Machine Learning for Data Science HW6 Baysian methods

Nikolay Kormushev

MCMC

Distance weight

I assume that beta will be normally distributed and the mean will be negative because the farther we are from the hoop we should be penalized more. The normal distribution is because the changes in the penalty should gradually change results.

My assumptions are confirmed by Figure 1.

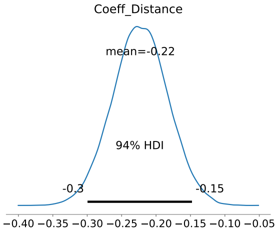

Figure 1. Distance weight distribution

50 vs all samples

In Figure 3 we see that in both cases we get a normal distribution of the data but with 50 samples it has a much bigger variance. This is to be expected since we have less information to update our beliefs on. In other words with less samples a wider distribution of possible weights have a higher posterior probability/give good results for the regression.

Which is more important for shot success, angle or distance?

I would say the distance is more important which can also be seen from the weight for the distance being higher than the one for the angle. I estimated $P(wDistance > wAngle)$ and got 99% which confirms this. It also makes sense since if you are too far the chances of scoring can go to 0 while the angle influence is determined based on the distance.

Does shot success increase or decrease with increasing angle (the further on the sides we are)?

Looking at the negative mean of the posterior it decreases since the results with a penalized angle are most likely: $P(wAngle < 0) = 69\%$. Nevertheless there are cases when it is positive maybe because some angles can have a positive effect.

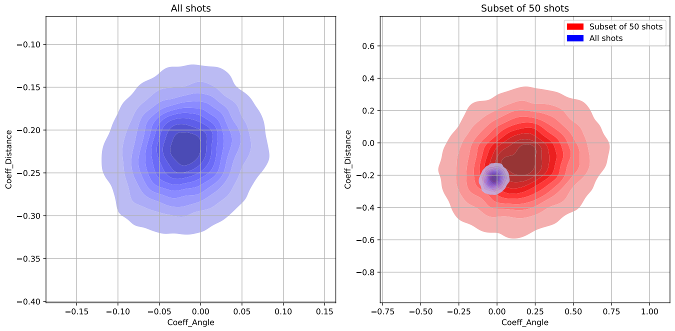

Figure 2. Distance, Angle scatterplot

Laplace Approximation

Derivations for gradient and hessian

$$
\begin{aligned}
\frac{\partial P(\beta|y)}{\partial \beta_k} &= -\frac{1}{N}\sum_{i=1}^{N}\left[\frac{\partial y_i \log(\sigma_i)}{\partial \beta_k} + \frac{\partial (1-y_i)\log(1-\sigma_i)}{\partial \beta_k}\right] \\
&- \frac{\partial}{\partial \beta_k}\sum_{i=1}^{W}\left(-\frac{1}{2}\left(\frac{\beta_i}{\mathrm{std}_0}\right)^2\right) \\
\frac{\partial \log \sigma_i}{\partial \beta_k} &= \frac{1}{\sigma_i}\cdot\frac{\partial \sigma_i}{\partial \beta_k} = (1-\sigma_i)\cdot x_{i,k} \\
\frac{\partial \log(1-\sigma_i)}{\partial \beta_k} &= \frac{-1}{1-\sigma_i}\cdot\frac{\partial \sigma_i}{\partial \beta_k} = \sigma_i \cdot x_{i,k} \\
\frac{\partial \sigma_i}{\partial \beta_k} &= \sigma_i\cdot(1-\sigma_i)\cdot\frac{\partial(-\beta\cdot X_i)}{\partial \beta_k} = \sigma_i\cdot(1-\sigma_i)\cdot x_{i,k}
\end{aligned}
$$

We substitute and get:

$$
\begin{aligned}
\frac{\partial P(\beta|y)}{\partial \beta_k} &= -\frac{1}{N}\sum_{i=1}^{N} y_i\cdot(1-\sigma_i)\cdot x_{i,k} + (1-y_i)\cdot\sigma_i\cdot x_{i,k} \\
&+ \frac{\beta_k}{std_0^2} = -\frac{1}{N}\sum_{i=1}^{N} x_{i,k}\cdot y_i - x_{i,k}\sigma_i + \frac{\beta_k}{std_0^2} \\
\frac{\partial P(\beta|y)}{\partial \beta_k \partial \beta_j} &= -\frac{1}{N}\sum_{i=1}^{N} -x_{i,k}\frac{\partial \sigma_i}{\partial \beta_j} + \frac{1\{j=k\}}{std0^2} \\
&= \frac{1}{N}\sum_{i=1}^{N} x_{i,k}\cdot x_{i,j}\cdot\sigma_i\cdot(1-\sigma_i) + \frac{1\{j=k\}}{std0^2}
\end{aligned}
$$

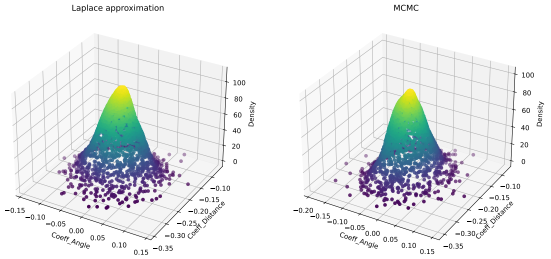

Figure 3. Laplace vs MCMC

Implementation and comparison

For the implementation I use an fmin_bfgs to optimize the posterior of the betas which is the log loss multiplied my assumption of the prior. I get the Hessian and invert it with my derivations. I used the inverse hessian to get the curvature. The diagonal is the variance matrix which I use to find the std. I compared with the Inverse Hessian from the optimizer which was not normalized initially and I needed to do that to get a similarly scaled distribution to MCMC.

From Figure 3 we see the results of the two models are identical with very small differences like outliers, Laplace having a bit higher density in its center and the means being slightly different.