Final Project Documentation

Angela Zheng and Oluwatitomi Adebajo

19/12/2022

Introduction

This document summarizes the various algorithms that the GLMM package uses in order to provide a solution to the research questions involved with the `epilepsy.csv` data set. Notably, each function within the package serves as its own algorithm to aid in making likelihood-based inferences towards understanding the Poisson GLMM's fit to the data.

The question being answered with respect to the data is: does the drug Progabide significantly reduce the number of seizures a patient experiences after accounting for age?

Maximum Likelihood Inferences

Linear Predictor Form

The model used in this package takes on the following form:

$$\mu_i = \exp(\beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} x_{3i} + \beta_3 x_3 i + Z_i)$$

where β_1 = age, β_2 = treat*expind, β_3 = expind, and Z_i represents the random effect ($Z_i \sim \mathrm{N}(0, \sigma^2)$). β_2 represents the effect of the drug that is administered *during* the clinical trial. As such, we are interested in testing whether there is significant interaction between the drug and the measurement period.

Parameter Estimates

The algorithm within **estimation_functions.R** uses the `run_model()` function to find the estimates for β_0, the $\hat{\beta}$'s, $\hat{\sigma}^2$, the Z_i's, and the t-test statistics for each β predictor. As a result, `run_model()` outputs a printed summary that is similar to functions such as `lm()` and `summary.lm()`. The point estimates for β_0 and $\hat{\sigma}^2$ are based on the expressions derived where:

$$\hat{\beta}_0 = -log\left(\frac{-n}{\sum_{i=1}^{n}\sum_{j=1}^{k}(y_{ij}e^{-x_i\beta})e^{Z_i^{(t)}}}\right)$$

and

$$\hat{\sigma}^2 = \frac{\sum_{i=1}^{n} Z_i^{2(t)}}{n}$$

However, since the other outputted features of `run_model()` cannot be computed analytically such as the point estimates for the β's, the random effects Z_i's, and the t-test statistics for each β, it is simpler to just use the existing R function `glmer()` from the lme4 package to find these values.

Standard Errors

The standard errors for the parameter estimates are calculated in the **bootstrapSE.R** file with a function called `btsp()`. This function is based on the following parametric bootstrapping algorithm:

1. Randomly simulate a set of $\hat{z}_i$ value using the estimate of $\hat{\sigma}^2$ from `run_model()`'s output (based on fitting the model to the cleaned `epilepsy` data set) which includes the `sigmasq` component
    - This step uses the `rnorm()` function to randomly generate values since we know that $Z_i \sim N(0, \sigma^2)$
2. Compute μ_i now that we have all the estimates $\hat{beta}_0$, β, and z_i using the equation where $\mu_i = \exp(\beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} x_{3i} + \beta_3 x_{3i} + Z_i)$
    - This step uses the z_i values from Step 1 as well as the $\hat{\beta}_0$ and β estimates obtained from`run_model()`'s `beta` component
3. Simulate an entirely new set of Y_{ij} values now that we have the computed μ_i from Step 2
    - This step uses the `rpois()` function to randomly generate values since we know that $Y_{ij} \sim \text{Poisson}(\mu_i)$
4. Input the simulated data of Y_{ij} values into the `run_model()` function where the point estimates can be returned and stored
5. Repeat Steps 1 to 4 using the `replicate()` function B times which is the value that the user passes to the function to indicate the number of bootstrap replications

Consequently, this bootstrapping algorithm will compute the standard errors for all parameters using the `sd()` function based on each parameter's samples of size B.

One consideration to make note of is that because the range of seizure values in the epilepsy data set is so large (0 to 302 seizures) there are instances when the model cannot converge when the simulated data is used. Since we cannot trust the estimates if the model did not converge, the btsp() function ensures that those estimates are not used in the final calculation of the standard error, in order to preserve accuracy.

Confidence Intervals

The confidence intervals utilize the point estimates from the `run_model()` function as well as the standard errors from the `btsp()` function. The function to implement confidence intervals are found in the **confidenceIntervals.R** file and this algorithm is implemented within the `ci()` function. The confidence intervals for the β_0, β, and σ^2 estimates are based on the following expressions:

$$\text{C.I for } \beta_0 = \hat{\beta}_0 \pm 1.96(SE(\hat{\beta}_0))$$

and

$$\text{C.I for } \beta = \hat{\beta} \pm 1.96(SE(\hat{\beta}))$$

and

$$\text{C.I for } \sigma^2 = \left[\frac{(n-1)\hat{\sigma}^2}{\chi^2_{\alpha/2,n-1}}, \frac{(n-1)\hat{\sigma}^2}{\chi^2_{1-\alpha/2,n-1}} \right]$$

Therefore, `ci()` calls on the `run_model()` and `btsp()` functions to save the parameters' point estimates and standard errors respectively, and then computes the confidence intervals based on the expressions above for β_0, β, and σ^2.

Hypothesis Testing

The fitting of the model is used to answer research questions about the data set. We are interested in whether the drug significantly reduces the number of seizures a patient experiences after accounting for age. That means we need to find the effect of the interaction between the two treatment groups and the two periods of measurement. Specifically, we must look at the interaction between **treat** and **expind** within the `epilepsy.csv` data. The predictor for this interaction, denoted as β_2, can be tested with the following statement:

$$H_0 : \beta_2 = 0$$
$$H_a : \beta_2 < 0$$

This research question requires a t-test based on the Student's t-distribution. The hypothesis test is fairly simple with the use of the `run_model()` function, as one of its components already includes `test_stat`. This is the t-test statistic, denoted by t^*, which is then tested against the critical value, denoted by $t_{\alpha,n-1}$. This can be rewritten as $P(t^* < t_{\alpha,n-1})$. A common significance level of $\alpha = 0.05$ can be used to conduct this test.

To find the critical value $t_{\alpha,n-1}$, we can use the `qt()` function. If the test statistic for the interaction predictor between **expind** and **treat** is less than the critical value, reject the null hypothesis and assume that the alternative hypothesis is true. Otherwise, do not reject the null hypothesis.

If the null hypothesis, H_0, is true, this is equivalent to the claim that the drug does *not* significantly reduce the number of seizures a patient experiences. In other words, the drug does not have an effect on the subjects. If the alternative hypothesis, H_a, is true, this is equivalent to the claim that the drug does significantly reduce the number of seizures a patient experiences.

Based on the results of the vignette, the test statistic was not less than the critical value, hence we failed to reject the null hypothesis. One thing to keep in mind however, is that there were significant outliers in the number of seizures seen in the data. The reason this is important is because outliers tend to increase our estimated standard errpr, and this reduces the value of the test statistic (as it is calculated in the following way: $t' = \frac{\hat{\beta_2}-0}{SE(\hat{\beta_2})}$). A smaller test statistic means it becomes less likely to reject the null-hypothesis. Although the results of this data gives us reason to reject the null hypothesis and conclude the drug is not effective, given that the value of the test statistic is so close to the critical value, we would suggest that more data is gathered, in order to draw more conclusive results about the effect of the drug.