# Forecasting & Futurism

July 2014 – Issue 9

# **Big Data in** Life Insurance
## Does it exist? If so, how should we handle it?

*By Richard Xu and Dihui Lai*

Predictive modeling is a growing capability in the life insurance industry. There are more and more discussions about how applications of predictive modeling can be used to increase production or to efficiently manage risks. At the same time, the term "big data" is commonly used in public media and within the actuarial community.

### DO WE REALLY HAVE BIG DATA?

The expression "big data" is not consistently applied and can have different meanings in different situations. According to Wikipedia, big data is "a collection of data sets so large and complex that it becomes difficult to process using on-hand database management tools or traditional data processing applications." In principle, data should be considered "big" when it is close to a magnitude of billion gigabytes[1] (exabyte). Data sets of this size are typically found in areas such as genomics, climate science, astronomy and nervous system connectomics. Strictly applying this definition, big data in the life insurance domain is probably more of a marketing term than a reality. Current applications of predictive modeling for life insurance are predominantly based on structured data, which is readily available from existing internal systems (e.g., policy data, claim data). Even if we include many of the external data sources that are becoming available to life companies, we are still a long way from reaching the exabyte limit.

Even if data assets of life insurance companies are not as large as some other industries, this does not mean that existing data management tools or modeling technique are sufficient to handle the ever-expanding data sets. There are many business analytic projects involving huge amounts of data that result in a traditional approach being ineffective or even impossible. One example is a traditional life experience study where study output can easily reach tens of millions of records. Building a predictive model on data of that size can already prove to be challenging. So, practically speaking, the life insurance industry is indeed facing "big data" in that the data is big enough that it can no longer be effectively processed or analyzed using traditional methods.

## HOW CAN WE HANDLE BIG DATA?

Vague as the term "big data" is, the solutions to the challenges it creates can vary. Two major issues arise as a company's data volume increases: capacity and speed. Upgrading the hardware (memory and processors) can be a simple and inexpensive solution. If necessary to go beyond the limits of a desktop PC, a terminal server provides great memory capacity and has the advantage of incorporating multiple processors. This can be one possible solution for a reasonably large data set.

Cluster computing techniques are also relevant to the topic of big data analysis. This approach, including Massively Parallel Processing (MPP) and Hadoop system, partitions and processes data across a number of distinct but interconnected computing nodes. The final result is assembled once the individual bits and pieces are completed. MPP has a longer history than Hadoop and has the advantage of using SQL as its interface.[2] Hadoop, on the other hand, processes data in parallel using a MapReduce framework.[3] Although powerful, a cluster solution can be expensive to construct and maintain.

Other than attacking big data with these atomic tools, it is sometimes more efficient to solve memory or speed issues using better memory allocation techniques or algorithms. The R package *ff* provides a disk-stored data structure that can be accessed as if it were in RAM. Additionally, the R package *bigmemory* is especially good for dealing with large matrices of data. However, these types of packaged solutions are best for solving specific problems and might not be ideal for problems that go beyond the intended scope. Commercial software such as SAS or Revolution R provide a better ability to deal with large data sets and are generally better integrated with large data packages.

After all, problems dealing with big data are usually case-specific and solutions will depend greatly on the nature of the data set. In the following section we will demonstrate a real-world example of how big data can be approached.

## CASE STUDY: BUILDING A GLM USING BIG DATA

The generalized linear model (GLM) is widely accepted as an efficient tool in insurance analytics. The standard *glm* function provided in R meets most of the everyday demands and applications of GLM. However, the *glm* function becomes less efficient when faced with big data. The fitting procedure can be very slow on a data set of several million records. Additionally, the calculation process might not even complete on a regular desktop PC due to memory overflow.

For this case study, we used a 5.64 million record data set that was initially used for industry post-level term lapse study and demonstrate the efficiency of the available GLM routines from three different R packages (*primary R, biglm* and *RevoScaleR*[4]). The lapse model we tested consists of 16 independent parameters and assumes that lapses follow a Poisson distribution.

All functions tested for this experiment (summarized in Table 1) finished the job in a reasonable amount of time. The *glm* function required about 2 GB memory and four minutes to finish, while directly calling the *glm.fit* function shortened the procedure significantly. In comparison to the built-in *glm* function, *bigglm* is much more economical in terms of memory allocation, while the routine requires a comparable amount of time to finish. The *rxGlm* function is excellent in speed (finishing the procedure in less than a minute), yet this function requires only slightly more memory than *bigglm*. In summary, the built-in *glm* function is flexible and easy to use, but not ideal for big data. The *bigglm* function is excellent in memory efficiency, but *rxGlm* is superior in computing speed.

CONTINUED ON **PAGE 24**

BIG DATA … | FROM PAGE 23

| Function | Elapsed Time (s) | Memory* (Mb) |
|---|---|---|
| **glm** | 185 | 2408 |
| **glm.fit** | 78.3 | 1056 |
| **bigglm** | 209 | 2.3 |
| **rxGlm** | 28.8 | 43.5 |

Table 1: Comparison of GLM Function Using Different R Packages. The CPU time is evaluated using the built-in function proc.time in R and the memory usage is evaluated using a wrapped-up gc function. The model is run on a PC desktop (Intel core i7-3770 CPU 3.4GHz and 12 GB).

Building a successful model requires construction of multiple models and then selecting the best among the candidates. The procedure can be computationally intense and time-consuming. Optimizing the model selection procedure would be beneficial to modelers. By default and without any add-on packages, R only uses one core for processing. Parallel computing packages such as *multicore*, *snowfall* can take advantage of multi-core features and speed up the model selection tasks. For the case study, we tested the *snowfall* package to demonstrate the power of parallel computing. The same data set was used for this test. Six variables (in other words, six models) are tested for significance.

The results showed that the built-in *glm* function failed to complete due to memory error. The *bigglm* function finished the routine in 16 minutes, with <10 MB of memory usage while parallelizing the procedure reduces the time by half. The usage of *snowfall* only reduces the procedure by about 10 seconds when *rxGlm* is used as the core function. Overall, parallelism can speed up the model selection procedure but can put some stress on the memory demands.

| Approach | Elapsed Time (s) | Memory (Mb) |
|---|---|---|
| lapply*+glm | Memory overflow | |
| snowfall+glm | Memory overflow | |
| lapply+bigglm | 1004 | <10 |
| snowfall+bigglm | 497 | <10 |
| lapply+rxGlm | 54.4 | 44.4 |
| snowfall+rxGlm | 43.2 | 204 |

Table 2: Comparing the Serial Approach and Parallel Approach. The evaluation of CPU time and memory is the same as what is described in Table 1. *lapply is a built-in R function that enables the process of a list of models serially.

## CONCLUSION

Big data is no doubt a big topic in the world of insurance and will become even bigger in the future. Tools are available to help us, but we must be careful in making our decision. Depending on the nature of projects and data attributes, the optimal solution can vary. In the case study presented, we see that Revolution R provides the best solution if speed is the priority, while *biglm* should be considered if memory is of greater concern. Big data is on its way and will no doubt present challenges. To be successful, companies need to prepare.

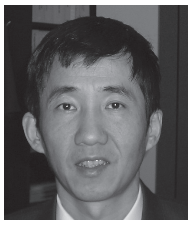
## REFERENCE

[1] Francis, Matthew. 2012. Future Telescope Array Drives Development of Exabyte Processing. *Ars Technica*, April 2.

[2] *Massively Parallel Processing* (DW)—*A Technical Reference Guide for Designing Mission-Crtical DW Solutions*, http://technet.microsoft.com/en-us/library/hh393582.aspx

[3] Borthakur, Dhruba. 2007. The Hadoop Distributed File System: Architecture and Design.

[4] Rickert, Joseph B. 2011. The RevoScaleR Data Step White Paper.



*Richard Xu*

**Richard Xu,** FSA, Ph.D., is senior data scientist at RGA Reinsurance Company in Chesterfield, Mo. He can be reached at *rxu@rgare.com.*

*Dihui Lai*

**Dihui Lai,** Ph.D., is data scientist analyst at RGA Reinsurance Company in Chesterfield, Mo. He can be reached at *dlai@rgare.com.*