Milliman Research Report

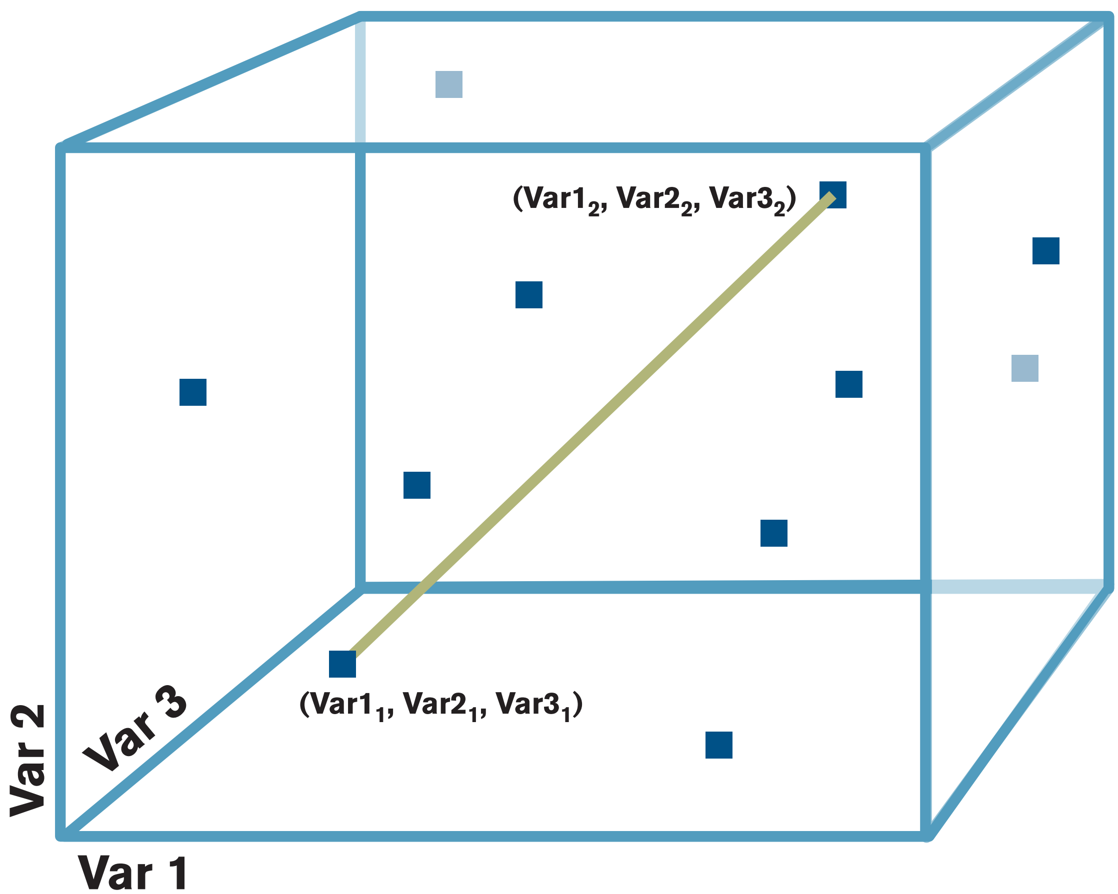
Prepared by:
Milliman, Inc.

Avi Freedman
FSA, MAAA

Craig Reynolds
FSA, MAAA

August 2008

Cluster analysis: A spatial approach to actuarial modeling



Milliman

Milliman, whose corporate offices are in Seattle, serves the full spectrum of business, financial, government, and union organizations. Founded in 1947 as Milliman & Robertson, the company has 49 offices in principal cities in the United States and worldwide. Milliman employs more than 2,000 people, including a professional staff of more than 900 qualified consultants and actuaries. The firm has consulting practices in employee benefits, healthcare, life insurance/financial services, and property and casualty insurance. For further information visit www.milliman.com.

TABLE OF CONTENTS

EXECUTIVE SUMMARY

Due to advancements in computer software and hardware, most companies can now run seriatim calculations for actuarial modeling in applications involving a single scenario forecast. In fact, companies with moderate-size in-force blocks and good hardware and software can sometimes run even multiscenario applications on a seriatim basis for a reasonable number of scenarios.

However, just as the 1990s saw actuarial models migrate from deterministic approaches to the more common use of stochastic scenarios, we are now on the verge of a significant step forward into the world of nested stochastic scenarios.

Nested stochastic applications–and also stochastic projections nested within deterministic projections–will be useful whenever an insurer wants to see a distribution of emerging earnings or surplus across multiple scenarios where the reserves or capital are set using stochastic models. However, nested stochastics result in dramatic increases in the run time of an actuarial model. It is often impractical to perform seriatim nested stochastic models for in-force blocks of any material size. Even for less computer-intensive applications, time constraints and the desire to avoid extremely large files can make the development of small models very useful.

In order to streamline and expedite this process, Milliman has developed and implemented a new type of automated model compression process that we call *cluster modeling*. Based on cluster analysis techniques frequently used in social science and other applications, cluster modeling enables users to efficiently model millions of policies into just a few thousand, or even a few hundred, model points.

In order to streamline and expedite this process, Milliman has developed and implemented a new type of automated model compression process that we call *cluster modeling*. Based on cluster analysis techniques frequently used in social science and other applications, cluster modeling enables users to efficiently model millions of policies into just a few thousand, or even a few hundred, model points. The process can accurately reproduce the results of the original seriatim model across a range of economic or experience scenarios.

Cluster modeling automatically assigns all policies from a seriatim in-force file to one of a small user-selected number of representative model points.[1] Conceptually, it computes a *distance* of each policy from every other policy, and defines the importance of each policy as the product of its size and the distance from its nearest neighbor. Cluster modeling then assigns the least important policy to its nearest neighbor and grosses up the in-force amount for that neighbor. The process continues until the resulting model is reduced to an appropriate, user-specified size.

Many companies, if they have adequate hardware, are currently able to run stochastic scenarios without these techniques. However, methods for developing highly compressed models will become more and more important in the future. As the need for stochastic analysis, including nested stochastic analysis, increases, we believe that this tool will be very useful to actuarial modelers.

1 Although we use the term “policy,” the input information may already be partially mapped into cells. Alternately, we could operate on assets rather than liabilities.

INTRODUCTION

Virtually all life insurance companies around the world employ actuarial models for use in applications such as financial forecasting, product pricing, embedded value, risk management, and valuation. Depending on the company and the application, such models may be seriatim, or reflect some degree of compression. The classic process of developing cellular models from seriatim data is a combination of science and art.

In contrast, *cluster modeling*, based on cluster analysis techniques frequently used in social science and other applications, is a new type of automated model-compression process developed and implemented by Milliman. Cluster modeling automatically assigns all policies to a small user-selected number of representative model points. Conceptually, it computes a *distance* of each policy from every other policy, and defines the importance of each policy as the product of its size and the distance from its nearest neighbor. Cluster modeling then assigns the least important policy to its nearest neighbor and grosses up the in-force amount for that neighbor. The process continues until the resulting model is an appropriate, user-defined size.

Release 6.7 of MG-ALFA® contains an implementation of this algorithm. In this report, we summarize how cluster modeling works and provide examples of how we have used it with selected client models. We believe this new technique represents an important leap forward in the process of making stochastic and nested stochastic modeling practical in a production environment.

In this report, we summarize how cluster modeling works and provide examples of how we have used it with selected client models.

BACKGROUND

In some applications, models tend to be seriatim, or nearly so:

- regulatory or tax-basis financial reporting
- C-3 Phase 2 standard scenario

In other areas, actuarial modeling techniques compress similar liability cells together using what we refer to as "classic" modeling techniques. This may be done for historical reasons or in an attempt to achieve computational efficiency. A few areas where this would be common include:

- cash-flow testing
- asset-liability modeling
- stochastic modeling
- generally accepted accounting principles (GAAP) valuation and forecasting

Such models might commonly reduce the cell count in a forecast or valuation application by one to two orders of magnitude.

Due to advancements in computer software and hardware, most companies can now run seriatim calculations for actuarial modeling in applications involving a single scenario forecast. Companies with moderate-size blocks of in-force and good hardware and software can sometimes run even multi-scenario applications on a seriatim basis for a reasonable number of scenarios.

Just as the 1990s saw actuarial models migrate from deterministic scenarios to more common usage of large numbers of stochastic scenarios, we are now on the verge of a significant step forward into the world of nested stochastic scenarios. Nested stochastic applications (and also stochastic projections nested within deterministic projections) will be useful whenever an insurer wants to see a distribution of emerging earnings or surplus across multiple scenarios where the reserves or capital are set using stochastic models.

However, just as the 1990s saw actuarial models migrate from deterministic approaches to the more common use of stochastic scenarios, we are now on the verge of a significant step forward into the world of nested stochastic scenarios.

Nested stochastic applications–and also stochastic projections nested within deterministic projections–will be useful whenever an insurer wants to see a distribution of emerging earnings or surplus across multiple scenarios where the reserves or capital are set using stochastic models, as they might be for international financial reporting standards (IFRS), principle-based approach (PBA) for reserves or capital, economic capital, VA CARVM, Statement of Position 03-1, FAS 133, fair value, dynamic hedging, or C-3 Phase 2.

Nested stochastics result in dramatic increases in the run time of an actuarial model. It is often impractical to perform seriatim nested stochastic models for blocks of in-force of any material size. As an example, consider the following calculations that might be desired to carry out a dynamic hedging analysis:

- 30-year projection with annual reporting cycles
- 1 million liability model points
- 1,000 scenarios
- dynamic hedge rebalancing at the end of each year, with each rebalance requiring path projections to maturity with the following specifications:
  - 30-year projection
  - 100 paths for the base case and each of 20 shocks (10 up and 10 down), resulting in 2,100 paths at each projection node

The total number of liability cell projections for this liability portfolio is:

$$(30 \text{ years}) * (1 \text{ million cells}) * (100 \text{ paths}) * (1 + 2 * 10 \text{ shocks}) * (1{,}000 \text{ scenarios})$$

$$= 63 \text{ trillion cell-path projections}$$

Assume for the moment that you have an extraordinarily fast system that can project 10,000 cell-paths per second. This would still add up to about 6.3 billion seconds, or around 200 years. This is clearly not practical. We are forced to consider other options.

CLASSIC APPROACHES TO MODELING

When addressing the problem above, there are really only a few drivers that can reduce the run time:

- faster hardware and software
- more machines
- fewer scenarios
- fewer paths or shocks
- less frequent rebalancing
- fewer cells

While these options can improve run time, no single one solves the entire problem. We focus here on reducing cells because that has the most potential impact. Typically, a range of options is used in actuarial modeling:

(1) seriatim
(2) modified seriatim, combining policies issued in the same month, in the same plan code, to policyholders of similar characteristics
(3) as (2), but also modeling to quinquennial or decennial issue ages, or combining cells across issue months or premium modes
(4) as (3), but pursuing other modeling techniques, such as combining risk classes or mapping minor plans into major plans

Actuaries can take the first three options and easily automate them in SAS, Access, projection software modeling tools, or other modeling applications. While these options are useful, they rarely yield compression in a ratio of much more than 10–1. Typically, actuaries will then proceed to option (4). However, that has several drawbacks. Among them:

- The actuarial modeler needs to know something about each minor plan in order to map it into a major plan.
- Rules for mapping are subjective and can be hard to automate.
- Mapping rules need to be refreshed and enhanced as new plans are created or other characteristics of the in-force block change.
- Traditional model validation techniques (which often focus on actual versus model values of opening reserves, premium, cash values, and policies in-force) do not necessarily confirm that a model will work well across multiple scenarios. Even for a single scenario, a tight opening balance-sheet validation does not guarantee good fit of income projections.
- Rules are particularly hard to derive and apply for certain product types, notably multilife policies or policies with guaranteed minimum benefit (GMB) features that may have nonhomogeneous values of in-the-moneyness or historical behavior.

AN ALTERNATIVE APPROACH

Cluster modeling offers many advantages over classic actuarial modeling techniques. For instance, it

- applies to any product type and can be extended to include assets as well as liabilities.
- achieves far better compression ratios for a given model-to-actual fit.
- is easily automated.
- can be maintain and applied in similar ways at later valuation dates.
- allows customization to place different priorities on different measures of model fit.
- applies to seriatim in-force or to modeled in-force to create an even more modeled in-force.
- allows easy adjustment to the number of model points to produce more or less model granularity, depending on the application.
- allows easy on-the-fly analysis of model fit for differing levels of model granularity, without rerunning a model.

Cluster modeling draws on the underlying theory from cluster analysis (various techniques in many fields that reduce a large set of observations to a much smaller number of clusters). From cluster analysis comes the important aspect of "distance measure"—a measure of similarity between any two observations.

Cluster modeling draws on the underlying theory from cluster analysis (various techniques in many fields that reduce a large set of observations to a much smaller number of clusters). From cluster analysis comes the important aspect of "distance measure"—a measure of similarity between any two observations.[2] One typical method of defining a distance measure is the Euclidean distance formula, which we use in cluster modeling. With this approach, the distance measure is determined by locating each observation at a point in n-dimensional space and then finding the distance as the square root of the sum of squares of the distance in each dimension.

In a nutshell, the process works as follows:

1. As the user, you define an arbitrary number of *location variables* for each policy. A location variable is a variable the value of which you would like your compressed model to be able to closely reproduce. For items that vary by scenario, you might use values from a single scenario or a handful of *calibration scenarios*, such as:

   a. reserves, cash value, account value, or premium per unit as of the projection date
   b. present value of GMB claims per unit
   c. sum of the premiums paid in the first five years of the projection per unit
   d. first-year liability cash flow per unit
   e. present value of profit (PVP) per unit

2. You define a *size* variable to represent the importance of a given policy. This ensures that large policies are not mapped away as readily as small policies, all else being equal. For example, the size variable would typically be represented by face amount for life insurance or account value for deferred annuities.

3. You divide the business into *segments*, which instructs the program not to map across segment boundaries. Segments might be plan code, issue year, GAAP era, or any other dimension of interest. Reasons for using segments include:

   - to decrease calculation time required to perform the cluster modeling, which is roughly proportional to the sum of the squares of the number of policies in each segment (A group run as one segment will take approximately ten times as long as the same business split into 10 equal-sized segments. Assuming that the segments serve to separate policies that would be unlikely to be mapped together in any case, the results would be essentially the same.)

2 In an article in North American Actuarial Journal in 2002, Yvonne Chueh gives an example in which a distance measure is used to select a sample of interest-rate scenarios. In that paper it is the distance between any two scenarios, not policies, that is being measured, and no concept of size or segments come into play. We believe that some companies may be using cluster techniques in some form, but are not aware of any published description.

- for reporting, reconciliation, or similar reasons, where you might wish to keep policies from one segment of business from being mapped into another segment

- whenever the location variables by themselves do not, in your judgment, sufficiently distinguish policies in different segments

4. You specify the number of cells that the compressed model should contain.

5. The process defines the *distance* between any two policies using an n-dimensional sum-of-squares approach, as if the n-location variables defined a location in n-dimensional space. Thus, as an example, with three location variables, *Var1*, *Var2*, and *Var3*, the distance between policy 1 and policy 2 could be measured as:

$$\sqrt{(Var1_1 - Var1_2)^2 + (Var2_1 - Var2_2)^2 + (Var3_1 - Var\ 3_2)^2}$$

   In this definition, the location variables must be appropriately scaled. You may find it convenient to "normalize" each of the location variables by dividing each one by the size-weighted standard deviation of the associated variable. Users can also introduce weights to place different priorities on matching different distance variables.

   With computer storage limits, not all distances can be kept in memory. However, after users have set the initial distance calculations, there is relatively little need for recalculation.

6. The *importance* of each policy is defined by the cluster-modeling process as the policy size times the distance from the nearest policy. Thus, a policy is unlikely to be mapped to another if it has a large size and is far away from others; however a small policy or one that is very close to another is likely to be mapped to another policy.

7. At each step, the process finds the policy with the lowest importance and maps it to its nearest neighbor (the *destination policy*), adjusting the size of the destination policy in the process. The user then repeats this step until the model has the desired number of model points.

At this point, only the user-specified target-number of clusters remains. In the last step, the process finds the most representative policy in each cluster, which is the policy in each cluster that is closest to the average location of all cells in the cluster. In general, each cell in the compressed in-force file will consist of a policy from the original in-force file, scaled up (i.e., with all variables that are logically additive grossed up by the size of the entire cell group over the size of the original policy, plus all other variables taken from the original policy).[3]

3 In certain contexts it may be preferable to add up the values from the cells in the cluster rather than scaling. This might be done, for example, to ensure that the model policy count matches the actual number.

GRAPHICAL REPRESENTATION

The following three figures help demonstrate the cluster-modeling process. In these figures, we assume just two location variables that reflect two dimensions. The scatter plot in Figure 1 represents the value of each location variable by the point placement on the two-dimensional graph. The size of each dot represents the size of the policy. In Figure 2, each policy has been assigned to a cluster. Finally, the resulting four-point model is shown in Figure 3 with the size of the four model points appropriately grossed up.

FIGURE 1

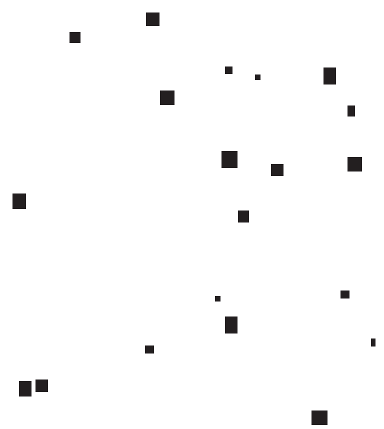


FIGURE 2

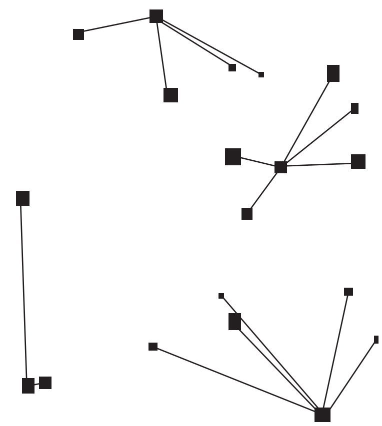
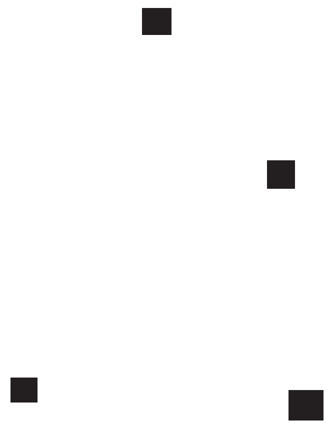


FIGURE 3



POTENTIAL APPLICATIONS

We can generate various types of models using the the cluster-modeling process described above:

- **Medium-sized models**: These essentially replace classic models, closely tracking what the result would be from running the model seriatim. In this case, we use a fairly large number of segments or cells per segment. The location variables would be those variables typically used in creating models (issue age, issue year, in-the-moneyness, etc.), or financial variables as in the discussion above, or a mixture. This application is most similar to classic modeling. However, in classic modeling, the user has to evaluate the model in light of a specific number of cells and experiments with whatever adjustments are necessary to reach that target ("This model plan is not too big; what if I widened the age bands and in-the-moneyness bands a little bit?…"). In the cluster model process, the user needs to specify the tradeoffs only once, in the distance function, and the program will do the rest of the work.

- **Small models**: These models are designed to reproduce seriatim results closely enough to estimate, say, a CTE. However, any modeling (particularly for products with significant optionality) creates some distortion; the results cannot be expected to be quite so close as would be possible with a larger model, especially with tail scenarios.

- **Very small models**: These might be used, for example, to run on a very large set of scenarios in order to select a set of sample scenarios to run with the small models. Likewise, very small models might be used to quantify the "delta" from sensitivities to key assumptions where otherwise you would usually need to use a large number of scenarios.

For models that are fairly large, many distance functions would work reasonably well. For smaller models, the selection of a distance function is not trivial, requiring some thought and tests against larger models to see how well variables of interest fit. A few iterations may be necessary before an appropriate function is found.

APPLICATION TO A TRADITIONAL LIFE/HEALTH MODEL

First, let us consider an example to illustrate how matching a few variables can have the effect of closely matching cash flows throughout a projection.

We start with a model of approximately 120,000 cells of traditional Japanese life-health business. The model includes life, accident and health, and annuity benefits on a traditional product chassis. A smaller (200-cell) model was created using a distance function that consisted of:

- initial reserve (weight 1)
- first-projection-year premiums (weight 1)
- first-projection-year claims (weight 1)
- present value (PV) of proxy profits (weight 8)
- PV of proxy profits through 10 projection years (weight 6)
- PV of proxy profits through 20 projection years (weight 6)

As a check on the modeling process, we compare the values of the location variables between the original 120,000-cell liability model and the new 200-cell cluster model.

FIGURE 4

200-CELL CLUSTER MODEL VS. 120,000-CELL CLASSIC MODEL
LIFE-HEALTH MODEL
(¥ MILLIONS)

	ORIGINAL	NEW	DIFFERENCE	RATIO
INITIAL RESERVE	372,911	371,605	(1,306)	99.65%
FY PREMIUMS	85,708	81,645	(4,063)	95.26%
FY CLAIMS	36,485	35,162	(1,322)	96.38%
PVP	154,467	154,444	(23)	99.99%
PVP-10	77,808	77,634	(174)	99.78%
PVP-20	119,924	120,001	77	100.06%

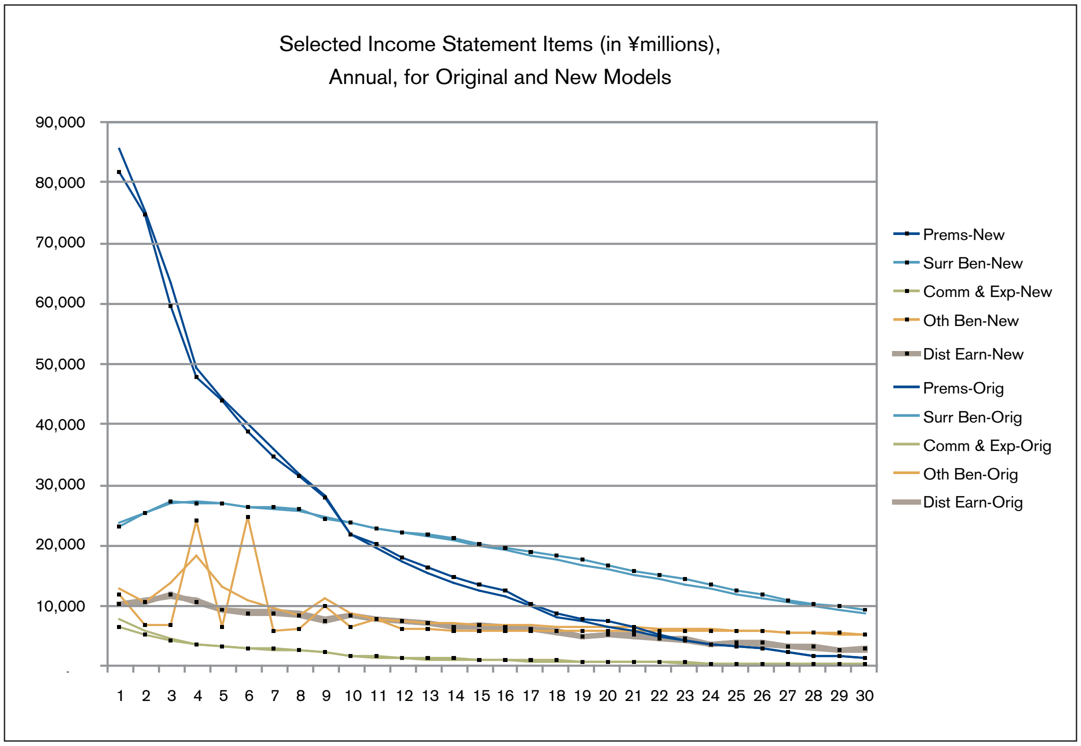
This chart confirms that we were effective in matching those variables. Note also that the variables with larger weights tended to better have fit, as expected.

It is interesting, however, to graph the resulting year-by-year cash flows, showing cluster model vs. classic model. Results for several cash-flow items and for distributable earnings can be seen in Figure 5. They show that the smaller model closely reproduces the cash flows, not only overall, but also year-by-year (with the exception of some modest timing noise on some maturities in the early years which approximately nets to zero).

In other words, even a fairly simple distance function can do a good job of reproducing the cash flows and other items. (Small items, however, may well have ratios significantly different from 100% if not specifically controlled for.) Of course, some models will be more “well-behaved” than others, and testing is necessary.

FIGURE 5

FIVE PAIRS OF VALUES ARE SHOWN; THE PAIR REPRESENTED BY THICK LINES REPRESENTS DISTRIBUTABLE EARNINGS; THE OTHERS REPRESENT (IN DESCENDING ORDER AS OF YEAR 1) PREMIUMS, SURRENDER BENEFITS, OTHER BENEFITS, AND COMMISSIONS/EXPENSES.



APPLICATION TO A TERM LIFE MODEL

As a more detailed example of this process, the following illustration shows a block of term life insurance with approximately 1.1 million policies in-force. For this block, we define location variables and weights as follows:

- initial reserve (weight 10)
- present value of proxy cash flows (PVCF) for years 1–5, 6–10, 11–15, 16–20, 21–25, 26–30, each with weight 4, and overall with weight 10 (proxy cash flows equal premiums less commissions and expenses less death benefits)
- premiums (for the six five-year groups) and death benefits (for the six five-year groups), with declining weights 1, 0.7, 0.5, 0.3, 0.2, 0.1

We used issue year to define segments for this block, because the company had periodically changed its product design and structure, maintaining the ability to analyze in-force value by issue era. Level term period was also used, primarily to save time (policies with different level-term periods would rarely be mapped across in any case).

The full seriatim model was rerun once, capturing seriatim values of each of the location variables to a file. We then executed the compression process with target-model sizes of 10,000 and 300 policies. Figure 6 compares how certain key location variables fit for each of the model sizes.

FIGURE 6

TERM MODEL FIT ANALYSIS
($ MILLIONS)

VARIABLE	SERIATIM ACTUAL	10,000-CELL MODEL	10,000 DIFF	300-CELL MODEL	300 DIFF
INITIAL RESERVE	4,097	4,089	-9	4,076	-22
CUM. PV OF CASH FLOWS	2,156	2,156	0	2,163	6
CLAIMS 1–5	2,378	2,321	-56	2,322	-56
CLAIMS 6–10	2,749	2,701	-48	2,675	-74
CLAIMS 11–15	2,455	2,417	-38	2,404	-51
CLAIMS 16–20	1,573	1,553	-21	1,538	-35
CLAIMS 21–25	997	983	-15	985	-12
CLAIMS 26–30	547	537	-10	532	-15
PREMIUM 1–5	3,255	3,187	-69	3,233	-22
PREMIUM 6–10	3,318	3,271	-47	3,269	-49
PREMIUM 11–15	3,309	3,270	-39	3,273	-36
PREMIUM 16–20	3,313	3,290	-23	3,295	-18
PREMIUM 21–25	2,706	2,689	-17	2,707	1
PREMIUM 26–30	2,047	2,033	-14	2,077	29
PVCF 1–5	481	477	-4	482	1
PVCF 6–10	190	195	4	191	0
PVCF 11–15	278	280	2	276	-2
PVCF 16–20	537	536	0	539	2
PVCF 21–25	399	398	-1	397	-2
PVCF 26–30	271	271	-1	278	7

At first glance, it appears nothing has been accomplished if the seriatim model must be run to get the data to produce a faster and smaller model. However, now that this model has been determined, it can be used to run a large number of scenarios for economic capital calculations or ERM that might be impractical with larger models. As an example of this, Figure 7 compares the present value of after-tax profits at a 5% discount rate under five different scenarios (including the base scenario for the location variables) and both levels of model granularity.

FIGURE 7

	TERM MODEL PV (AFTER-TAX PROFITS) ($ MILLIONS)				
SCENARIO	SERIATIM	10,000-CELL MODEL	10,000 DIFF	300-CELL MODEL	300 DIFF
BASE	4,309	4,304	5	4,295	14
MORTALITY*115%	3,649	3,656	-7	3,651	-3
MORTALITY*85%	4,978	4,960	18	4,945	33
LAPSE*115%	3,714	3,709	5	3,685	29
LAPSE*85%	5,251	5,246	5	5,266	-15

Clearly, the fit is very good for both models. This gives some degree of confidence that it would work well under other types of scenarios such as stochastic-mortality simulations, pandemic analysis, or a range of economic scenarios. However, as the number of model cells drops, the fit modestly worsens; the user would have to determine how much variability is acceptable in exchange for the reduced run time. Likewise, depending on the application, you might want to increase the weight on premiums and claims separately, or to include location variables reflecting the pattern of reserve runoff.

APPLICATION TO A VARIABLE ANNUITY MODEL

We also applied the cluster technique to a variable annuity (VA) model, run through 1,000 economic scenarios. Traditional mapping techniques are challenging to employ for this product line for many reasons, including the following:

- Otherwise similar policies might have decidedly different in-the-moneyness ratios.
- Policyholder investment allocations that drive future performance are infinitely granular and flexible.

Because of these and other similar factors, it is hard for classical techniques to produce a VA model that fits well with a compression ratio of more than 10– or 20–to–1. However our model of 250 cells created from an in-force block of 210,000 policies has an 840–1 ratio.

We used seven segments corresponding to the company's plan grouping. The size variable is starting account value.

A VA model is useful only if it performs well under a variety of economic scenarios. Many companies find it hard to run their full model seriatim across 1,000 scenarios or more for VA CARVM or C-3 Phase 2. However the seriatim models can be run across a few representative scenarios in order to develop appropriate location variables. In this case, we used two scenarios:

- level interest rates, all equities with a total return of 15% per year forever
- level interest rates, all equities with a total return of -5% per year forever

The user has a flexible choice of how many and which scenarios to use for calibration. We have provided these two to illustrate boundaries in policy performance. Our location variables follow, with weights as shown:

- general account value (AV) (weight 3)

- AV in the three separate account funds that, for AG34 purposes, are considered anything other than equity (i.e., bond or balanced) (The model uses 10 separate account funds. Another possibility would have been to include each fund as a location variable–weight 6.)

- PV of death benefits, surrender benefits, partial withdrawals, and annuitization benefits, in each of the two calibration scenarios (weight 1, except 0.5 for annuitization)

- PV of proxy profits in each of the two scenarios (weight 10 for +15 and 15 for -5; based on trial and error) (The latter was more difficult to match, and in a typical usage, the bad scenarios require the most concern.)

- AV at the end of 10 and 20 years, in each of the two scenarios (weight 3)

The fit of certain variables is shown in Figure 8. Note that there is some shuffling among the funds; this could be reduced by using a separate location variable for each fund, but at the likely cost of worsening the fit of the other variables.

FIGURE 8

SCEN	VARIABLE	SERIATIM	VA MODEL FIT ($ MILLIONS) 250-CELL MODEL	DIFFERENCE
	INITIAL GA AV	766	734	-33
	INITIAL SA1 AV	742	439	-303
	INITIAL SA2 AV	2,982	3,034	53
	INITIAL SA3 AV	2,165	2,537	371
	INITIAL SA4 AV	2,000	2,056	56
	INITIAL SA5 AV	1,938	1,881	-57
	INITIAL SA6 AV	2,119	2,131	12
	INITIAL SA7 AV	1,821	1,900	79
	INITIAL SA8 AV	1,532	1,304	-228
	INITIAL SA9 AV	1,240	1,192	-49
	INITIAL SA10 AV	284	354	70
+15	PV DEATH BENS	3,463	3,406	-57
+15	PV SURRENDER BENS	17,050	17,052	2
+15	PV PARTIAL WDS	5,179	5,211	32
+15	PV ANN BENS	559	462	-97
+15	PV PROFITS	1,256	1,240	-16
+15	AV 10	5,305	5,360	56
+15	AV 20	2,101	2,062	-39
-5	PV DEATH BENS	1,846	1,758	-88
-5	PV SURRENDER BENS	7,772	7,843	71
-5	PV PARTIAL WDS	3,104	3,134	29
-5	PV ANN BENS	273	202	-71
-5	PV PROFITS	-544	-540	4
-5	AV 10	1,224	1,228	4
-5	AV 20	40	38	-2

The model was tested by running both the 250-cell cluster model and the company's 9,000-cell classic model (as well as a 50-cell cluster model, not shown above) on a set of 1,000 scenarios. Results are shown in terms of ending surplus after 30 years, in millions of dollars. (Taking into account a 30-year discount factor, an ending surplus of $42M corresponds to a present value of 10 basis points of initial AV.)

Results for each of the three scenarios are graphed in Figure 9, ordered according to the surplus produced under the 9000-cell scenario. (The best 20 scenarios are eliminated from the attached table to preserve the scale of the graph). Because the results of the 9000-cell model are the basis of the ordering, the curve representing this model is smooth.

We see that even the 50-cell model generates results close to the 9,000-cell model, albeit with more volatility than the 250-cell model. Even the 250-cell model shows some noticeable deviations, presumably in scenarios where certain funds (those for which the model fit is loosest) perform unusually well or poorly. However, such deviations are smoothed when averaged over a larger number of scenarios.

FIGURE 9

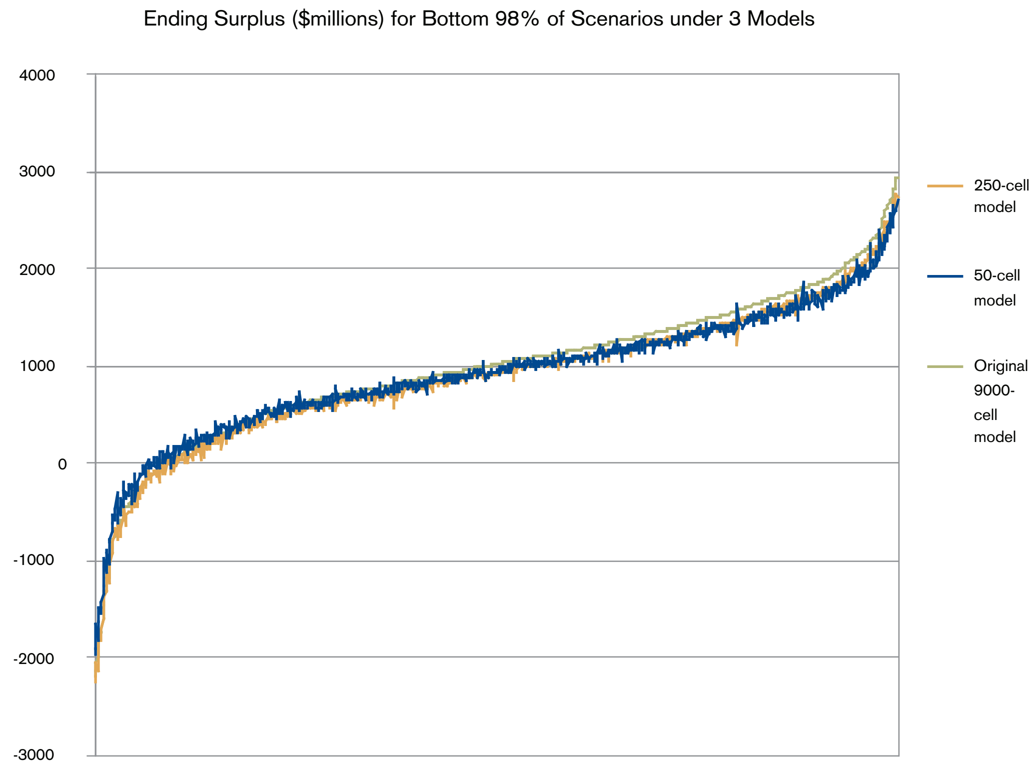


Figure 10 compares the 250- and 9,000-cell models at various CTE levels:

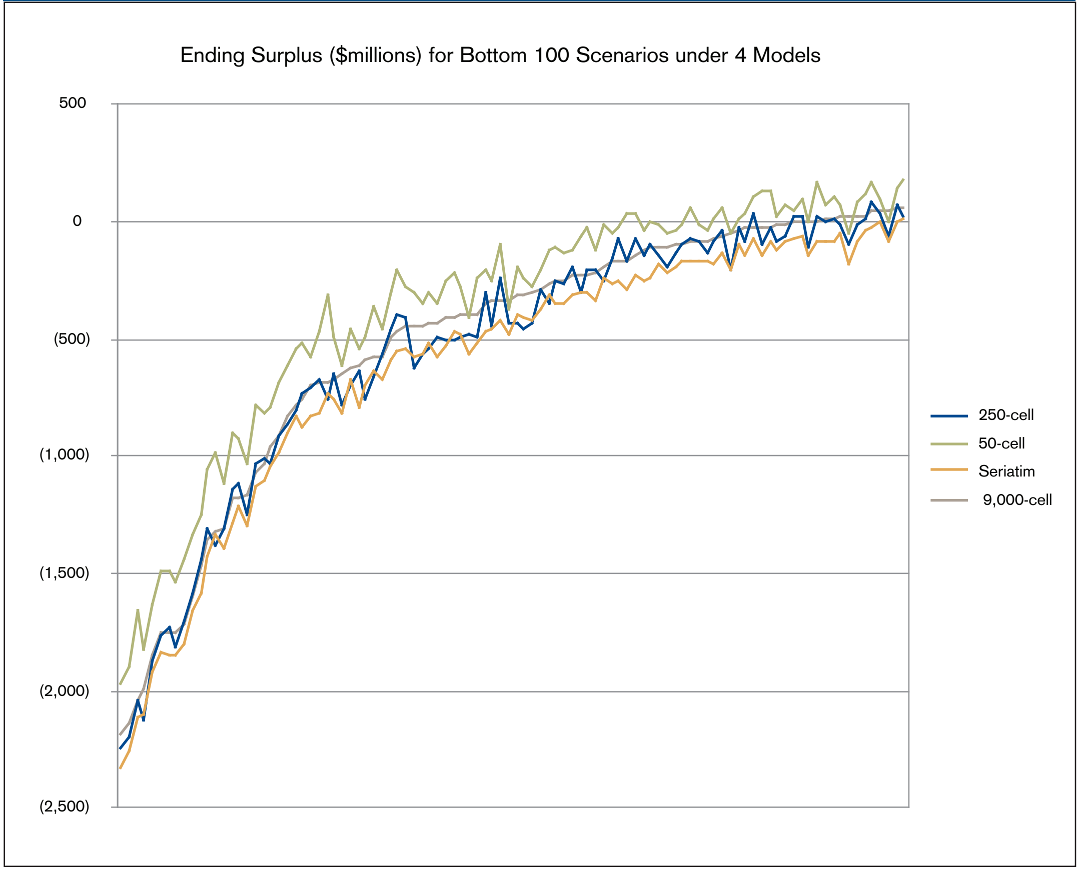
FIGURE 10

	VA MODEL CTE OF TERMINAL SURPLUS 250 CELL CLUSTER MODEL COMPARED TO 9,000 CELL CLASSIC MODEL ($ MILLIONS)		
CTE LEVEL	**250 CELLS**	**9,000 CELLS**	**DIFFERENCE**
0%	**$965**	**$1,058**	**$93**
50%	**359**	**427**	**67**
65%	**159**	**218**	**59**
75%	**-28**	**21**	**49**
90%	**-548**	**-515**	**32**
95%	**-981**	**-939**	**42**
99%	**-1,909**	**-1,879**	**30**

In this case, as can be seen from either Figure 9 or Figure 10, the 250-cell cluster model gives systematically lower results than the 9,000-cell model—$93M on average (recall that $42M corresponds to 10 basis points). Most of this error, however, is in the 9,000-cell classic model. Testing with the original seriatim model reveals that the 9,000-cell model is approximately $70M higher on average than the seriatim model.

In many settings, attention is likely to be focused on the tail. Figure 11 compares results for the four models (50 cells, 250 cells, 9,000 cells, and seriatim) on the 100 worst-case scenarios (as measured by the 9,000-cell model). While the 50-cell model apparently diverges too much from the seriatim to be useful for tail analysis, the 250-cell model is actually an improvement over the 9,000-cell classic model.

FIGURE 11



VALIDATING A MODEL

Any model needs to be validated in some way. Models derived using classic techniques are frequently validated by using a static validation (to show that balance-sheet items are similar to the actual in-force), and a dynamic validation (to show that income-statement items are reasonably similar to actual experience). We believe that classic modeling techniques are often applied in a fairly ad hoc manner, with relatively little testing of fit in different scenarios (i.e., results for different items obtained by using the model, as compared to results obtained by using a seriatim calculation). This is because the relatively large model sizes give assurance that the model will not diverge significantly.

With models of much smaller sizes more testing may be appropriate. While the nature of such testing will depend on the application, we can offer some general guidelines. We assume that a seriatim model is available for testing, albeit impractical to run against a full set of scenarios.

On a static basis, start by reviewing the fit of the location variables. Such a review gives an idea of the trade-off between matching different variables; it may lead to adjusting the weights applied to the location variables or to changing the number of cells used in the compressed model. Users may also include results of other variables in the static validation, such as a proxy PVP for other scenarios (not necessarily used in calibration). These variables can provide additional confidence that the model will be accurate when applied to a wide range of scenarios.

Once the user is comfortable with the fit indicated on a static basis, the small model can be run, to compare results against those from the original model.

If the user creates a new model as a general-purpose replacement for the original model, he or she should compare all components of the income statement and balance sheet for all years. To ensure a good match, users should confirm whether the proxy for PVP is inadequate and needs to be refined, or if more attention needs to be paid to the pattern of reserve runoff, rather than just the initial value.

To confirm validity for a model intended to be run in a large number of scenarios, run the seriatim model against a small number of scenarios. (You can select these based on the results of the large model–for example, depending on the application, a stratified sample of tail scenarios, rather than random scenarios from throughout the distribution.)

How much testing and validation needs to be done will depend on the particular application and on what provides adequate assurance to auditors, regulators, or others who need to be convinced of the accuracy of the model. Our expectation, however, is that multiscenario testing will be performed the first time the model is run and occasionally thereafter, but not each month or quarter. Static testing is likely to be performed each time a new model is created.

NEXT STEPS

We believe this approach has great potential. Experience will provide more guidance in the determination of location variables and associated weights and how many calibration scenarios are necessary. Undoubtedly the "right" choice will depend on the particular in-force block and the application. This technique offers considerable advantages over classic actuarial modeling techniques.

This approach is applicable well beyond working with liabilities. The MG-ALFA interface allows this same technique to be used for asset modeling, with appropriate choices of location variables and scenarios. A similar interface to scenario compression, using the same techniques, is also under development.

Many companies, if they have adequate hardware, are currently able to run stochastic scenarios without these techniques. However, methods for developing highly compressed models will become more and more important in the future. As the need for stochastic analysis, including nested stochastic analysis, increases, we believe that this tool will be very useful to actuarial modelers.

Experience will provide more guidance in the determination of location variables and associated weights and how many calibration scenarios are necessary. Undoubtedly the "right" choice will depend on the particular in-force block and the application. However, this technique offers considerable advantages over classic actuarial modeling techniques.

Milliman

1301 Fifth Avenue, Suite 3800
Seattle, WA 98101
(206) 624-7940

www.milliman.com