Case Study Part 2: Improving Financial Projections for Long-Term Care Insurance with Predictive Analytics

By Missy Gordon and Joe Long

Developing accurate financial projections of long-term care (LTC) insurance is easy—if you have a crystal ball. For those without one, it's no small feat! In this article, the second in our series on LTC projections and predictive analytics, we dive deeper into how predictive analytics can be used to help overcome some of the challenges. Our discussion includes how predictive analytics can help determine the amount of credibility we should give the historical experience, as well as how it can help navigate the complex interactions that underlie this experience.

In our first article[1] we set the stage by discussing the importance of giving the "right" amount of weight to a company's experience when adjusting an industry benchmark in order to produce a projection assumption that generalizes well to future data. We then introduced the bias-variance trade-off, a concept in predictive analytics that highlights the importance when developing a model of not overreacting or underreacting to the data (i.e., choosing the "right" amount of data weight). We discussed how the traditional "actual-to-expected" or "A:E" study goes about doing this by using credibility weighting to adjust a benchmark. This typically includes a judgment-based decision in assigning the credibility of the data—for example, choosing 271 or 1,082 events as fully credible in limited fluctuation credibility. The American Academy of Actuaries does a great job of further discussing the intricacies of applying this and various other credibility methods to LTC experience in their Long-term Care Credibility Monograph.[2]

After setting the stage with the traditional A:E approach, we then discussed how predictive analytics can be used to remove this judgment-based decision of determining data credibility through techniques that focus on balancing the bias-variance trade-off in an automated fashion. To illustrate this we introduced the penalized generalized linear model (GLM), which can automatically traverse the bias-variance trade-off by testing a range of penalties to determine the "right" amount of weight to give to company data versus an industry benchmark. This ability to test the credibility of the experience in a scientific manner is one of the great benefits of predictive modeling. Hugh Miller discusses this and provides additional benefits in his paper that links the penalized GLM approach to an actuarial credibility approach.[3]

Before jumping into the results of the case study there are a few more important items we would like to discuss in this article to further set the stage. Understanding how to automate the process of finding the "right" amount of weight when using a penalized GLM is an important concept. We will add detail on how to do this using a handy trick from the machine learning realm known as a *k*-fold cross-validation (CV), which helps us select the penalty. We will also introduce the gradient boosting machine (GBM) algorithm. GBMs are another predictive modeling technique that can take the automation one

step further by creating interactions among the variables in the model with little user input. Without this automation the process would otherwise consist of challenging judgment-based decisions.

To wrap things up we will close with a discussion on important items to consider when using one of these techniques, as there is no silver bullet when it comes to developing assumptions using predictive analytics. Depending on the intended use, you may find yourself utilizing simpler techniques or an approach that combines multiple techniques.

DETERMINING DATA CREDIBILITY

In the prior article we discussed how the penalized GLM automatically traverses the bias-variance trade-off. However, we did not look closely at how one selects the penalty that determines the amount of credibility or weight given the data. The most common method for selecting the penalty to use in a penalized GLM is through a technique known as the *k*-fold CV. As you advance in your journey into using predictive analytics you will come across this technique more often than not, as it is frequently used in the machine learning realm to assess how a model might perform on future data independent from its construction. Modelers use this technique a lot because it's simple. This technique can be used across a variety of predictive modeling algorithms because it directly estimates expected model performance by testing the model on data that wasn't used to train the model (an out-of-sample test). This is in contrast to classical statistical tests of fit that typically rely on methods to adjust the test of fit that was calculated on data used to train the model (an in-sample test).

To conduct a *k*-fold CV, the algorithm randomly partitions the data into k equal-sized subsets and then iteratively trains and tests the model independently on each subset of the data. Each time the model is trained, it uses only k – 1 subsets of the data. The remaining kth subset is then used to test the performance of the model on unseen data (e.g., data that wasn't used to train the model in developing its predictions). A typical performance metric used is the mean squared error (MSE), which is the average of the squared difference between the actual and predicted value. Once the performance has been tested on each unseen *k* subset, we then average the performance to produce a single average expected performance metric.

This process gives an estimation of how well a model might generalize to new experience. Using such a technique allows us to use all the data we have for testing, which is important in cases where you cannot afford to withhold data to test the model. Figure 1 shows an illustration of how a 3-fold CV would be performed.

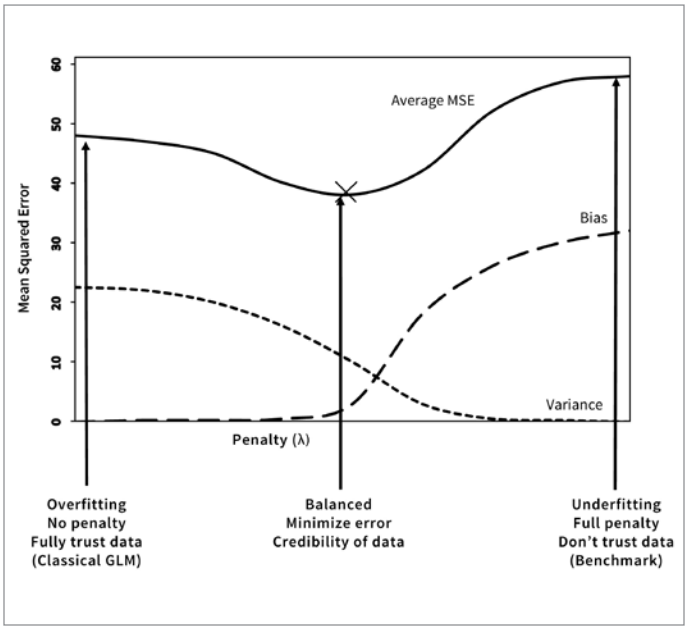
Figure 1
3-Fold CV Performance

3-Fold	Test 1	Test 2	Test 3	MSE on holdout data	Average
1 33%	1 Holdout	1 Use	1 Use		Teat 1
2 33%	2 Use	2 Holdout	2 Use		Test 2
3 33%	3 Use	3 Use	3 Holdout		Test 3
Calibration data	100%	100%	100%		

Returning to our example of using a k-fold CV to select the penalty for a penalized GLM, we typically test 100 penalties that range from no penalty (data has full weight) to a high penalty (data has no weight and uses only the benchmark). We then compare the average performance each penalty produces when tested on the unseen data to select the penalty that gives the "right" amount of weight to our company experience. This can be done by selecting the penalty that has the best performance (lowest error) produced by the k-fold CV. Figure 2 provides an example of this and also shows how this process balances the bias-variance trade-off to help us determine the "right" weight to give the company data.

In Figure 1 we showed an example of a 3-fold CV, but using 10 to 20 folds is typical. Therefore, when a range of 100 penalties

Figure 2
Identifying the Penalty with the Best Performance



is tested, we are training 1,000 to 2,000 models and testing the prediction error with a few lines of code to assess which penalty will give us the "right" amount of data weight to minimize prediction error. This robust process is in contrast to the typically judgment-based decision in a traditional A:E study.

NAVIGATING COMPLEX INTERACTIONS

LTC projection assumptions have complex interactions. For instance, claim termination rates vary significantly by age and duration. Often ages and durations are banded to increase credibility, which raises several questions: which are the right ages to band, which are the right durations to band, and are the duration bands the same for each age band? With a traditional A:E study or even a GLM, these decisions must be incorporated into the structure of the model. Such decisions can be tough to make and are usually based on analyzing high-level slices of data, which can be manually intensive to navigate.

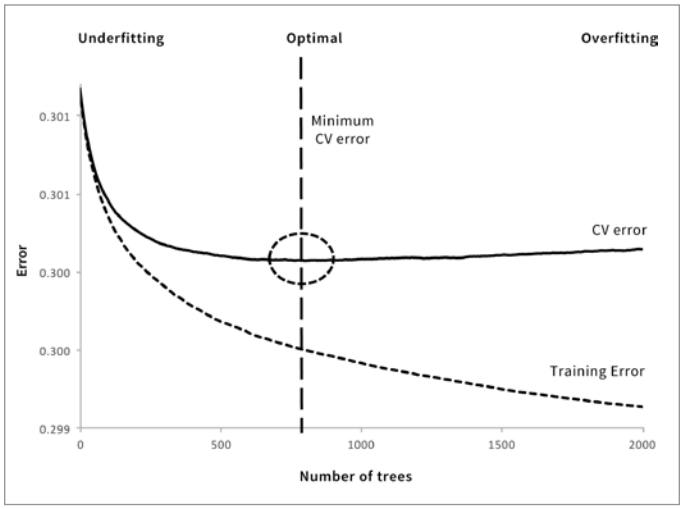
A GBM doesn't have a fixed structure like a GLM. It is a flexible, nonparametric algorithm that typically uses an ensemble of decision trees to develop predictions. This automatically creates key interactions of the independent variables in the model. At each decision point in the trees, the model cycles through each variable and chooses where to slice it to make a decision of the optimal data split that minimizes the prediction error. This process determines variable importance and how to slice variables such that the model has the ability to navigate complex interactions in an automated fashion.

Using this state-of-the-art predictive modeling technique, one can replace most of the traditionally judgment-based decisions of this type of analysis with a more statistically robust and reproducible process. Similar to a penalized GLM, a GBM automates the decision of how much weight to give the historical experience versus the benchmark (i.e., the amount of data credibility). However, it also takes the automation a step further by determining what key interactions of variables should be used to adjust the benchmark. While the GBM automatically develops the interactions, it is critical that the resulting relationship be reviewed by an experienced actuary for reasonableness. If the relationships are not making sense, then additional feature engineering may be needed or it might be that a GBM isn't the solution for a particular problem.

A GBM model includes a number of inputs that control the model's complexity and its learning process (i.e., hyperparameters). These hyperparameters are similar to the penalty in the penalized GLM, in that they are used to help balance the bias-variance trade-off. Just like with the penalized GLM, a standard approach for tuning such hyperparameters is to use a k-fold CV. However, due to the increased number of hyperparameters to consider, this tuning process is more ambiguous than tuning the penalty in the penalized GLM. As such, experienced practitioners will have different approaches for tuning the hyperparameters in a GBM.

In general, if the hyperparameters of a GBM are tuned properly, the final set of hyperparameters should produce a model such that there is little change in the k-fold CV performance metric around the last few hundred or so trees used in the model. The graph in Figure 3 shows this result, where the error around the location of the minimum k-fold CV is relatively flat when more or fewer trees are added to the model, as shown by the red circle in the graph. This produces a more stable model, which gives a wider safety net that guards against overfitting or underfitting. In practice, after reviewing this graphed output, one might tune the hyperparameter more, such that the green CV error line flattens out, making this range larger.

Figure 3
Tuning Hyperparameters with CV Performance



When trained properly, a GBM helps remove most of the judgment-based decisions from the traditional process. However, a shortcoming of a GBM is that it does not extrapolate where there is limited or no experience. As with traditional methods, judgment is necessary when extrapolating results based on limited to no historical experience.

GLEANING INFORMATION FROM A GBM

A single decision tree is easy to look at to see what is driving the predictions. It provides a nice map of yes/no questions one can follow to see the path taken to arrive at the final predictions. However, a GBM model typically contains hundreds to thousands of trees in it, making an exploration of the trees a daunting if not impossible process.

Luckily there are some nice tricks to gleaning information on what is driving the predictions in a GBM model. The simplest

is by looking at the variable importance measure, which identifies how useful a variable is at reducing the prediction error when training a GBM model. When using the GBM to adjust a benchmark, this variable importance can then be used as a measure to see what key variables were driving the most change in the benchmark used.

The GBM model also doesn't provide the nicely formatted factor adjustments of a traditional A:E study or a penalized GLM. Instead, the model creates a prediction by summing up thousands of predictions across all the trees in the model. We can get an idea of the marginal effect a variable has on the outcome, similar to how one interprets the coefficients in a GLM model, by using what is called a partial dependence plot. Through such an analysis we can explore the impact each variable has on the assumption and assess whether the relationships are reasonable.

IMPLEMENTATION CONSIDERATIONS

When developing a new assumption it is very important at the start of the project to consider if your company has any implementation constraints. For example, a projection system may not have the ability to accept new variables, or it may be necessary to have the adjustments formatted in a specific way for management to review. As discussed in the previous section, a GBM doesn't produce nicely formatted adjustments like a traditional A:E study or a penalized GLM model does. If a specific format is needed, a penalized GLM might be the best approach. In such situations, we tend to use a GBM to help us explore the data by looking at the variable importance measures and partial dependence plots to find the key variables and relationships driving the change in the experience. We then use those findings to help us construct penalized GLM models.

Another alternative is to output a new updated assumption on a seriatim basis. Or perhaps if the number of variables in a model is not too large, you can output every combination of variables in the GBM model such that you can format it into standard tables that your projection system might already be set up to accept.

PUTTING ALL THE PIECES TOGETHER

We have discussed the importance of the bias-variance trade-off, introduced two popular predictive analytics techniques, and considered when you might reach for one over the other. In our next article, we will discuss a case study of how we used such techniques to develop LTC claim termination projection assumptions. ■

Missy Gordon, FSA, MAAA, is a principal and consulting actuary at Milliman. She can be reached at *missy.gordon@milliman.com.*

Joe Long is an assistant actuary and data scientist at Milliman. He can be reached at *joe.long@milliman.com.*

ENDNOTES

1 Published in the December 2017 issue of *Long-Term Care News.*

2 American Academy of Actuaries (August 2016). Long-Term Care Credibility Monograph. Retrieved Jan. 31, 2018, from *http://actuary.org/files/imce/LTC_Credibility_Monograph_08172016.pdf.*

3 Miller, H. (August 2015). A Discussion on Credibility and Penalised Regression, With Implications for Actuarial Work. Institute of Actuaries of Australia. Retrieved Jan. 31, 2018, from *https://actuaries.asn.au/Library/Events/ASTINAFIRERMColloquium/2015/MillerCredibiliyPaper.pdf.*

4 Another common choice is to select the penalty that is one standard error away from the minimum *k*-fold CV error.

5 Figure 2 was adapted from Figure 6.5 on page 218 of the textbook *An Introduction to Statistical Learning.*