STATS305B – Lecture 8

Jonathon Taylor
Scribed by Michael Howes

01/31/22

Contents

1 Bayesian GLMs

By putting a prior on the parameters β we can turn the logistic regression model into a Bayesian model for binary data. Suppose we have $\beta|X \sim g$ where g is a density and that

$$Y|X,\beta \sim \text{Bernoulli}\left(\frac{\exp(X^T\beta)}{1+\exp(X^T\beta)}\right),$$

that is,

$$\text{logit}(\mathbb{P}(Y=1|X,\beta)) = X^T\beta.$$

The log-posterior is,

$$\begin{aligned}\log(g(\beta|Y)) &= \log L(\beta|Y) + \log(g(\beta)) \\ &= \sum_{i=1}^n Y_i X_i^T\beta - \log\left(1+\exp(X_i^T\beta)\right) + \log(g(\beta)).\end{aligned}$$

In general this is not going to be recognizable as a familiar distribution. This is a problem since when doing a Bayesian analysis we often want to draw samples from the posterior distribution to construct credible intervals, or calculate empirical posterior means. Fortunately there is a huge literature about how to sample from complicated distributions.

1.1 Metropolis–Hastings

Metropolis–Hastings is one sampling method that is based on constructing a Markov chain with stationary distribution $g(\beta|Y)$. We start with a proposal kernel $K : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$ which is a function with the property that for all $\beta \in \mathbb{R}^p$, $K(\cdot|\beta)$ is a probability density. One common choice of proposal

kernel is $k(\beta'|\beta) = \mathsf{N}(\beta, \Sigma)$ where Σ is fixed. For metropolis–Hasting we iteratively sample β' from $K(\cdot|\beta^{(k)})$ we then define $\beta^{(k+1)}$ to be β' with accept probability,

$$\min\left(1, \frac{g(\beta'|Y)K(\beta|\beta')}{g(\beta|Y)K(\beta'|\beta)}\right),$$

and we set $\beta^{(k+1)} = \beta^{(k)}$ otherwise. The resulting sequence $\{\beta^{(k)}\}_{k\geq 0}$ is a Markov-chain with stationary distribution $g(\beta|Y)$. The hope is that the sequence converges quickly to this stationary distribution. Note that if K is a symmetric function (such as $K(\beta'|\beta) = \mathsf{N}(\beta, \Sigma)$), then the accept probability becomes

$$\min\left(1, \frac{g(\beta'|Y)}{g(\beta|Y)}\right).$$

1.2 Gibbs sampling

It is often easier to sample from univariate distributions than from multivariate distributions. Gibbs sampling tries to exploit this. For each $j = 1, \ldots, p$ define

$$g_j(\beta_j|\beta_{-j}, Y) = \frac{g(\beta|y)}{\int_{-\infty}^{\infty} g(\beta|Y)d\beta_j}.$$

That, if $g_j(\cdot|\beta_{-j}, Y)$ is the conditional density of β_j. It is often the case that $g_j(\cdot|\beta_{-j}, Y)$ is a recognizable density even if $g(\beta|Y)$ isn't (but there is no guarantee this will happen). The Gibbs cycles through $j = 1, \ldots, p$ drawing samples β_j^{new} from $g(\cdot|\beta_{-j}, Y)$ and then updating the j^{th} coordinate of the sample. Sometimes the index j is chosen randomly.

1.3 Hit-and-run sampling

Suppose we are using the latent threshold model for Y. That is we have binary data Y with

$$\mathbb{P}(Y_i = 1|X_i) = F(X_i^T\beta),$$

for come CDF F. This model corresponds to having *latent variables* $T_i \overset{\text{iid}}{\sim} F$ such that,

$$Y_i = \begin{cases} 1 & \text{if } T_i \leq X_i^T\beta, \\ 0 & \text{else .} \end{cases}$$

We then have $g(\beta|Y) = g(\beta|T \leq X\beta)$. Thus instead of sampling from $g(\beta|Y)$ we can instead draw sample (T, β) from $(F, g(\cdot))$ and then only keep the samples that satisfy $T_i \leq X_i^T\beta$ if and only if $Y_i = 1$. We can then throw away that T's and keep the sampled β which be distributed according to $g(\beta|T \leq X\beta)$.

This is especially easy to do if we have a probit model and a normal prior on β. In that case, we wish to sample from $Z \sim \mathsf{N}(\mu, \Sigma)|AZ \leq b$ for some fixed A and b. By transforming A and b, we may assume that $\Sigma = I$. The inequality $AZ \leq b$, conditioned on Z_{-j} reduce to,

$$L(Z_{-j}, A, b) \leq Z_j \leq U(Z_{-j}, A, b).$$

Thus, we can simply draw Z_j from $\mathsf{N}(\mu_j, 1)$ truncated to the interval $[L, U]$. This gives an algorithm similar to Gibbs sampling.

1.4 STAN

There are many samplers out there and their properties are active areas of research. For the applied Bayesian statistician though, there are built in programs that sample effectively for most problems. The programming language STAN is built for applied Bayesian analyses. The most common sampler used in STAN is not metropolis–Hastings or the Gibbs sampler but rather a type or Markov chain sampler based on Hamiltonian dynamics.

STAN can be used to create maximum a posteriori estimates, to calculate posterior means and to calculate credible intervals.

2 Multinomial models

Often our response Y is not a binary variable but can actually take k different values. To create a regression model for a categorical response, we can work with the multinomial exponential family. Recall the multinomial distribution is given by,

$$f(y_1, \ldots, y_k|\pi) = \binom{N}{y_1, \ldots, y_k} \prod_{j=1}^{k} \pi_i^{y_j},$$

where $\sum_{j=1}^{k} y_j = N$ and $\sum_{j=1}^{k} \pi_j = 1$ and $\pi_j \geq 0$ for all j. We have seen before that this choice of parameters only has $k-1$ degrees of freedom despite having k parameters. To work around this, we will use category k as a *baseline category*. That is we have parameters

$$\left\{ \pi : \pi_j \geq 0 \text{ for } j = 1, \ldots, k-1 \text{ and } \sum_{j=1}^{k-1} \pi_j \leq 1 \right\}.$$

We then set $\pi_k = 1 - \sum_{j=1}^{k-1} \pi_j$. The density $f(\cdot|\pi)$ is supported on $y \in \mathbb{Z}_+^{k-1}$ such that $\sum_{j=1}^{k-1} y_j \leq N$ and

$$f(y_1, \ldots, y_{k-1}|\pi) \propto \left(\prod_{j=1}^{k-1} \pi_j^{y_j} \right) \left(1 - \sum_{j=1}^{k-1} \pi_j \right)^{N - \sum_{j=1}^{k-1} y_j}.$$

Thus,

$$\log L(\pi|Y) = \sum_{j=1}^{k-1} y_i \log \left(\frac{\pi_i}{1 - \sum_{l=1}^{k-1} \pi_l} \right) + N \log \left(1 - \sum_{l=1}^{k-1} \pi_l \right).$$

We therefore have an exponential family with sufficient statistic $(y_1, \ldots, y_{k-1})$ and natural parameters,

$$\eta_j = \log \left(\frac{\pi_j}{1 - \sum_{l=1}^{k-1} \pi_l} \right).$$

Now suppose we have categorical data $(X_i, Y_i)_{i=1}^n$ where $X_i = X[i, \cdot] \in \mathbb{R}^p$ are covariates and $Y_i \sim \mathsf{Multinomial}(N_i, \pi_i)$. We can create a regression model where,

$$\eta_{i,j} = X[i, \cdot]^T \beta[\cdot, j].$$

The parameters β live in $\mathbb{R}^{p \times (k-1)}$ since we have p covariates and k categories. We also have $X \in \mathbb{R}^{n \times p}$ as usual and thus $\eta \in \mathbb{R}^{n \times (k-1)}$ since we have n values of Y and each Y_i is one of k categories. In this model, the original parameters $\pi_{i,j}$ are given by

$$\begin{aligned} \pi_{i,j} &= \frac{\exp(\eta_{i,j})}{1 + \sum_{l=1}^{k-1} \exp(\eta_{i,l})} \\ &= \frac{\exp(X[i, \cdot]^T \beta[\cdot, j])}{1 + \sum_{l=1}^{k-1} \exp(X[i, \cdot]^T \beta[\cdot, l])}. \end{aligned}$$

This is the *baseline logit model*. The function

$$\eta \mapsto \left(\frac{\exp(\eta_{i,j})}{1 + \sum_{l=1}^{k-1} \exp(\eta_{i,l})} \right)_{j=1}^{k-1},$$

is called the *soft-max* function. The soft max function can be thought of as a smooth version of argmax.

2.1 Fitting a baseline logistic model

The log-likelihood from β is,

$$\begin{aligned}
\log L(\beta|Y) &= \sum_{i=1}^{n} \log L(X[i,\cdot]^T \beta | Y_i) \\
&= \sum_{i=1}^{n} \sum_{j=1}^{k-1} Y_{i,j} X[i,\cdot]^T \beta[\cdot, j] + \sum_{i=1}^{n} N_i \log \left(1 - \sum_{l=1}^{k-1} \frac{\exp(X[i,\cdot]^T \beta[\cdot, l])}{1 + \sum_{l'=1}^{k-1} \exp(X[i,\cdot]^T \beta[\cdot, l'])} \right) \\
&= \sum_{i=1}^{n} \sum_{j=1}^{k-1} Y_{i,j} X[i,\cdot]^T \beta[\cdot, j] + \sum_{i=1}^{n} N_i \log \left(\frac{1}{1 + \sum_{l=1}^{k-1} \exp(X[i,\cdot]^T \beta[\cdot, l])} \right) \\
&= \sum_{i=1}^{n} \sum_{j=1}^{k-1} Y_{i,j} X[i,\cdot]^T \beta[\cdot, j] - \sum_{i=1}^{n} N_i \log \left(1 + \sum_{l=1}^{k-1} \exp(X[i,\cdot]^T \beta[\cdot, l]) \right) \\
&= \sum_{i=1}^{n} Y_i^T (X\beta)[i,\cdot] - \sum_{i=1}^{n} N_i \log \left(1 + \sum_{l=1}^{k-1} \exp(X[i,\cdot]^T \beta[\cdot, l]) \right) \\
&= \mathrm{Tr}((X\beta) Y^T) - \sum_{i=1}^{n} N_i \log \left(1 + \sum_{l=1}^{k-1} \exp(X[i,\cdot]^T \beta[\cdot, l]) \right).
\end{aligned}$$

Note that we can write,

$$-\log L(\beta|Y) = \sum_{i=1}^{n} \left[Y_i^T (X_i \beta) - \Lambda(X_i^T \beta) \right],$$

where Λ is the cummulant generating function for Y. Thus,

$$\begin{aligned}
\nabla - \log L(\beta|Y) &= \sum_{i=1}^{n} Y_i^T X_i - \mathbb{E}_\beta[Y_i] \\
&= X^T (Y - \mathbb{E}_\beta[Y]),
\end{aligned}$$

where $\mathbb{E}_\beta[Y_i]_j = N_i \frac{\exp(X[i,\cdot]^T \beta[\cdot, j])}{1 + \sum_{l=1}^{k-1} \exp(X[i,\cdot]^T \beta[\cdot, l])} = N_i \pi_i(\beta)_j$. Note that since $\beta \in \mathbb{R}^{p \times (k-1)}$, $\nabla - \log L(\beta|Y) \in \mathbb{R}^{p \times k-1}$ also. We also have

$$\frac{\partial^2}{\partial_{\beta_{ij}} \partial_{\beta_{lm}}} - \log L(\beta|Y) = \sum_{c=1}^{n} X_{c,i} X_{c,l} \mathrm{Cov}_\beta(Y_c)_{jm},$$

where, The full Hessian $\nabla^2 - \log L(\beta|Y)$ is a *tensor* in $\mathbb{R}^{p \times (k-1)} \bigotimes \mathbb{R}^{p \times (k-1)}$. It takes in a matrix in $\mathbb{R}^{p \times (k-1)}$ and outputs a new matrix in $\mathbb{R}^{p \times (k-1)}$.