate on the MPG of certain cars and adjust payment of their drivers to compensate for trips. Yet another possible application exists in companies that repay their employees for gas on their commute to work. They may find a model that can predict the MPG rating of a car useful to predict their monthly spending.

2 Project Aims

Our project aims to take a Bayesian Inference approach to creating a model to predict the MPG of a car given some set parameters about said car. We will infer a prior distribution for the parameters, as well as a likelihood for the model as a whole. We will then use Markov Chain Monte Carlo (MCMC) to approximate the values for the weights of each of these parameters. More specifically, we will use the Metropolis-Hastings algorithm defined in [1]. We will use the "Auto MPG Data Set" dataset from [2] to build our deterministic model for predicting MPG.

3 Approach/Methods

After analyzing the dataset we determined there were seven usable features. The number of cylinders in the engine, the displacement (the measure of the summative volume of the cylinders in a car's engine, excluding the combustion chambers), the horsepower of the engine, the weight of the car, the acceleration of the car, the model year, and the origin of the car (US, Europe, or Japan). The origin feature is label encoded as $1, 2, 3$ corresponding to each region. The predictor variable is the MPG rating of a car; this will be the observed variable in our model.

The number of cylinders, model year, and origin are discrete random variables; the displacement, horsepower, weight, and acceleration are continuous random variables. For the purposes of this model we will treat MPG as a discrete random variable. We convert all the values in the MPG column to integers by rounding to the nearest ones place. We also Min-Max scaled all of the feature variables—this was done to avoid overflows when learning each of the parameters in our model. We computed Min-Max scaling via the following equation:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

Where x is the original value of one of our features with all the recorded values for each car as a column vector, and x' is the normalized value. This moves all the values between the range $[0, 1]$.

Much of our inspiration behind converting MPG to a discrete value lies in our desire to model it with a Poisson distribution. This is necessary because Poisson distributions are used to measure a predicted variable that is some count per some unit of time or space [3]. MPG is the count of miles per unit volume gallon. Poisson distributions require discrete values since there is a factorial in the denominator. Another reason for this approach is that MPG cannot be negative, so using typical linear regression and assuming our predicted variable is normal does not make much sense. This is because a line is certain to yield negative values for some input of features. The Poisson model will keep our predicted variable between $[0, \infty)$. The Poisson distribution is defined for a random variable X as follows:

$$f(k; \lambda) = \Pr(X = k) = \frac{\lambda^k e^{-\lambda}}{k!} \; . \tag{1}$$

To build a regression model to predict the MPG of a car we will use Poisson regression, sometimes referred to as log-linear regression. The model is defined as:

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

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

Where y_i is our predicted variable MPG, and x_1 through x_7 are each of our feature variables. Each of the β's are learned parameters of our model. We will assume they all follow a normal distribution with mean μ_{β_k} and variance $\sigma^2_{\beta_k}$.

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

For cylinders, displacement, horsepower, and weight we assume that $\mu_{\beta_k} = -0.25$ and $\sigma^2_{\beta_k} = 0.0625$. For acceleration, model year, and origin we assume that $\mu_{\beta_k} = 0.25$ and $\sigma^2_{\beta_k} = 0.0625$. This is because when we plot each of the feature variables against the predicted variable we either observe a negative or positive linear relationship. All of the slopes of these relationships were observed to be around 0.25 or -0.25. This also seems realistic because as the number of cylinders goes up, MPG will generally go down (likewise for displacement, horsepower, and weight). Also, newer cars will generally have higher MPG, and faster acceleration helps increase MPG too. For origin this ended up working out nicely by happenstance for the encoded values. If we logically think about this though, American-made cars like trucks and muscle cars are gas-guzzlers compared to cars from Japanese manufacturers such as Subaru. For the bias term we chose a generic weakly informative prior ($\mu_{\beta_k} = 0$ and $\sigma^2_{\beta_k} = 1$) because we were not certain about the distribution. The prior for each β is represented as:

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

For numerical stability we took the log of the prior so that we wouldn't have to multiply a bunch of small probabilities together. The log prior for each β is the following:

$$\ln \pi_0(\beta_k) \propto -\ln(\sigma_{\beta_k}\sqrt{2\pi}) - (\beta_k - \mu_{\beta_k})^2/(2\sigma^2_{\beta_k}) \tag{3}$$

We made what may be a naive assumption that the feature variables are independent from one another. There may be some correlation between variables such as horsepower and acceleration, but enough outliers are present to justify assuming independence (some cars may be designed with top speed in mind more so than acceleration, leading to skews in the relationship). Therefore, the joint prior and joint log prior are:

$$\pi_0(\beta) \propto \prod_{k=0}^{7} \pi_0(\beta_k) \qquad \ln \pi_0(\beta) \propto \sum_{k=0}^{7} \ln \pi_0(\beta_k) \tag{4}$$

For the likelihood function, our model follows a Poisson distribution, therefore, our likelihood should be derived from (1). Our likelihood is the following:

$$\pi(y \mid \beta) \propto \prod_{i=0}^{n} \frac{\lambda_i^{y_i} e^{-\lambda_i}}{y_i!} \tag{5}$$

Where $\lambda_i = \exp(\beta_0 + \beta_1 x_1^i + \beta_2 x_2^i + \cdots + \beta_7 x_7^i)$. Let the superscript i denote the ith feature element used in the prediction λ_i, β represent all of our learned parameters, and n represent the number of observations. Substituting λ_i in we get:

$$\pi(y \mid \beta) \propto \prod_{i=0}^{n} \frac{\exp(\beta_0 + \beta_1 x_1^i + \beta_2 x_2^i + \cdots + \beta_7 x_7^i)^{y_i}}{\exp(\exp(\beta_0 + \beta_1 x_1^i + \beta_2 x_2^i + \cdots + \beta_7 x_7^i)) y_i!} \tag{6}$$

Again, for numerical stability we converted this to a log likelihood:

$$\ln \pi(y \mid \beta) \propto \sum_{i=0}^{n} y(\beta_0 + \beta_1 x_1^i + \cdots + \beta_7 x_7^i) - \exp(\beta_0 + \beta_1 x_1^i + \cdots + \beta_7 x_7^i) - \ln(y!) \tag{7}$$

There is a chance that the likelihood is incorrect due to the assumptions we made. For a Poisson regression model there are four 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 for $i = 1, \ldots, 7$ [3]. It's somewhat likely that the mean and variance of our observed variable are not the same. In this case we would need to switch to a negative binomial-based model to account for dispersion, or use a quasi-likelihood [3].

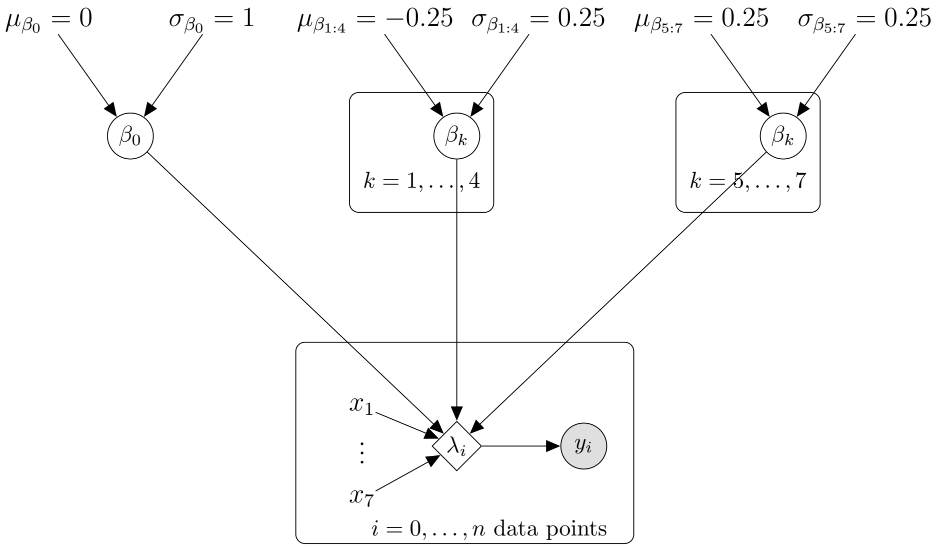
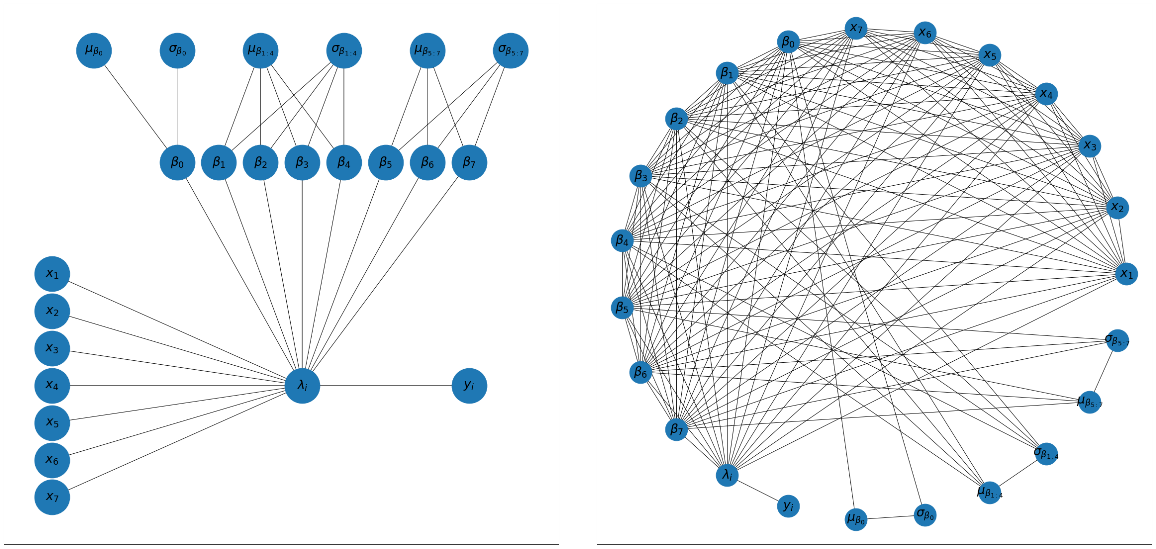

$\mu_{\beta_0} = 0$
$\sigma_{\beta_0} = 1$
$\mu_{\beta_{1:4}} = -0.25$
$\sigma_{\beta_{1:4}} = 0.25$
$\mu_{\beta_{5:7}} = 0.25$
$\sigma_{\beta_{5:7}} = 0.25$
β_0
β_k
$k = 1, \ldots, 4$
β_k
$k = 5, \ldots, 7$
x_1
$\vdots$
x_7
λ_i
y_i
$i = 0, \ldots, n$ data points

Figure 1: Belief network for Poisson regression model. x_j represents the features used to predicted the observed data y_i for i observed data points. There are 8 parameters that need to be learned represented by β_k, where $k = 0, 1, \ldots, 7$. λ_i is a deterministic variable that results from the Poisson regression with the relationship $\log(\lambda_i) = y_i$.



(a) Markov network and Skeleton graph.

(b) Moralized and triangulated graph.

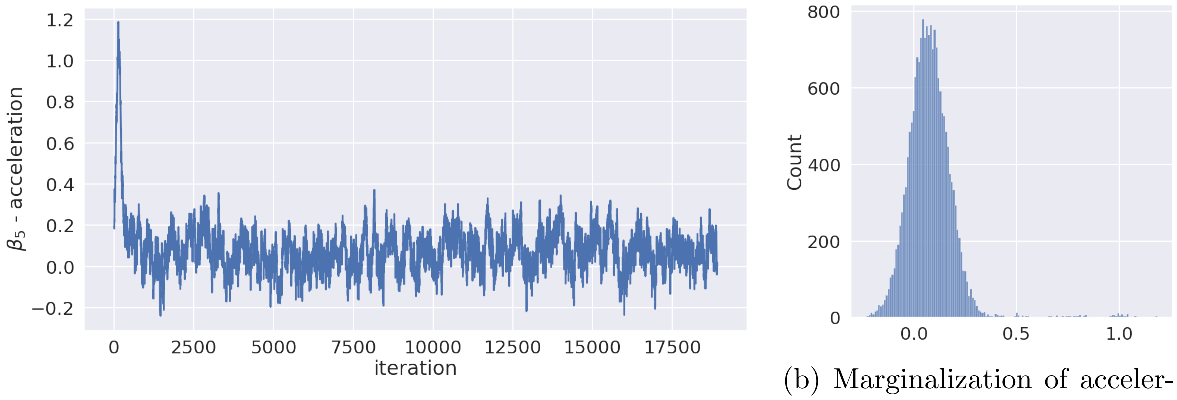
Figure 2: Various types of graphs/networks representing the Poisson regression model. These figures were generated using Networkx. The triangulated graph was verified with Networkx's is_chordal function.

For our specific problem, a belief network is more appropriate than a Markov network. This is due to the direct causal nature of our data, as well as the direct relationship between inputs and output in our deterministic model. There is no need for the capability to represent cyclic dependencies, leading us toward a fairly straightforward decision of using the belief network shown in figure 1.

3.1 Metropolis-Hastings Algorithm

To learn the β parameters we use the Metropolis-Hastings (MH) algorithm, which is a variant of MCMC. This algorithm depends on the posterior being correctly proportional to the real posterior. This is because in MH, the acceptance ratio will eliminate any constants. The assumptions we made earlier may be satisfied given that MPG is truly a Poisson distribution, but it is also somewhat likely that they are not, due to the mean and variance not being equal. There is a better (and still computationally efficient) way to sample, should it be given that the distribution is not truly Poisson. As mentioned above, this would be by using a negative binomial likelihood in place of the Poisson to capture dispersion, or a quasi-likelihood. It's possible that MPG experiences over-dispersion, meaning that the variance is equal to the mean times some constant [3]. If this is the case, our assumptions would not be satisfied for MH.

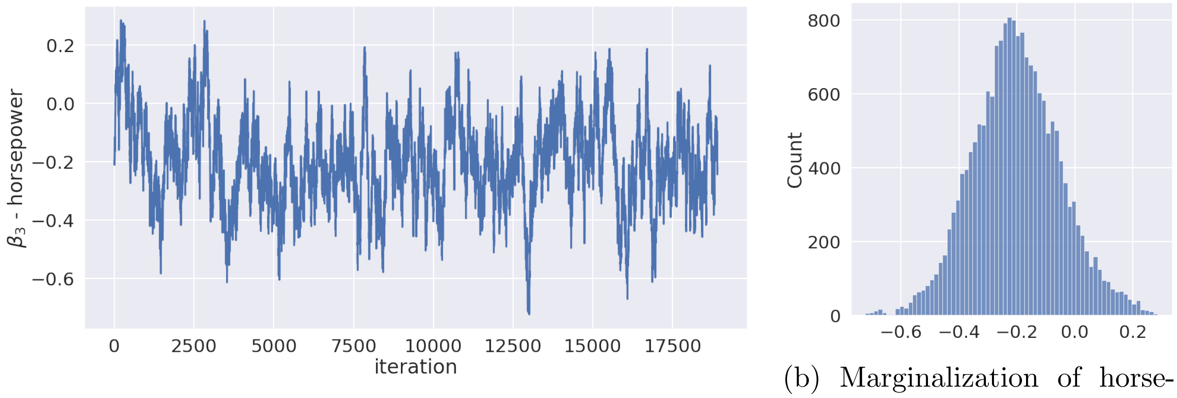
For the sampling algorithm we created an 80/20 test-train split with the data. Then we ran MH for $100,000$ iterations with a burnout of 25% (meaning we discarded the first $25,000$ iterations in hopes that we see convergence by that point). For our proposal functions we sampled from eight normal distributions, respectively centered at each β parameter, with a standard deviation of 0.025. This yielded an acceptance rate of 23.317%. We initiated the parameters as 0.2 and -0.2 depending on if we expected them to be negative or positive, the bias we initiated to 0.



(a) Trace of acceleration parameter β_5.

(b) Marginalization of acceleration parameter β_5.

Figure 3: Trace plot and marginal for the acceleration parameter. The diagrams show only accepted samples, and the burn in period has not been removed.



(a) Trace of horsepower parameter β_3.

(b) Marginalization of horsepower parameter β_3.

Figure 4: Trace plot and marginal for the horsepower parameter. The diagrams show only accepted samples, and the burn in period has not been removed.

We also needed to convert the acceptance ratio for the MH algorithm since we are using log priors and a log likelihood. The updated acceptance ratio α for MH is:

$$\alpha = \exp(\ln \pi(y \mid \beta') + \ln \pi(\beta') - \ln \pi(y \mid \beta) - \ln \pi(\beta)) \tag{8}$$

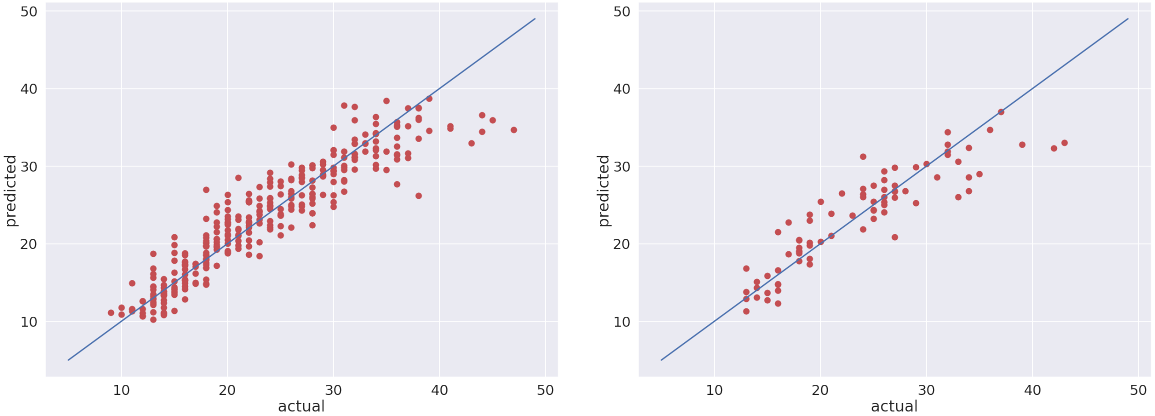
Where β' is the candidate and β is the current sample. This allows us to avoid overflows from multiplying small probabilities together.

One can see in figures 3 and 4 that both trace plots and marginal plots are essentially what we expected. Both marginals look nearly identical to normal distributions, which we expect due to their respective priors. Additionally, the mean weights are positive for acceleration and negative for horsepower, which we also predicted. It can be seen on the trace plots that convergence did indeed take a bit (hence the usefulness of a burn-in period in the MH algorithm), leading to slight tails on the marginalization plots (more noticeable on the acceleration plot).

4 Results

After running MH, we averaged the results on the samples, excluding the ones during the burn in period. With the train set we achieved an R^2 of 0.86758 (we are explaining about 87% of the variance in our predicted variable MPG). The mean squared error (MSE) for the train set is 8.327. Figure 5a shows the difference between the actual MPG and the predicted MPG in the train set. Overall, our model does quite well with fitting the data. When we look at the test set we see a slight drop in performance, which is generally expected. The R^2 for the test set is 0.806 and the MSE is 10.34. Figure 5b shows the difference between the actual MPG and the predicted MPG in the test set.

Some potential improvements to this model could be swapping the Poisson regression model out for a negative binomial, allowing us to capture any dispersion that is occurring. We can observe that in figure 5 as the MPG increases (the points tend to dip a little towards the higher MPGs). This is likely because there is some over-dispersion in the observed data.



(a) Error in train set predictions.

(b) Error in test set predictions.

Figure 5: The error in both the train and test sets, points closer the $x = y$ line are more accurate.

5 Conclusion

Overall, we are very pleased with our results. We had various concerns with our priors, as we previously noted that our assumption of independence between the variables was a bit naive and could've quite easily thrown us off course. Additionally, we were a bit apprehensive about our chosen likelihood (mean and variance possibly not being equal as a Poisson distribution assumes). However, upon seeing the relative accuracy of our model in predicting automobile fuel efficiency, we may have over-concerned ourselves with our assumptions and model choice.

6 Code and Data

Code for the project can be found here: repository and the dataset here: link.

References

[1] David Barber. *Bayesian reasoning and machine learning.* Cambridge University Press, 2012, pp. 559–560.

[2] Dheeru Dua and Casey Graff. *UCI Machine Learning Repository.* 2017. URL: http://archive.ics.uci.edu/ml.

[3] Paul Roback and Julie Legler. *Beyond Multiple Linear Regression: Applied Generalized Linear Models And Multilevel Models in R.* CRC Press, 2021. Chap. 4.