MATH 3338: Mathematical Modeling (Fall 2019)

Instructor: Dr. Hoa Nguyen
Project Advisor: Courtney Rohde, ACAS

Project Title
Using GLMS to Estimate Auto Insurance Losses by Policy Characteristics

Phase I Information

Task 1: Calculate the Indication *(30 mins)*

This isn't a model, so let's get it out of the way first. I've uploaded an Excel workbook to the project Google drive. It's all set up to calculate the indicated rate change for our imaginary company, "MBM Insurance," except that it's missing some key figures in the yellow cells. Fill in the cells using the indications formulas from the workbook column titles and/or my first presentation (I've included a PDF version on the project Google drive) to complete the calculation.

The yellow cells are only on the first tab. The other two tabs show the more detailed derivation of the development and on-level factors, respectively.

I don't intend for this task to take you a very long time, but I do encourage you to explore the whole workbook. Remember that the goal is to make our best guess at what the premiums and losses will be in the future period which we are considering. You should be able to get an intuitive idea of what types of adjustments are made to the premium and loss data, and why they are necessary for making rates on a *prospective* basis.

Each student on the team should fill out the workbook themselves. I encourage each of you to make your best effort individually before you collaborate. But do compare your results, and discuss your understanding of the workbook if that would be helpful.

The output this task is the indicated rate change in % form (to one decimal place) – the orange cell D35 of the indications workbook.

Task 2: Complete the R Demo *(20-40 mins)*

I've written Demo instructions in a separate document. Download the instructions, data, and R file from the project Google drive & jump right in.

The goal of Task 2 is for the students to get oriented to the basic functions of the R interface. You will also learn about the way R assigns values to variable names, and what types of *arguments* (inputs) are needed by the functions we want to use. The demo will walk you through creating a very straightforward linear model using a tiny example data set.

Each student on the team should go through the demo individually. Each of you should be comfortable with the functions in the demo before moving on.

You can definitely put R on your personal laptop and run through the demo just fine – it's not very big. However, when we get to the actual pure premium data in Task 3, a standard laptop may be sluggish dealing with 100,000 rows of data. If this is a problem, you might ask Dr. Nguyen if Trinity has a faster computer in a lab that you could use for Task 3.

Task 2 doesn't have an output, but success would be to replicate the summary results and plots from the Demo instructions.

Task 3: Run the first GLM *(1-2 hours)*

1. Data

The main goal of this phase is for the students to model a pure premium dataset to determine indicated rating factors. The source dataset is called "ppdata_total.csv" and can be found on the project Google drive. Before modeling data, we must understand what the data represents. The data consists of 100,000 records for MBM Insurance. For each record, information has been collected for the following dimensions:

- Gender – represents the gender of the insured
- Rating Area – represents the territory (i.e. place of residence) of the insured
- NCD – represents the number of years that the insured has been claims-free
- Protected NCD – indicates whether or not the insured has purchased accident forgiveness
- Driving Restriction – limitations on who can drive the vehicle
    - Named – named insured only
    - Any – any driver
- Vehicle Age – age of the vehicle
- Loss Year – year in which the loss occurred
- Exposures – represents the credibility of each observation (i.e. the weight)
- Developed Loss – raw loss developed to ultimate
- Pure Premium – developed loss divided by the exposure

I have separated the data into 80% and 20% samples, called the "Train" and "Test" datasets, respectively. We will build the model on the Train data, and we will use the Test data in a later phase to evaluate how well our model will perform on data it hasn't seen before. For this phase, use the "ppdata_train.csv" file from the project Google drive.

I recommend that you use excel to open this file and explore the types of values it has. Familiarize yourself with the variable names (column headers) and the level names (different entries for each variable).

Think about what type of results you might expect from a model built on auto insurance data with these variables. Which characteristics might you expect to have higher or lower predicted values? Which might indicate a relatively "risky" policy?

Please note that this sample data comes courtesy of **Willis Towers Watson.** I am greatly appreciative!

2. Setting up the R script

I have provided some of the model code that needs to be run to fit the Generalized Linear Model (GLM). It is called "PurePremiumModel.R" and can be found on the project Google drive.

Since this GLM uses the special (and unusual outside of the insurance context) Tweedie probability distribution, we need to download some special functions for R, or else the software won't know what we are talking about when we ask it to use the Tweedie distribution. These functions come in packages. With R open, go to Packages > Install Packages… If this is the first time you're doing this, it will ask you to select a "CRAN mirror." You can just pick USA (TX) if that's an option. In the next window, you can scroll through a list of available packages. They are alphabetized. I want you to get the "tweedie" package. If it tells you your library is not writeable and asks if you want to make a personal library instead, you can say yes. Then in the R main interface, it should tell you "tweedie" has been successfully unpacked. Repeat the process to get the "statmod" and "broom" packages. "Broom" may take a minute to install.

Then open the R script and update all the file paths to point to your project folder.

I've left out model formula in both the "glm" and "lm" lines, so you can figure these out yourselves. The format is there, with the other arguments besides the formula (such as "data = ppdata" and "family =tweedie") but where it says "y ~ x1 + x2 + ... + x3", those names won't work. You need to replace the y and x values with the variable (column) names in your data.

Assume the rating formula in practice is as follows:

Premium = Base Rate * Gender Factor * Rating Area Factor * NCD Factor * Protected NCD Factor * Driving Restriction Factor * Vehicle Age Factor

Look at your data. What is your response variable (y), that we are trying to predict? What are your predictor variables (x), which cause the predicted loss to vary?

There is no base rate in the data. Should it be mentioned in the formula? Why? To understand this, go back to the demo. Where did the intercept term come from? Was that in the tiny sample data set?

Are there columns in the data that you don't use? Why? What purpose do they serve?

One note about the Vehicle age factor. If you include that one in your model as "VehicleAge," it will only produce one coefficient for that variable, instead of one for every non-base level like it does for the other variables. This is because R is pretty smart – it recognizes that VehicleAge has a lot of numerical levels that seem to go in order, suggesting it is a continuous variable. Which it is. Then R assumes we want to fit a curve to this variable, (that is, estimate a single slope parameter) rather than estimating a parameter for every non-base level like we do for categorical variables. This is a great approach for most continuous variables, and it would likely improve the model if we let R fit a curve. But this is called a "simplification" on the variable, and I want to save those for Phase II. So for now, to have R fit each level of VehicleAge separately, include it in your model formula as "as.factor(VehicleAge)" which should produce one fewer coefficients than there are levels, matching my results.

3. Fitting the Models

Once you have filled in the rating formulas and updated all file paths, you should be able to run the script in its entirety. I expect it will take only a few seconds to run on most machines. In addition to the Tweedie GLM, the code also fits a classical linear model to the data.

The code will export two sets of coefficients to your project folder (the last two lines of code) – one for the GLM, and one for the linear model.

To determine whether or not you have properly run the code, you can compare these exports to the ones I've uploaded to Google Drive. If you get different results, consider how the formula I used may have looked. How can you edit yours to get the same results?

4. Determine the indicated rating factors

With a log link function, indicated rating factors are determined by exponentiating each model coefficient. For example, if the model coefficient for "Gender = Male" is 0.534, then the indicated rating factor is $e^{0.534} = 1.706$. Remember that each rating variable (i.e. Gender, Vehicle Age, Driving Restriction, etc.) will have one level missing. Each missing level has an implicit model coefficient of 0. Thus, the indicated rating is $e^{0} = 1.000$ for that level. Lastly, do not ignore the intercept term! Unlike the other model coefficients, it is not related to a specific rating variable. However, it still needs to be exponentiated to calculate the fitted value.

Look at the indicated factors (the exponentiated coefficients). Do they make intuitive sense considering the definition of the variables? Do they match your *a priori* expectations? For example, would an insured's premium be higher or lower if they were claims-free for 3 years than if they were claims free for 0 years?

5. Manually Calculate the Fitted Values

Although R produces fitted values, it's a useful exercise to manually calculate the fitted value yourself to deepen your understanding of the model. To calculate the fitted values with a log link function, you multiply the appropriate rating factors together for each record on the dataset (so, you will need to produce 80,000 fitted values for the training dataset). Mathematically:

Fitted Value = Intercept Factor * Gender Factor * … * Vehicle Age Factor

Recall what the fitted value represents. It is defined as:

$$\mu = e^{X_1\beta_1 + \dots X_n\beta_n} = e^{X_1\beta_1} * \dots * e^{X_n\beta_n}$$

This is the product of the indicated rating factors that were determined in section 4 above.

To map each record in the dataset to its rating factors, use the **vlookup** function in Excel. I have provided a simple example of calculating fitted values on the project Google drive.

To determine whether or not you have properly calculated the fitted values, use the following code in R (after you have run the model code):

```
mean(fitted(glm))
```

Compare the mean of the fitted values in your workbook to the mean shown in R. They should match exactly.

You can follow a similar process to calculate and check the fitted values for the linear model. Just remember that the linear model is additive and its estimates do not need to be exponentiated. The formula would just be

Fitted Value = Intercept + Gender Estimate + … + Vehicle Age Estimate

Take a quick glance down the columns. How do the linear model fitted values compare with the GLM fitted values?

6. Final Deliverables

The final deliverable for this task is an excel spreadsheet with three tabs, the first two being similar to the fitted values example I uploaded. On the first tab, you should have the model training dataset along with a GLM fitted value and a linear model fitted value for each record (to add calculations and additional tabs to the data set, you can create a copy of it by doing "save as…" an Excel workbook, since these things won't save properly in a .csv). On the second tab, you should have the individual model coefficients for each rating variable along with the corresponding indicated rating factors for the GLM (the exponentiated coefficients). On the third tab would be the model coefficients for the linear model. These are not easily translatable into rating factors, because the linear model does not allow for a multiplicative rating structure.

Good job you guys! At this point, you have run a GLM and produced the indicated rating factors – that is, the model's best estimate of how we should modify the insurance policy base rate for each of the different rating characteristics.

In Phase II, we will learn a few simple ways to improve and evaluate the model.

If you need me, you can email me and we will set up a time to talk. It might just be a quick phone call, or we could meet in person at Trinity. My next actuarial exam is on 10/30, so I will be around campus a lot in the weeks leading up to that, studying in the library. I'd be happy for any of you to come join me.