Authors
Aubrey Clayton
Director — ERS Insurance Research

Contact Us

Americas
+1.212.553.1658
clientservices@moodys.com

Europe
+44.20.7772.5454
clientservices.emea@moodys.com

Asia (Excluding Japan)
+85 2 2916 1121
clientservices.asia@moodys.com

Japan
+81 3 5408 4100
clientservices.japan@moodys.com

Deep-Learning the Cash Flow Model

Abstract

Previous research notes have demonstrated the successful application of proxy function techniques for estimating values and reserves/capital for insurance liabilities. For the most part, these approaches require a time horizon and statistic of interest (such as the mean or CTE) to be specified ahead of time. Extending the techniques beyond a single point in time creates a problem of path dependence, meaning the function fitting takes place over a potentially high-dimensional space of risk variables.

In this paper, we consider the application of more recent "deep learning" techniques to these problems. We develop a proxy for the insurance liability cash flow model itself, considered as a rule that associates a time series of cash flows to a series of risk variables. The machine learning algorithm we use (LSTM) is particularly adept at handling this sort of problem structure, and we can train proxy functions to reproduce cash flows with a high degree of accuracy.

We demonstrate the power of this approach using two examples involving a portfolio of complex variable annuities: cash flow testing and projected liability valuation in a business planning scenario. Proxy results are seen to compare favorably with "brute force" Monte Carlo simulation, at a small fraction of the computational cost.

Table of Contents

1. Introduction

Proxy function techniques such as Least Squares Monte Carlo (LSMC) have proved to be useful in a wide variety of calculation-intensive problems in the insurance world over the last decade. They are a regular fixture for firms using "internal models" in the calculation of capital requirements under Solvency II, as this ordinarily requires a prohibitively large number of Monte Carlo scenarios to accomplish. Previous research has successfully extended the same concepts to other realms, for example dynamic hedging [1, 2] and projection of CTE-based reserve and capital requirements [3, 4, 5, 6].

The common feature of these problems, which allows for the extension of proxy function methods, is their nested-stochastic structure, typically defined by a set of "inner" scenarios branching off from an "outer" scenario at a given future time. The principal difference from application to application is whether the inner scenarios are drawn from a risk-neutral or real-world probability distribution, from which the desired statistical metric is derived. For risk-neutral distributions applied to valuation problems, this metric is usually an average of discounted cash flows. For real-world distributions in the setting of reserve/capital requirements, the relevant statistic is usually a conditional tail expectation of accumulated deficits, but the outer/inner structure is broadly similar. Figure 1 shows the schematic similarity between the two nested-stochastic frameworks.

The LSMC approach consists of extracting a functional relationship between the desired metric and the underlying risk variables at the given time horizon using a small number of inner scenarios. These scenarios are passed through a liability cash flow model to produce (deliberately) inaccurate estimates of the desired statistic, which are then used to train the function by polynomial regression or other means.

Figure 1   Example nested-stochastic problems
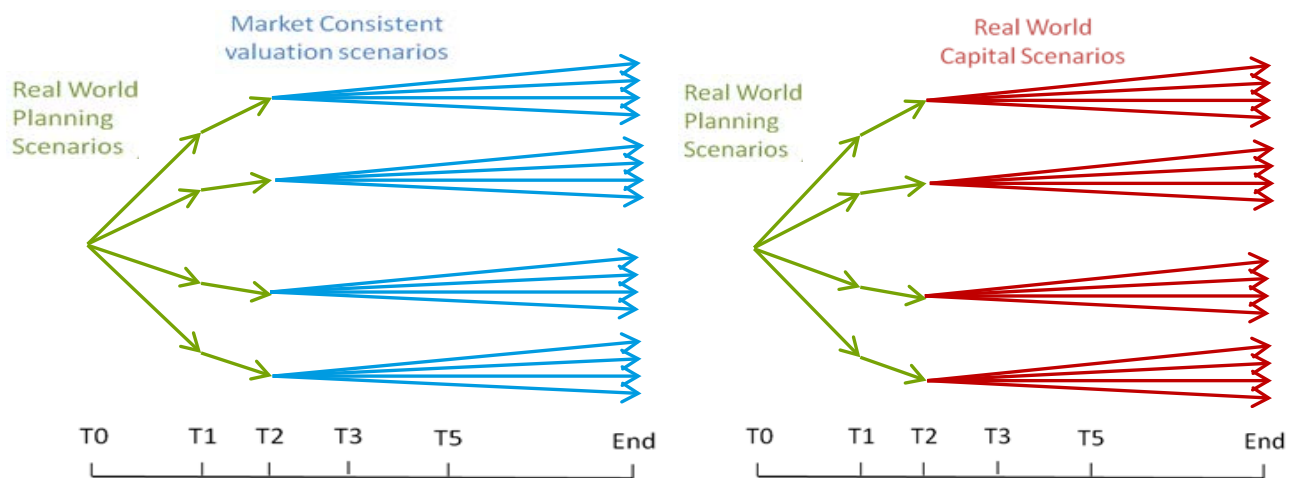
In this paper we consider an entirely new approach to training proxy functions by learning the cash flow model from the ground up. We are particularly motivated to do so because of the difficulty of using the previous methods to develop functions suitable for multiple future times. For example, each of the following questions related to financial planning, cash flow testing, and asset-liability management (ALM) can require multi-timestep analysis:

  » What will the realized cash flows of a book of assets and liabilities be over a given planning scenario?
  » How might the liability value change through time, and what effect will this have on asset management decisions?
 » What will balance sheets look like, including stochastic reserve/capital requirements?

As these types of analyses become more standard under principle-based reserving ("PBR") and risk-based capital ("RBC") regimes, the emerging need for fast calculation beyond a single time horizon has become clear.

With multiple timesteps, however, comes the problem of path dependence. That is, any statistic of interest at a future time along an outer "path" may depend on the entire path up that point, which naturally increases the number of dimensions of risk variables the proxy function might, in principle, depend on. In certain contexts, some ad hoc strategies for handling path dependence have

performed well. For example, in [2] we describe an approach for "collapsing" the path into a small number of summary risk variables (e.g., current fund value, asset allocation, time since last guarantee reset), specific to each policy in the liability portfolio, and fitting proxy functions at the policy level. In [6], we recommend restricting the paths themselves to a small dimensional subset of all possible paths, based on the "what if" scenarios that are of actual interest for business planning purposes. But each of these carries a limitation: policy-level analysis can only apply to metrics (such as liability value or Greeks) that behave additively when moving from policy to portfolio, so the approach will not work for sub-additive metrics such as VaR or CTE, and constraining the paths to a small number of degrees of freedom may not be fully realistic and would exclude any stochastic analysis.

The technique we describe here is more all-purpose. Instead of tailoring a proxy function to a particular statistic corresponding to a particular probability distribution in a nested-stochastic setup, we consider the problem of learning the logic of the cash flow model itself. That is, if we consider, say, an insurance liability model as a rule to turn any "input" economic scenario into an "output" series of cash flows, we can attempt to learn this rule and replace the cash flow model by a proxy. If successful, this would simultaneously allow for any of the foregoing analysis, since the proxy model could be applied to any combination of planning, real-world, or risk-neutral scenarios. Figure 2 illustrates how the same proxy model could be used to project cash flows under hybrid risk-neutral/financial planning scenarios (as needed for projecting realized cash flows and market-consistent balance sheets) and under risk-neutral/real-world scenarios (as needed for stochastic projections and run-off capital requirements including embedded dynamic hedging).

Figure 2   A proxy model for cash flows applied to different combinations of scenario types
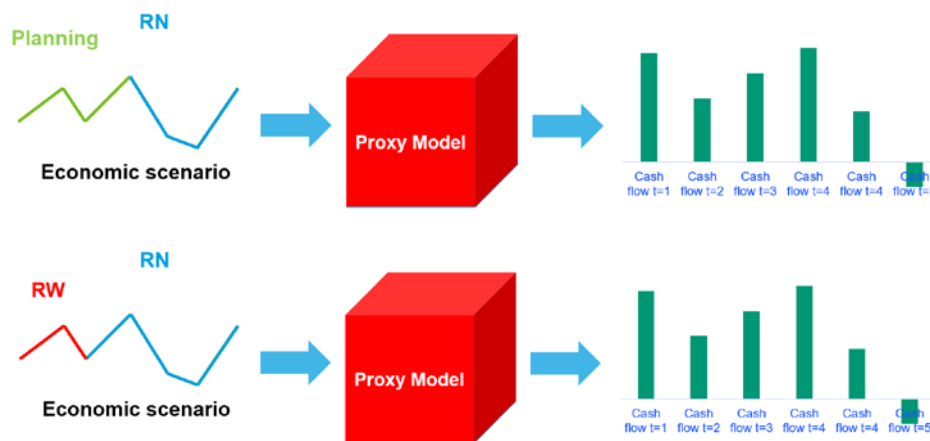
With a proxy function in place, the recalculation of these cash flows is nearly instantaneous, so even computations requiring billions or trillions of scenarios over multiple layers of nesting can be accomplished, to within a certain degree of approximation depending on the quality of the proxy fit.

The fitting technique must therefore also be more sophisticated, since the number of dimensions inherent in the scenarios is now as great as the number of timesteps multiplied by the number of risk variables.

2. Deep Learning

In recent years new "deep learning" methods have been developed for handling difficult, high-dimensional machine learning problems in such diverse fields as image recognition, machine translation, speech/handwriting parsing, board games, bioinformatics, and credit fraud prevention. Of particular importance for problems involving time-series data is the machine learning structure known as a recurrent neural network, which differs from a "feedforward" network in that connections between nodes may form (directed) cycles, allowing for temporal dynamic behavior (see [7] for a summary). Intuitively speaking, the key feature of this network structure is that looping the network back on itself allows the machine's guess for the previous timestep(s) to inform its guess for the next timestep. This allows for sequential memory effects, so that, for example, an algorithm parsing handwriting may remember that the previous letter was likely "Q" and use that knowledge to predict the next letter will be "U."

Vast neural networks and recurrent models may chain many layers and networks together. The usual process of training via backpropagation through such a network therefore involves multiplying many fractional values, creating the so-called "vanishing gradient" problem. To address this, Hochreiter and Schmidhuber [8] proposed a network structure known as Long Short-term Memory (LSTM), which includes memory cells that may store values over arbitrary time lengths, regulated by various "gates" that control the flow of information into and out of the cell. Such networks have since been used as components in consumer products developed by tech companies including Google, Apple, Microsoft, and Amazon.

Our problem of learning an insurance cash flow model has features suggesting the same learning techniques could apply here. Cash flows are time-sensitive, and the predicted cash flow at a given time should certainly affect predictions for subsequent cash flows. Also, features of the "early" part of an economic scenario, say a sudden market crash or global pandemic toward the beginning of a scenario, should continue to affect cash flows much later on, as it may set guarantee levels that persist through time or determine policyholder characteristics via mortality and lapses. Instead of specifying these features explicitly, though, we allow the LSTM network to learn what scenario information is relevant to store in memory. Figure 3 illustrates the network structure. The fitting algorithm is essentially backpropagation with a more complicated network architecture and model parameters numbering in the thousands. For the purposes of this research, we train the network using the Google TensorFlow library with Keras API.

Figure 3   Network structure for learning the insurance cash flow model
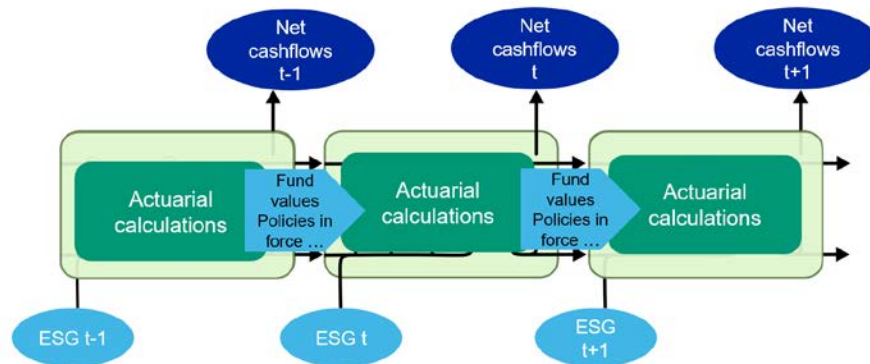
Since the learning problem we are addressing is no longer nested-stochastic, no allocation of outer/inner scenarios is required. Instead, we divide up scenarios into a training set and an out-of-sample validation set, produce a complete time series of cash flows for each using the full liability model, fit the network on the training data, and compare model predictions against actual cash flows on the validation set.

3. Example: Cash Flow Testing

In the first of our example applications, we consider a cash flow testing problem. Our goal is to project total liability cash flows under various planning scenarios for yield curves and equity returns over next 20 quarters. For this and the following exercise, we use an example block of liabilities consisting of approximately 75,000 variable annuity policies with a heterogeneous mix of accumulation (GMAB), withdrawal (GMWB), and death benefit (GMDB) guarantees at various levels. The policyholder characteristics (issue age, policy anniversary dates, etc.) are likewise realistically mixed. For the economic risk scenarios, we include initial market conditions followed by parallel yield curve movement and constant equity return. Specifically, we allow planning scenarios with the following degrees of freedom:

» Change to the level and slope of the yield curve over the first quarter (described by the first two principal component shocks)

» A parallel shift to the yield curve over the remainder of the next 20 quarters, defined by the size of the shift and the period over which the shift takes place

» The change in the U.S. equity index over the first quarter

» The subsequent returns of the equity index, assumed to be level for the remainder of the 20-quarter projection

In total then, we have six risk factors (four for yield curves and two for equities), any combination of which fully determines a planning scenario. Figure 4 illustrates one such path. These are materially identical to the scenarios used in [6], except that for the current work we fit to cash flows along the entire path.

Figure 4   Possible yield curve and equity evolution
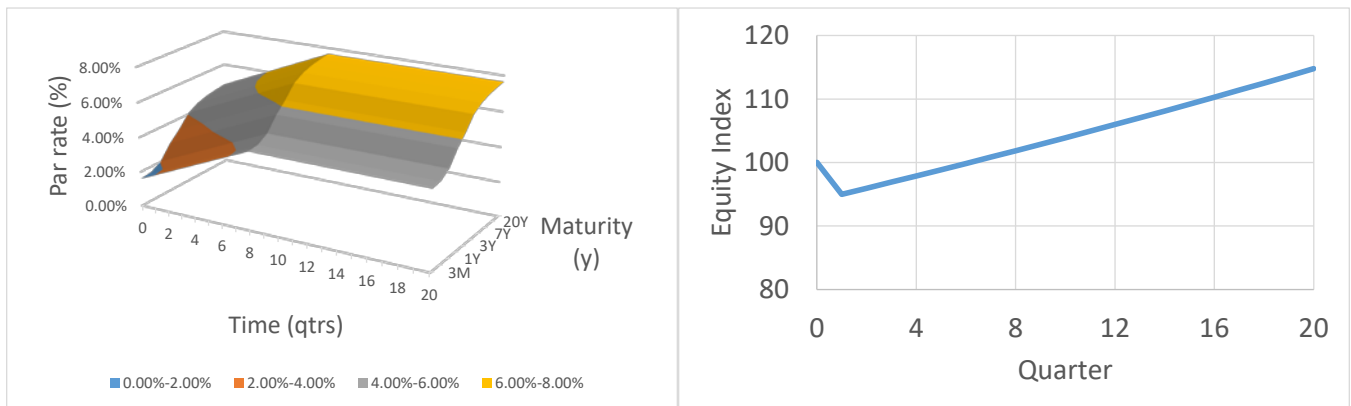
By focusing on realized cash flows rather than a terminal statistic such as CTE reserve, we have set ourselves a much more difficult learning problem. For example, in a scenario with a prolonged equities slump, the policy guarantees of different maturities cause the realized cash flows to exhibit a jagged behavior, shown in Figure 5. Decomposing the total cash flow into its component parts reveals that almost all the jaggedness is due to these benefits being realized, which depends on the guarantees of the policies currently in-force. Additionally, the functional relationship between any one of these cash flows and the underlying risk variables may be unsmooth, as varying account values lead to payoffs with the form of an equity put-option.

Figure 5   Realized liability cash flows during a prolonged equity market downturn
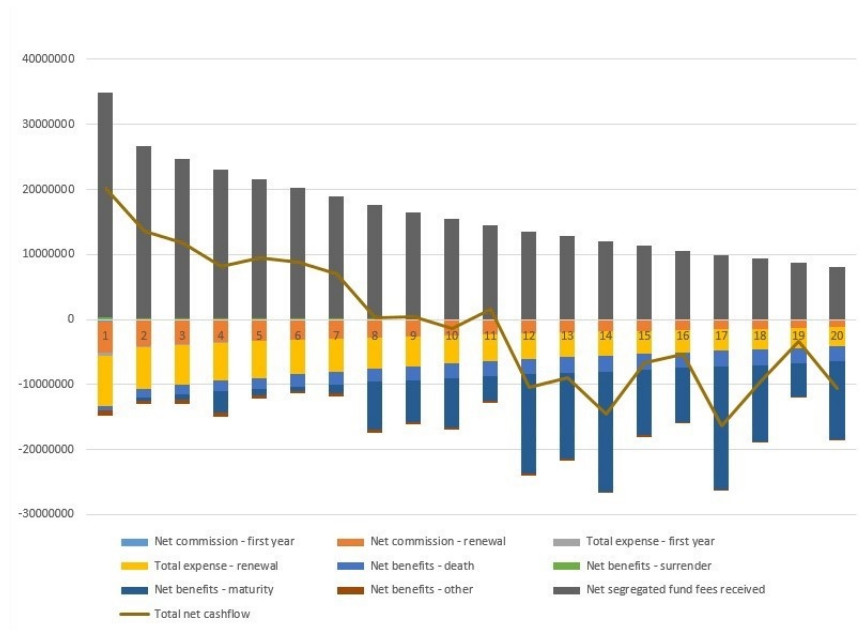
We use 90,000 combinations of economic risk variables for training and reserve 10,000 scenarios for validation. The resultant proxy model reproduces even the complex cash flow behavior shown above. For example, Figure 6 compares a single scenario of cash flows produced by the full liability model with those predicted by the proxy function.

Figure 6   Actual and predicted cash flows for a representative validation scenario showing complex cash flow behavior
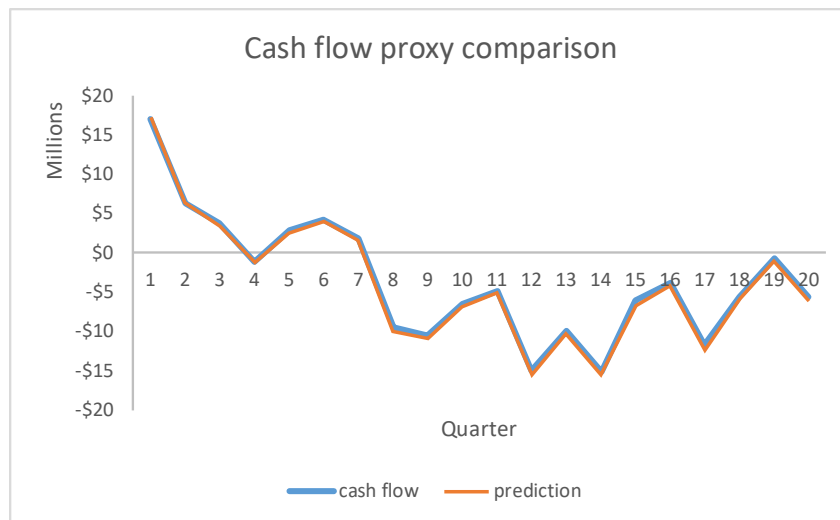
A scatterplot of all proxy vs. actual cash flows shows generally close agreement.

Figure 7   Proxy vs. actual cash flows, all validation scenarios and all timesteps
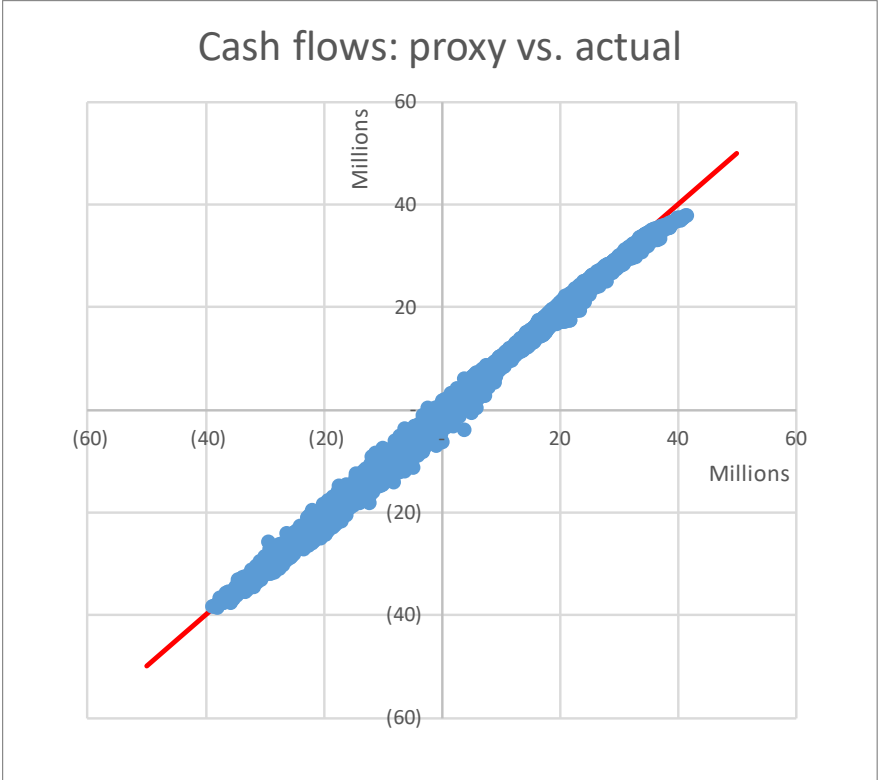
4. Example: Valuation

For our second case study, we consider a problem of risk-neutral valuation. Our goal is to project total liability cash flows under risk-neutral scenarios for interest rates and equity returns over the next 50 years, and then to use the cash flow proxy to estimate liability value as the average sum of discounted cash flows. This setup substantially complicates the machine learning process compared to the previous example. While the cash flows under the previous planning scenarios may have been complex, the scenarios themselves were confined to a six-dimensional space. In principle, we could have isolated the proxy fitting process to an individual timestep and used more conventional techniques (although polynomial regression would still have struggled, because cash flows are not smoothly dependent on risk variables, as described above).

The risk-neutral ESG we use has a single factor (Hull-White) model for interest rates and a constant volatility (Black-Scholes) model for equity returns, and we run the model for 50 annual timesteps. Therefore, the paths we generate have dimension 100, compared to the previous six.

However, we find that the LSTM proxy function can still replicate the cash flow model well even under these conditions. Aggregating discounted cash flow values into a liability present-value (PV) by scenario and comparing actual vs. proxy model shows good overall agreement. The out-of-sample R-squared for this fit is approximately 99.83%.

Figure 8   Proxy vs. actual liability PV for risk-neutral scenarios
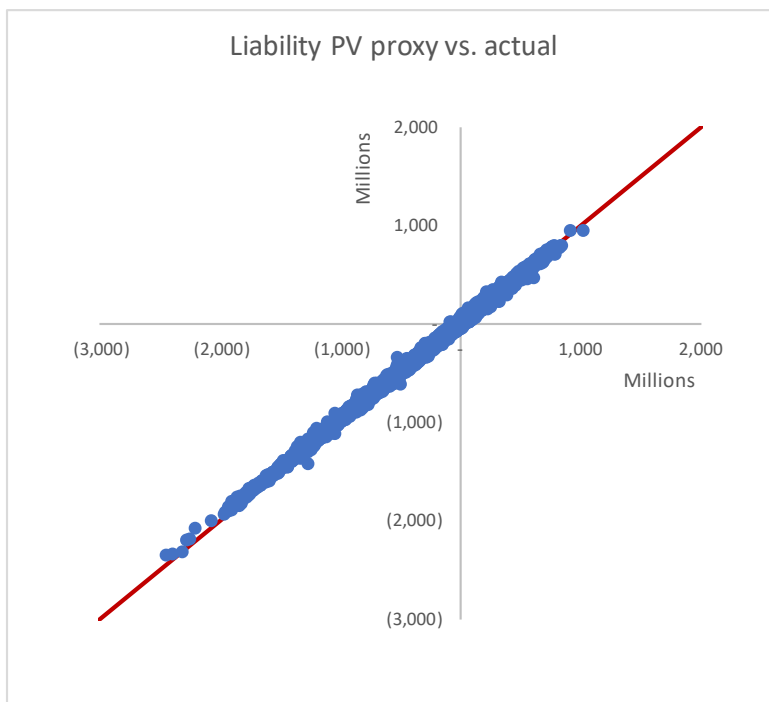
After averaging the liability PV over the validation scenarios, we find that what little residual error remains is averaged out, and the resulting estimated liability value has an error of only about 0.30%.

Figure 9   Actual and proxy liability value, average PV
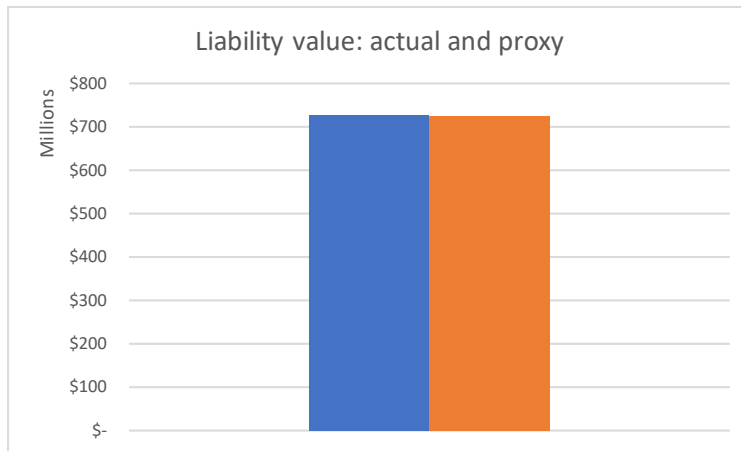
Finally, we can illustrate the true power of the cash flow proxy approach by combining the valuation methods of this example with the projections of the previous one. For instance, now that we have an all-purpose replacement for the liability cash flow model in the form of a proxy, we can ask the function for cash flows under 10,000 risk-neutral scenarios branching out from a single outer path at time 1, rather than branching from time 0. Each scenario has the same form as the risk-neutral scenarios used above, except that they all share risk factor values for the first year. But because the proxy function was not designed to be used only at a particular time horizon, it can be applied just as well to purely risk-neutral scenarios or hybrid planning/risk-neutral scenarios, and so on.

The same approach allows us to compute a projected liability value at any time, simply by arranging for the scenarios to have common values up to that time and discounting the cash flows appropriately. Figure 10 shows the projected liability values for every year up to time 5 in a sustained equity downturn scenario of -10% return per year. We compare proxy against actual model results calculated using "brute force" Monte Carlo. The largest relative error is around 6%, at time 1.

Figure 10   Projected actual and proxy liability values for t = 1, …, 5, average PV
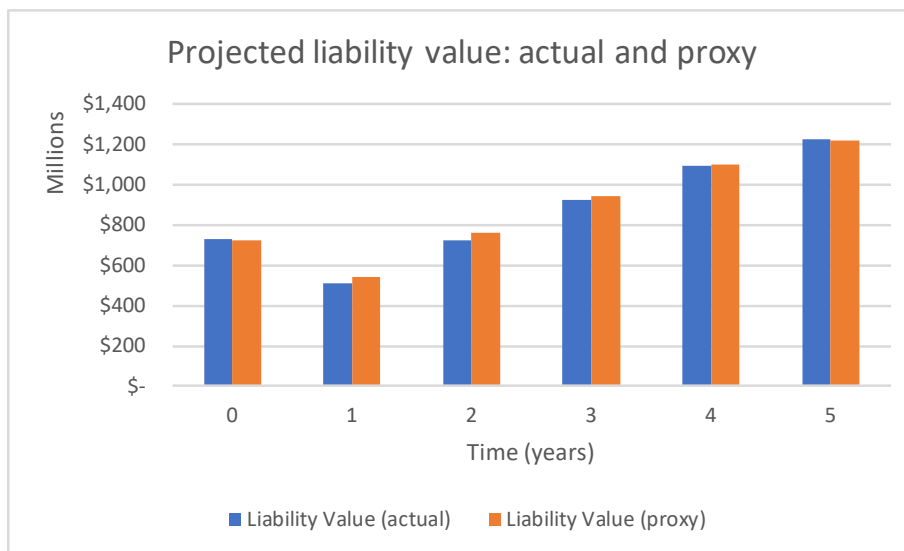
These projections, together with the realized cash flows from the previous example, could give a reasonably accurate and complete picture of the firm's projected income and balance sheets over the planning scenario. Other planning scenarios could be similarly analyzed with no additional overhead in fitting new functions or generating new scenarios. Once the proxy function is calibrated, it applies to any combination of risk variables. To generate these by Monte Carlo methods would require an additional, costly run each time.

5. Conclusions

The calculation problems required for business planning, cash flow testing, and ALM are quickly outpacing the ability of previous proxy function techniques to keep up. The problems are made especially difficult by the degree of path dependence they exhibit, which creates a problem of high-dimensionality for traditional function fitting techniques. For some isolated examples, the strategies we have developed previously can address path dependency well up to a point, but there is a clear need for a more all-purpose solution.

Here, we have demonstrated a technique for learning the logic of the cash flow model itself, thought of as a rule for translating scenarios into a time series of cash flows. This has the effect of introducing even more dimensions into the problem, since we are now considering a full series of liability cash flows as a function of the full scenario path, rather than fitting to a statistic of the future cash flow distribution as a function of a path up to a given time. Counterintuitively, by making the problem harder we have made it easier, since we can now make use of sophisticated machine learning algorithms developed in recent years and shown to be successful for problems much more high-dimensional than ours. In particular, the LSTM model structure seems well-suited to the task of projecting insurance liability cash flows.

With one cash flow proxy, we can replace the liability model altogether and project realized cash flows, liability values, and potentially much more. Since the proxy model does not depend on any choice of time horizon or probability distribution, it is suitable for a wide range of applications. The more demands for multi-period analyses placed on the firm by regulators or senior management, the more the proxy function will show its usefulness, and the greater the savings it will offer in time and money.

References

[1] Clayton, Aubrey, Steven Morrison, Craig Turnbull, and Naglis Vysniauskas, "Proxy functions for the projection of Variable Annuity Greeks." Moody's Analytics, 2013.

[2] Clayton, Aubrey, and Steven Morrison, "Proxy Methods for Hedge Projection: Two Variable Annuity Case Studies." Moody's Analytics, 2016.

[3] Morrison, Steven, Laura Tadrowski, and Craig Turnbull, "One-year projection of run-off conditional tail expectation (CTE) reserves." Moody's Analytics, 2013.

[4] Morrison, Steven, Craig Turnbull, and Naglis Vysniauskas, "Multi-year Projection of Run-off Conditional Tail Expectation (CTE) Reserves." Moody's Analytics, 2013.

[5] Clayton, Aubrey, Steven Morrison, Ronald Harasym, and Andrew Ng, "Proxy Methods for Run-off CTE Capital Projection: A Life Insurance Case Study." Moody's Analytics, 2016.

[6] Clayton, Aubrey, and Steven Morrison, "Fitting Proxy Functions for Conditional Tail Expectation: Comparison of Methods." Moody's Analytics (2018).

[7] Mandic, Danilo P. and Jonathon A. Chambers, *Recurrent Neural Networks for Prediction: Learning Algorithms, Architectures and Stability*. Wiley, 2001.

[8] Hochreiter, Sepp and Jürgen Schmidhuber, "Long short-term memory." *Neural Computation*. 9: 1735–1780, 1997.