Final Report: ACT496

Variable Annuity: A surrogate modelling approach

Raj Patel

Supervised by Professor Sheldon Lin

Department of Statistics and Actuarial Science, University of Toronto

Contents

Annuity and its Components:

Annuities are financial instruments that can guarantee lifetime stream of income during retirement. Two primary reasons why an individual buys annuities are: (1) accumulate money for retirement through tax-deferred savings (2) guaranteed monthly income lasting as long as the individual lives. However, besides these, unlike other investments, annuities also provide a variety of benefit alternatives such as protect against untimely death, provide principal guarantees, assure a specified amount of income when the contract is annuitized and/or guarantee withdrawals for life.

In general, most annuity contracts are issued by life insurance companies. In the contract, contract owner, annuitant and beneficiary are identified.

Contract Owner:

In the contract, the owner of the annuity who pays the premium is the contract owner. Generally, the owner is an individual or a couple. However, a trust or a partnership can be an owner as well (special tax rules apply for these cases).

Annuitant:

In the contract, the annuitant is the person upon whose life the annuity payments are based. Generally, the contract owner is the annuitant. So, the payments continue as long as the owner is alive. In some cases, two people, such as owner and the spouse, are designated as joint annuitants. As a result, the income continues as long as either one survives. Such annuities are called joint or survival annuities

Beneficiary:

Beneficiary is the person who receives any payments that may be due upon the death of the owner or annuitant.

Now that we know the basics of annuities, we will take a look at Variable Annuities.

Variable Annuity:

This is purchased either with a single premium or with periodic payments to help save for retirement. These premiums, based on the choice of contract owners from a wide range of investment options, are then invested. These are invested in a single underlying mutual fund or, in some cases, in a "fund of funds," which is a mutual fund that invests in several other mutual funds or in exchange-traded funds (ETFs). All of these are termed as VA subaccounts.

In this type of annuity, the contract owner determines the point at which accumulated principal and earnings are converted into a stream of income. However, the contract value or the income payments vary based on the investment performance of underlying subaccounts.

Variable Annuity has two phases: (1) Accumulation phase or a savings phase (2) Payout or retirement income phase. As with mutual funds, the investment return of variable annuities fluctuates. During the accumulation phase, the contract value varies based on the performance of the underlying subaccounts chosen. During the payout phase of a deferred variable annuity (and throughout the entire life of an immediate variable annuity), the dollar amount of the annuity payments may fluctuate, again based on how the portfolio performs.

Fees and Expenses

A variable annuity involves direct expenses in the form of insurance charges and indirect expenses in the form of management and other fees and expenses associated with the underlying mutual funds in which the variable annuity subaccounts invest.

The fees and charges commonly associated with variable annuities include mortality and expense risk charges (M&E fees), administrative charges, and distribution charges. In most contracts, the M&E fee pays for three important insurance guarantees:

- The ability to choose a payout option that provides an income that cannot be outlived at rates set forth in the contract at the time of purchase
- A death benefit to protect beneficiaries
- The promise that the annual insurance charges will not increase.

GMDB

Variable annuity contracts have traditionally offered a guaranteed minimum death benefit (GMDB) during the accumulation period that is generally equal to the greater of (a) the contract value at death or (b) premium payments minus any prior withdrawals. The GMDB gives contract owners the confidence to invest in the stock market, important in keeping up with market inflation, as well as the security to know their families will be protected against financial loss in the event of an untimely death.

Some life insurance companies offer death benefits that step up or increase based on pre-determined criteria. Called contract anniversary value or ratchet, these enhanced GMDBs are equal to the greater of (a) the contract value at death, (b) premium payments minus prior withdrawals, or (c) the contract value on a specified prior date. The specified date could be a prior contract anniversary date, such as the date at the end of every seven-year period, every anniversary date, or even more often. A ratchet GMDB locks in the contract's gains on each of the dates specified.

Some insurers offer a rising floor GMDB that is equal to the greater of (a) the contract value at death or (b) premium payments minus prior withdrawals, increased annually at a specified rate of interest. In some cases, a ratchet and a rising floor may be available within the same contract. Some contracts offer a choice of a ratchet or a rising floor.

GMLB

Guaranteed Minimum Income Benefit

A guaranteed minimum income benefit (GMIB) rider is designed to provide the investor with a base amount of lifetime income when they retire regardless of how the investments have performed. It guarantees that if the owner decides to annuitize the contract, payments are based on the amount invested, credited with an interest rate typically 4-5%. An investor must annuitize to receive this benefit and there is typically a seven-ten year holding period before it can be exercised.

Guaranteed Minimum Accumulation Benefit

A guaranteed minimum accumulation benefit (GMAB) rider guarantees that an owner's contract value will be at least equal to a certain minimum percentage (usually 100%) of the amount invested after a specified number of years (typically 7-10 years), regardless of actual investment performance. However, considering the financial risk associated with this, many GMABs require some form of asset allocation.

Guaranteed Minimum Withdrawal Benefit

A guaranteed minimum withdrawal benefit (GMWB) rider guarantees that a certain percentage (usually 5-7%) of the amount invested can be withdrawn annually until the entire amount is completely recovered, regardless of market performance.

Annuity Sales in US (in $ Billions)

	Variable	Fixed	Total
2015	133.0	102.7	235.7
2016	104.7	117.4	222.1
2017	98.2	105.3	203.5
2018	100.2	133.6	233.8
2019	101.9	139.8	241.7

Within Variable Annuity(VA) in US, Jackson has been the leader in the VA market for seven straight years. The top three VA sellers were Jackson, Equitable Financial and TIAA. Together, they represented 36% of the total VA market in 2019.

While within Fixed Annuity ib US, sales reached $139.8 billion in 2019, up 5% from the prior year. AIG led the sales for second consecutive year, selling Fixed Annuities worth $13.2 billion. While the other two in top three were New York Life and Allianz Life of North America. Together, they represented 23% of the U.S. fixed annuity market.

Liability using Simulation

Now that we saw how to calculate the fair annual fee for a simplified case, we will try to understand how to calculate the liability of a VA (without fees) when the underlying asset follows Geometric Brownian Motion (GBM).

In the previous section, we saw how to model GMAB with a 'Return of Premium' guarantee and observed that the liability of GMAB in that case was simply a put option. In this section, we will focus on the ideation behind pricing GMWB, GMAB and GMDB individually. Following that, we will see how to price a policy comprising of all those riders using Efficient Nested Simulation.

Liability of a Plain Vanilla GMWB without survival probabilities

Under this approach, initially, for simplicity, we assume that the policyholder makes equal withdrawals wh at the end of each time step until T years and hence there are $N = \frac{T}{h}$ guaranteed benefit payments of wh (where h is the size of each time step).

Furthermore, we will be using superscripts $-$ and $+$ to indicate the values of a quantity prior to and after the benefit payments are made at that time. So, the account value at time t before the withdrawal will be A_t^- while following the withdrawal, it will be A_t^+. For the first instance when $A_t^- \leq wh$, $A_t^+ = 0$. For $t^* > t$, $A_{t^*}^-$ and $A_{t^*}^+$ will be set to 0.

In this scenario, there will be no liability payments for the insurance company as long as $A_t^- \geq wh$ as there is enough in the account to pay the withdrawal payments. Liability starts once $A_t^- < wh$. This is when the withdrawal payment will be made out of the insurance company's pocket.

So, using the above approach, at time t, the liability will be:

$$L_t = \max(wh - A_t^-, 0)$$

Once we have this for each time step, we can take the present value of the liability payments at each timestep and that gives us the PV(liability) for the policy.

Example : Here, we show pricing of a liability of a GMWB with following parameters for the underlying asset (which we assume follows a Geometric Brownian Motion(GBM))

Parameter	Input
S_0 (Initial Asset Value)	\$100
μ	0.1
σ	0.18
T	10 (years)
δT	0.004 (1 day, considering 250 business days in a year)
r	0.04
w (withdrawal rate)	0.10
h (time step)	1
Simulations	1000

Based on the above inputs, we get the following result for 1000 simulations for pricing a GMWB liability:

Present value of this total liability for these 1000 scenarios: \$617.08

Therefore, expected present value of liability for a single policy is $\boxed{\$0.61708}$

Liability of a GMAB and GMDB

Here, we will show how to price a GMAB that guarantees a return of at least the original invested premium compounded at some annual growth rate g. So, the payout to the policyholder is: $max(e^{g\tau}AV_0, AV_\tau)$.

In this case, τ is the maturity for the policy, $g \geq 0$ is the guaranteed instantaneous growth rate, AV_0 is the initial account value while AV_τ is the account value at the time of the maturity of the policy. To simplify, we consider that there are no fees.

To make this a GMDB, we can use τ as a random variable. In that case, unlike standard financial options, GMDBs will have stochastic maturity with τ being the time of involuntary death. In the literature, these are called Titanic options. However, for simplicity, we will look into the deterministic case of GMAB where $\tau = T$ for the policyholder.

As $\tau = T$, the payoff for the policyholder is:

$$max(e^{gT}AV_0, AV_T) = \boxed{(e^{gT}AV_0 - AV_T)^+ + AV_T}$$

,where AV_t denotes the policyholder's account value at time t.

From the above result, we see that $(e^{gT}AV_0 - AV_T)^+$ is a weighted European Put Option with strike price $K = e^{gT}AV_0$ and the underlying asset AV_T.

Example : Here, we show pricing of a liability of a GMAB with following parameters for the underlying asset (which we assume follows a Geometric Brownian Motion(GBM))

Parameter	Input
S_0 (Initial Asset Value)	\$100
μ	0.1
σ	0.18
T	10 (years)
δT	0.004 (1 day, considering 250 business days in a year)
r	0.04
Simulations	1000

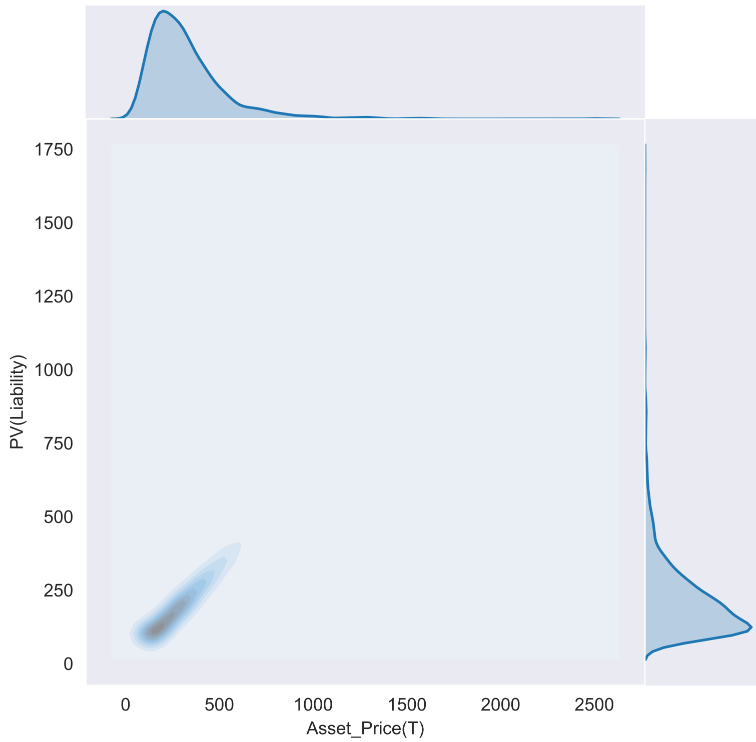


Figure 1: Kernel Density Estimation for GMAB

Based on the above inputs, we get the following result for 1000 simulations for pricing a GMAB liability:

Total liability at the end of 10 years for all 1000 scenarios: \$333341.67

Present value of this total liability for these 1000 scenarios: \$223445.60

Therefore, expected present value of liability for a single policy is $\boxed{\$223.44}$

Here, as mentioned earlier, if the time of death τ is a random variable, then we can treat this as a GMDB where exercise will be triggered by involuntary death at τ.

Now that we know how to find present value of the liability for GMAB, GMWB and GMDB, we will move to efficient simulation techniques where we will see how to price a policy with multiple riders.

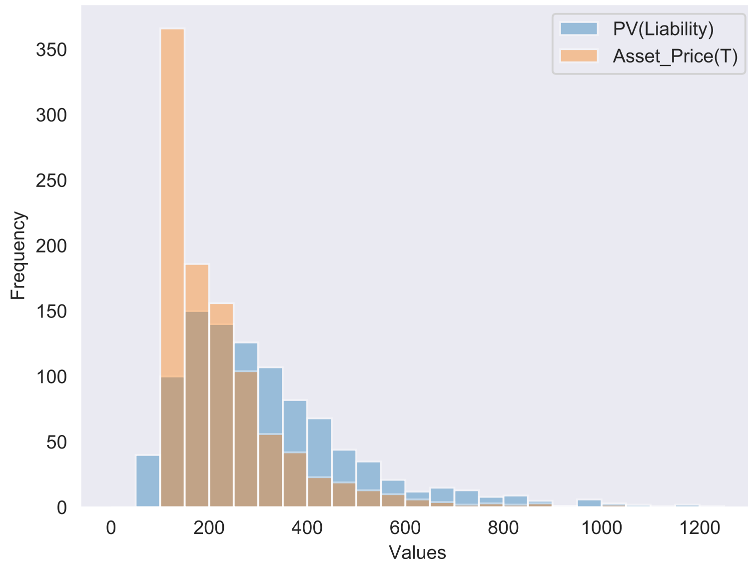


Figure 2: Histogram based on 1000 simulations for GMAB

Nested Simulation for Valuation

Although we observed the simple case of finding the present value of liabilities for different riders, in more realistic scenarios, closed form expressions are not available for calculating the VA liabilities. To overcome that, companies use the nested-simulation algorithm.

The nested simulation is a two-step simulation algorithm comprising of an outer loop and an inner loop. Without loss of generality, we consider time t=0 as the valuation time and time t=1 as the future time. Now, as part of the first step of this algorithm, each policy's account value is projected from time t=0 to t=1 using many outer-loops (where each outer-loop represents a real world scenario). In the second step, a large number of inner loops are simulated at each outer-loop to calculate the VA liability. This is done by averaging the present value of all the insurer's cash flows occuring as part of the inner-loop simulation. The sum of PV of these cash flows/liabilities for different outer-loop simulations for these policies give the distribution of total liability at time t=1 and this distribution is defined as predictive total liability distribution.

Due to demographic variation and different designs, a portfolio contains highly non-homogeneous contracts. As a result, running the nested simulation for a large VA portfolio can be time consuming. For a portfolio with 200,000 policies with 1000 outer-loops and 5000 inner-loops in each outer-loop, if we assume that the computer can process 2,000,000 simulations per second, then the algorithm will run for $\frac{200{,}000*1000*5000}{2{,}000{,}000}$ seconds which is almost 6 days. As a result, this is just not feasible from a risk management perspective.

Recent Developments and thier limitations

Considering the need for better methods for liability valuation, quite a lot of research has been done in the domain. Among that, most of the work relies on Least Square Monte Carlo (LSMC) method. The idea under this method is to approximate the VA liability by regressing the guaranteed payoffs on a set of polynomials. The procedure starts at the termination date and propogates backward through time.

However, there are couple of issues associated with this method. First is the unstable performance across different circumstances due to the accumulation of the estimation error through the backward procedure. While another issue is the determination of the order of polynomial basis function. When using this method on VA portfolios, these issues can even get magnified as VA portfolios contain a large number of non-homogeneous contracts.

However, research of efficient nested-simulation for large VA portfolios has been quite recent. Gan (2013), Gan and Lin (2015), Hejazi and Jackson (2016) and Gan and Valdez (2018) have all proposed unique ideations on efficient nested-simulation for VA portfolions. However, a common issue in these research has been the theoretical justification on the estimator of the quantity of interest. All of these primarily focused on compressing the number of policies. So, for the case where the underlying investment by the policyholders involves multiple assets, these methods will be inefficient as there will be large number of inner and outer loop projections.

Surrogate Modelling Approach by Lin and Yang:

To address all the limitations in the work done till date, Lin and Yang (2020) proposed a method in which both, the number of inner and outer loop projections and the number of policies are reduced. Under this, they first propose to use a model assisted population sampling framework to reduce the number of policies. Following that, they propose to use spline regression with scenario clustering to reduce the number of inner and outer loop projections. Since they incorporate different statistical models in this algorithm, they call this a surrogate model assisted nested simulation algorithm. The main focus of this report will be on the method they propose using spline regression with scenario clustering to reduce the number of inner and outer loop projections.

Moving beyond Linearity while modelling

Before going into the Surrogate Modelling approach, we need to understand what spline regression is. Within regression, once we relax the linearity assumption, there are multiple tools we can use. The first basic extension is Polynomial Regression.

Polynomial Regression is an extension of Linear Regression where alongside the linear predictors, we also use powers of predictors to model the response variable. So, instead of the standard model where y_i is the response, x_i is the predictor, β_i is the coefficient of the predictor and ϵ_i is the error:

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i$$

we have

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \cdots + \beta_m x_i^m + \epsilon_i$$

When the degree m is large, we can fit a highly non-linear function using Polynomial Regression.

However, Polynomial Regression imposes a global structure on the entire dataset. To overcome that, we can use Step Functions. Step Functions dissect the range of variable into K distinct regions. Once it is divided into K regions, it fits a different constant on each region. However, although the Step Functions help us overcome the global structural issue, they can only fit constants on different regions and hence can't help us capture attributes like trend in the data.

To overcome both the issues, Regression Spline is very useful and is in fact an extension of the above mentioned tools. Regression Spline divides the range of predictor into K distinct regions. Within each region, it fits a polynomial function. While fitting, these polynomials are constrained so that they join smoothly at the boundaries. As a result, in Regression Spline, the more the regions we separate the variable into, the more flexible the fit is.

A slightly different approach is to use Smoothing Spline. In Smoothing Spline, to fit the parameter $\beta's$ for our function, we use penalized least squares. Let's say we are trying to find g(x) that fits the observed data well. In that case, we would want to find function g that minimizes:

$$\sum_{i=1}^{n} (y_i - g(x_i))^2 + \lambda \int g''(t)^2 dt$$

where λ is a nonnegative tuning parameter selected using Cross Validation. The function g that minimizes the above equation is called the smoothing spline.

Spline Regression Approach

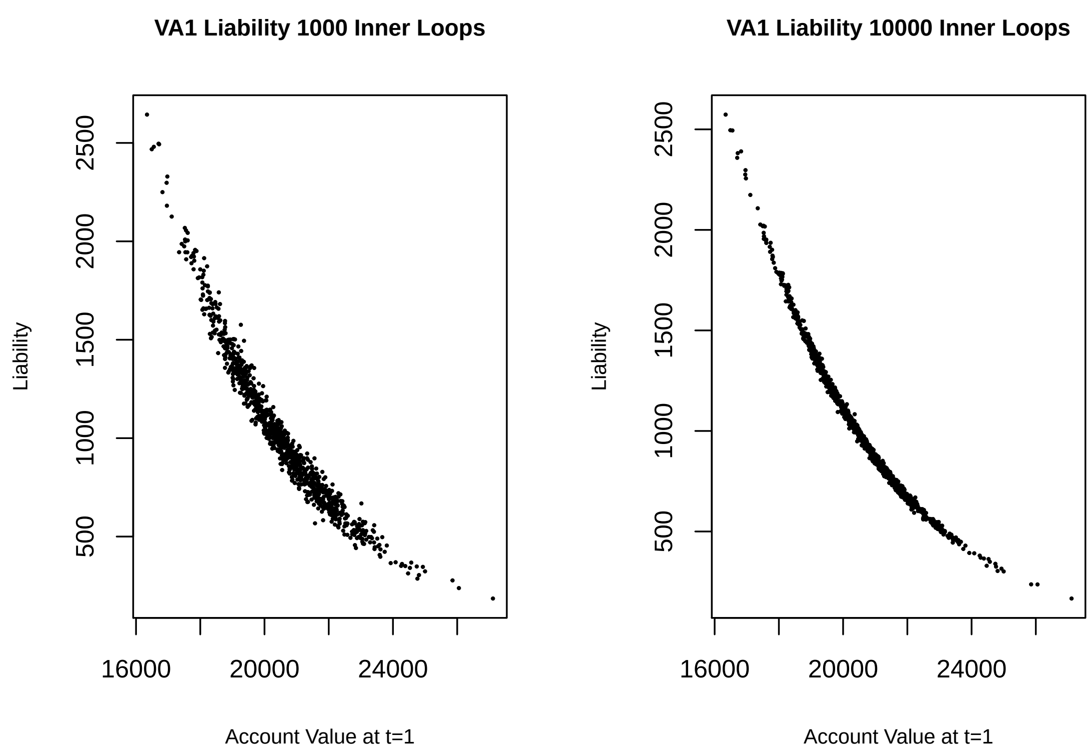
Under this approach, firstly, a set of representative outer loops is selected and then VA liabilities are calculated at those outer loops through multiple inner loops. This is done using scenario clustering. Once these are calculated, these VA liabilities are used to calculate true VA liabilities for all the outer loops. This is done by fitting a function on account value and different outer loops for a policy. This fitting can be done using Regression Splines. Previously, there have been attempts to perform similar operation using Polynomial Regression. However, in case of Polynomial regression, as we observed that the predictors impose a global structure on the response variable, it is not flexible enough to capture the dependency over the entire range. As a result, to avoid imposing such a global structure, Regression Spline is suggested by Lin and Yang in their paper.

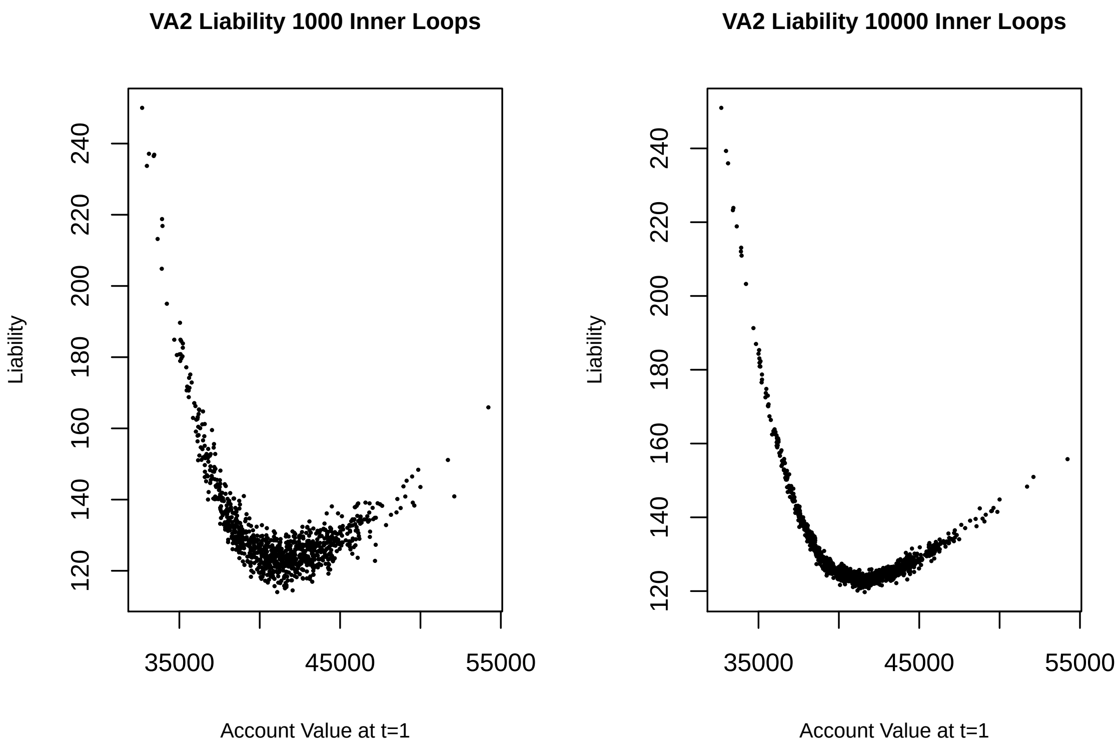
To illustrate the reduction in inner and outer-loop projection, they use the following 2 generic but common VA policies:

(1) VA1: Male, age 45, 20 years of maturity, GMDB + GMAB rider where the guaranteed death benefit base rolls up at 3% per year and the accumulation benefit base rolls up at 1% per year.

(2) VA2: Female, age 65, 15 years of maturity, GMDB + GMWB rider where guaranteed death benefit base is of ratcheting type and annual withdrawal rate equals 1/15.

Following that, two sets of nested simulation are ran for both the policies based on the regime switching lognormal model proposed by Hardy in 2001 using different sets of parameters to calculate the predictive distributions of their VA liabilities at t=1. Based on the simulation, the following observations were made: (1) the larger the number of inner-loops, the smoother the scatterplot. This is because running more loops reduces the estimation error. (2) the dependency between the predicted account value and the predicted liability is not linear and the pattern varies across policies.





To overcome this tradeoff between run time and accuracy, spline regression is used. In the context of VA, regression spline is fitted using account values and simulated liabilities. Practically, the liabilities do not need to be very accurate in order to fit the spline model and hence we can use less number of inner loops thereby enabling the simulation algorithm to run faster.

Training set selection for spline model

Now that we have seen the main idea, we will start investigating this method more comprehensively. Initially, the approach starts with an aim to reduce the number of outer loops. For a nested simulation algorithm with M outer-loops, each policy will have M predicted account value obtained through each outer-loop. At each account value, predicted liability is computed using multiple inner-loops. Now, by using spline regression, the running time may be reduced (because of smaller number of inner-loops). However, it may still be time consuming to simulate inner-loops for all outer-loops (this still is a case for a single policy which will be augmenting soon).

To overcome this, Lin and Yang propose an approach in their paper in which predicted liabilities are simulated only at some selected account values. Using those and their liabilities, a spline model is fit to estimate the liabilities at all outer-loops. Ideally this model should be close to a model fitted using the entire set of observations so that the estimated liabilities have higher accuracy.

To select the predicted account values, they use a clustering based method. In our context, let $x = (x_1, \ldots, x_M)$ be the predictive account values coming from a univariate distribution X. Let $Y = (Y_1, \ldots, Y_M)$ be the corresponding responses which in our case would be predictive liabilities of a policy at different outer-loops. Then relation between x and Y is modelled by:

$$Y_i = f(x_i) + \epsilon_i$$

where i=1,...,M. Then, to reduce running time, we aim to select a subset of predictive account values $x^* = (x_1^*, \ldots, x_m^*)$ where $m << M$. This helps in fitting spline model to a smaller amount of data.

Assuming that the model we fit is $\hat{f}$, we want to minimize:

$$\sum_{i=1}^{M} \mathbb{E}((\hat{f}(x_i) - f(x_i))^2)$$

For setting the selection criteria, an upper bound is provided in their paper which depends on the selected predictors x^*. Based on the theorem shown in the paper, let $\{C_1, \ldots, C_m\}$ be a partition of x such that $x_j^* \in C_j$ for j = 1,...,m and each C_j contains M_j elements, $\sum_{j=1}^{m} M_j = M$. When $p \geq 3$, the following holds (p+1 is the order of spline):

$$\sum_{i=1}^{M} \mathbb{E}((\hat{f}(x_i) - f(x_i))^2) \leq 3(M \max_j MSE(\hat{f}(x_i)) + \max_j MSE(\hat{f}'(x_j)) \sum_{j=1}^{m} \sum_{i=1}^{M_j} (x_i - x_j^*)^2 + \sum_{j=1}^{m} \sum_{i=1}^{M_j} O((x_i - x_j^*)^4)))$$

Now, as the partition becomes finer, the third term on right side (fourth order term) converges to 0 faster than the second term. Hence, we will ignore that and propose a method for selecting x^* that controls the first and second term. Furthermore, it is shown in the paper that asymptotically, the MSE of the derivatives of the penalized least square estimator does not depend on the predictor. As a result, the upper bound can always be written in a way that x^* only appears in the $\sum_{j=1}^{m} \sum_{i=1}^{M_j} (x_i - x_j^*)^2$.

Based on this result, Lin and Yang present an approach to select x^* such that $\sum_{j=1}^{m} \sum_{i=1}^{M_j} (x_i - x_j^*)^2$ is minimized. Although this method is not optimal in minimizing the objective function, it is very simple to implement in different situations. To implement this, find a vector which $\mathbf{x} = (x_1, \ldots, x_m)$ that minimizes $\sum_{j=1}^{m} \sum_{i=1}^{M_j} (x_i - x_j^*)^2$ using k-means clustering algorithm. This algorithm finds a partition whose objective is minimizing the within-cluster sum of squares.

In k-means clustering, we initially decide the number of clusters we want to identify in the data. Once we decide to identify k clusters, we start with k randomly selected data points. (we call this the initial k clusters). We then measure the distance from each point to all these k clusters and assign each point to the nearest cluster. Following that, we calculate the mean of each cluster. Once this iteration is done, we then repeat the same steps using the mean values as the new initial clusters. Finally, once the clusters don't change over iterations, we stop.

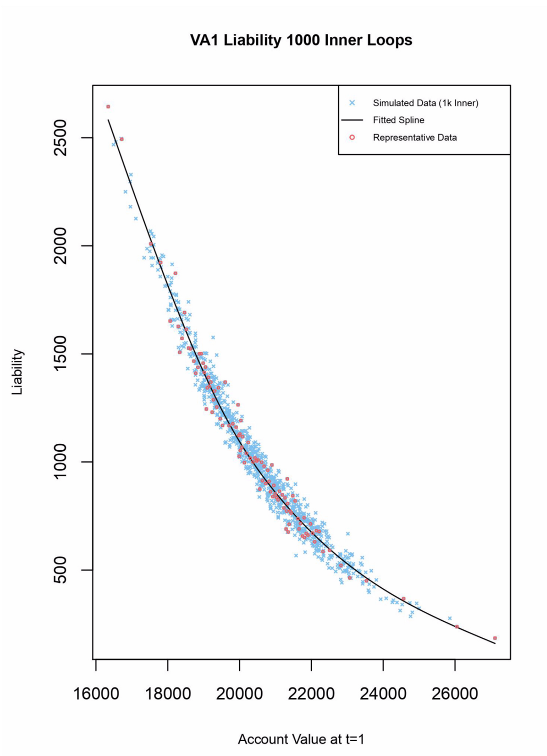
We do these numerous times as k-means clustering initializes randomly. Eventually, we select the run with least total variance.

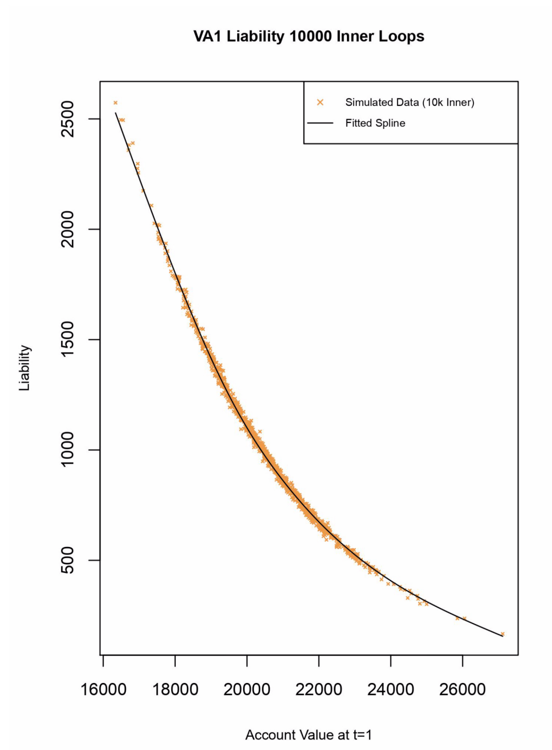
Selection of Representative Outer Loops

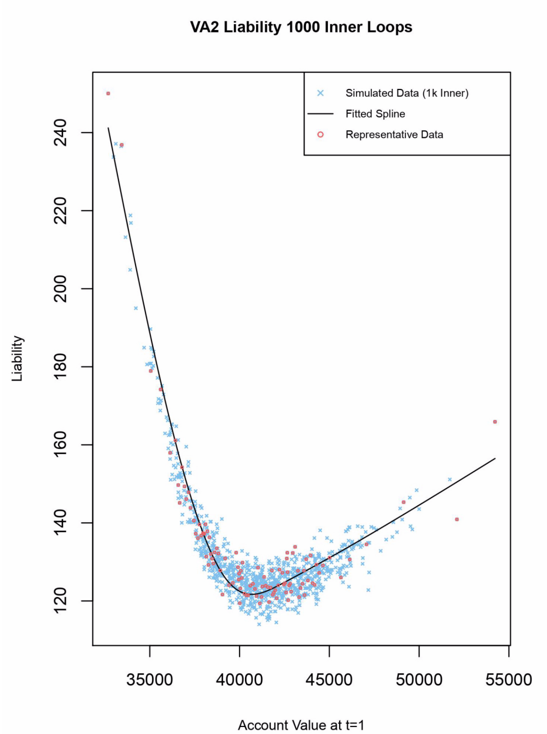
Practically, running the k-means algorithm with the predicted account values for different policies can result in different sets of training data. This can be due to couple reasons: (1) k-means clustering algorithm may start at randomly initialized cluster centres and stop at local optimum hence giving varying results. (2) clustering depends on policy's predictive account values which are different among different policies. This is inefficient as nested simulation algorithm runs for different set of algorithms for different policies.

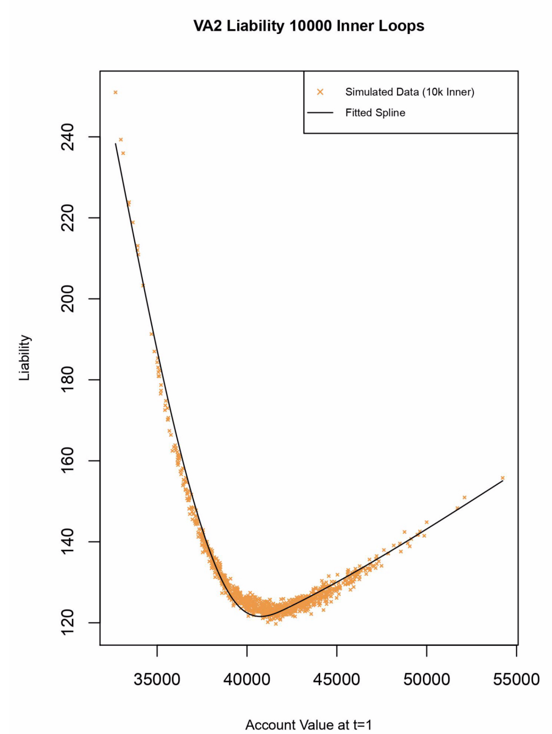
To overcome that limitation, the paper selects the training set based on the underlying asset's return from different outer loops which are referred to as representative outer-loops. To better control the variability at the tails, the max and min returns are also included. By doing this, one gets the same set of training data for all the policies in the portfolio.

Here, using 1000 inner-loop simulations and 100 outer-loops out of 1000, spline curve is fitted using 10 B-spline basis functions with equidistant knots. To compare that, the predictive VA liabilities obtained from the 100/10,000 simulation algorithm are used. Overall, the proposed method provides good approximations at all predicted account values including the two extreme regions. The results we get on the two generic policies mentioned above are as follows:


VA1 Liability 1000 Inner Loops
Simulated Data (1k Inner)
Fitted Spline
Representative Data
Liability
500
1000
1500
2000
2500
16000
18000
20000
22000
24000
26000
Account Value at t=1


VA1 Liability 10000 Inner Loops
Simulated Data (10k Inner)
Fitted Spline
Liability
500
1000
1500
2000
2500
16000
18000
20000
22000
24000
26000
Account Value at t=1

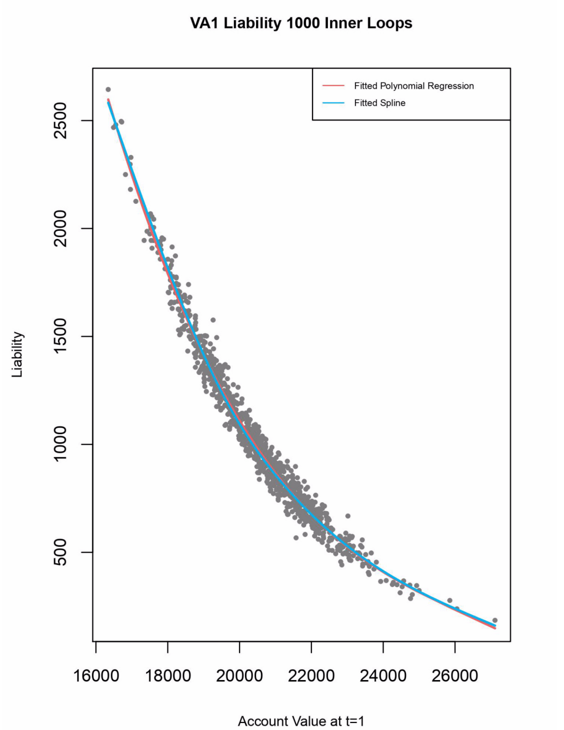

VA2 Liability 1000 Inner Loops
Simulated Data (1k Inner)
Fitted Spline
Representative Data
Liability
120
140
160
180
200
220
240
35000
40000
45000
50000
55000
Account Value at t=1

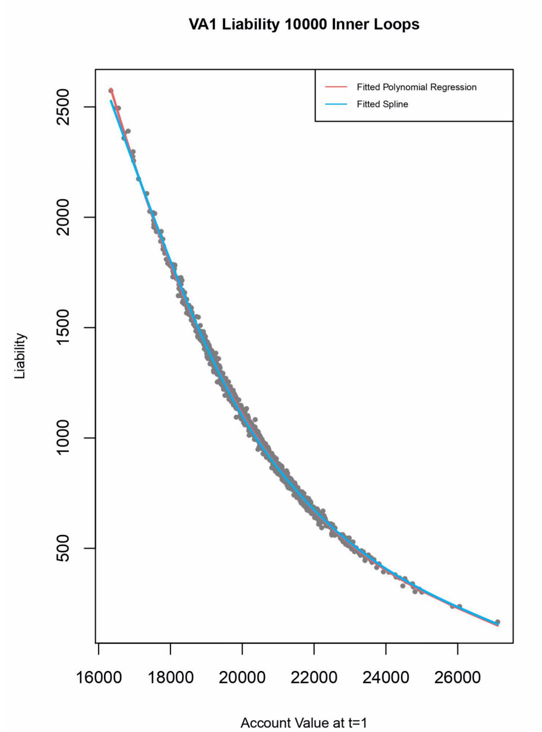

VA2 Liability 10000 Inner Loops
Simulated Data (10k Inner)
Fitted Spline
Liability
120
140
160
180
200
220
240
35000
40000
45000
50000
55000
Account Value at t=1

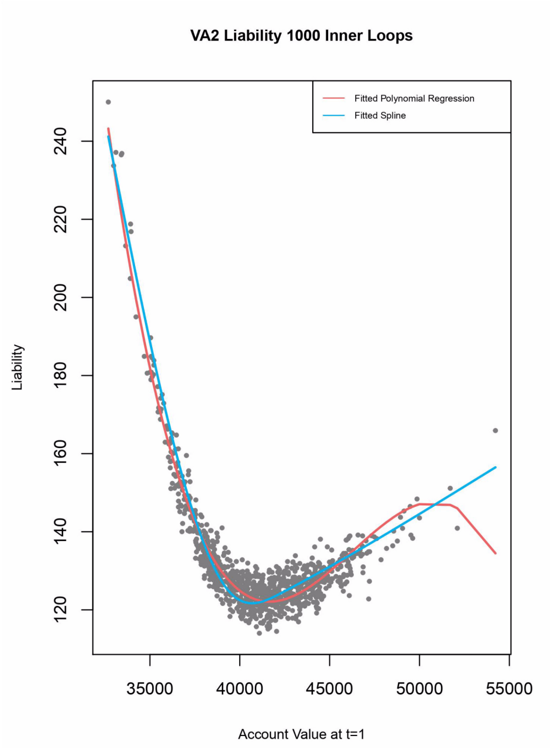
Comparing the spline approach and the polynomial regression approach

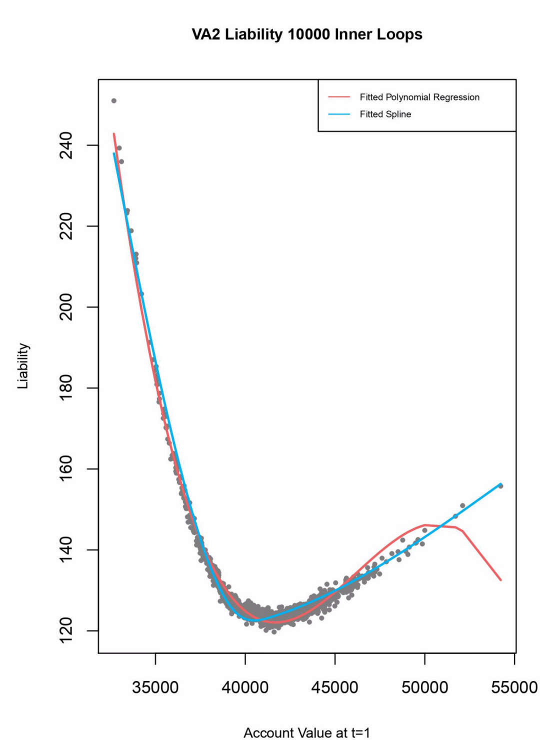
On comparing the result we get using Splines with the result we get using Polynomial Regression, we observe better results by using Splines. In policy 1, as the relationship between predictor and the response variable is not highly non-linear, both, polynomial regression and splines give good approximations. However, as the non-linearity in the relationship increases (similar to what we observe in policy 2), Splines give better results. This is because in Polynomial Regression, predictors impose a global structure on the response variable. Due to this, it is not flexible enough to capture the dependency over the entire range. Oftentimes, to gain more flexibility, we do use higher degree. But this extra flexibility can cause undesirable results at the boundaries as observed in policy 2. Splines on the other hand, avoid imposing such a global structure. To provide flexibility, they rather increase the number of knots due to which they can provide a better reasonable fit.

Observing this, one might argue to use a higher degree polynomial while fitting the model. However, generally, it is unusual to use degree greater than 3 or 4. This is because, for larger degrees, the polynomial curve can become overly flexible and can show wild behaviours especially at the tails. Moreover, with the group of policies in a portfolio being highly non-homogeneous, it is not practical to fix a degree that can generalize the results for all the policies. As a result, to generalize the results, Spline is preferred over Polynomial Regression.









Appendix:

VA Liability calculation:

Having seen the individual valuations, we will now incorporate all of this into a single approach and see how VA liability calculation is performed for policies with multiple riders. To make the scenarios more realistic, we will also incorporate survival probabilities here. As the simulations done in the later section of the report are based on the Surrogate Modelling Approach by Lin and Yang (2020), we will follow the methodology used in their paper for VA liability calculations.

Initially, under that approach, for all policies, account values are projected from t=0 to t=1 under real world scenarios. Here, we denote $G_0^{W/D/A}$ as the guarantee bases at time t=0 for withdrawal (W), death (D) and accumulation (A) rider respectively. Whereas, we denote $G^E = wA_0$ as the guaranteed annual withdrawal amount.

Following that $\rho^{D/A}$ is the roll-up rate of the death and accumulation benefit guarantee bases. At specific times t, we will be using superscripts $-$ and $+$ to indicate the values of a quantity prior to and after the benefit payments are made at that time.

So, withdraw guarantee base at time t=1 before payments are made is $G_1^{D/A-} = (1+\rho^{D/A})G_0^{D/A}$. If the guarantee is of ratcheting-type, then $\rho = 0$ in that case.

The account value after withdrawals is

$$A_1^+ = \max(A_0\frac{R_1}{R_0} - wA_0, 0)$$

Here, R_1 is the value of the underlying asset at time t=1 which follows real world dynamics based on regime switching log normal model while R_0 is the value at t=0. Following this, the guarantee withdrawal base is reduced by amount withdrawn $G_1^{W+} = G_0^W - wA_0$. After the withdraw benefit payments are made, the death and accumulation guarantee bases will be reduced by the withdraw amounts as well and hence will be:

$$G_1^{D/A+} = G_1^{D/A-} - wA_0$$

If the guarantee is of ratcheting type, then the guarantee base will be adjusted after the withdrawals are made. In that case, it would be:

$$G_1^{D/A+} = \max(G_1^{D/A-} - wA_0, A_1^+)$$

The second step, after account values are projected to t=1 is to calculate the fair value of embedded guarantees. Here, we will see how the cash flows evolve under a single risk-neutral path in a general setting from time $[t^+, t+1^+]$. At $t+1^-$, we make similar adjustments as above for all the guarantee bases.

$$G_{t+1}^{W-} = G_t^{W+}$$
$$G_{t+1}^{D/A-} = G_t^{D/A+}(1+\rho^{D/A})$$

Again, if the bases are ratcheting type, $\rho = 0$. From t^+ to $t+1^-$, the account value becomes $A_{t+1}^- = A_t^+ \frac{S_{t+1}}{S_t}$

Here, S_i represents account value at time i, which changes in accordance with the risk-neutral model unlike the change from t=0 to t=1 when the values were projected using real-world scenarios.

Following that, at time $t+1$, the policyholder withdraws $E_{t+1} = \min(G_{t+1}^{W-}, G^E)$ from the account. Hence,

$$A_{t+1}^+ = \max(A_{t+1}^- - E_{t+1}, 0)$$

In this case, the withdraw benefit can be expressed as put option's payoff with strike E_{t+1}

$$W_{t+1} = \max(E_{t+1} - A_{t+1}^-, 0)$$

Following that, the guaranteed bases are adjusted by the withdrawal amount and so they become:

$$G_{t+1}^{W/D/A+} = G_{t+1}^{W/D/A-} - E_{t+1}$$

If death happens during this, death benefit is made at the end of the period and is:

$$D_{t+1} = \max(G_{t+1}^{D-} - A_{t+1}^-, 0)$$

After the payments are made at t+1, if the policy is still enforced and if the benefit bases are of ratcheting type, then:

$$G_{t+1}^{D/A+} = \max(G_{t+1}^{D/A-} - E_{t+1}, A_{t+1}^+)$$

For GMAB rider, the benefit at maturity is

$$M_T = \max(G_T^{A+} - A_T^+, 0)$$

Hence, at the end, the liability of a guarantee rider for policyholder p for k-th world scenario is the present value of all the projected withdraw, death and maturity benefits:

$$\boxed{L_t = \sum_{s=t+1}^{T} {}_{s-t-1}p_{x+t}(1-q_{x+s-1})W_s e^{-r(s-t)} + \sum_{s=t+1}^{T} {}_{s-t-1}p_{x+t}(q_{x+s-1})D_s e^{-r(s-t)} + M_T e^{-r(T-t)}}$$

References:

[1] Agarwal, Girdhar G.; Studden, W. J. Asymptotic Integrated Mean Square Error Using Least Squares and Bias Minimizing Splines. Ann. Statist. 8 (1980), no. 6, 1307–1325. doi:10.1214/aos/1176345203. https://projecteuclid.org/euclid.aos/1176345203

[2] Areal, N., Rodrigues, A. & Armada, M.R. On improving the least squares Monte Carlo option valuation method. Rev Deriv Res 11, 119 (2008). https://doi.org/10.1007/s11147-008-9026-x

[3] Bacinello, A., Millossovich, P., Olivieri, A., & Pitacco, E. (2011, May 27). Variable annuities: A unifying valuation approach. https://www.sciencedirect.com/science/article/pii/S0167668711000618

[4] Facts + Statistics: Annuities. (n.d.). https://www.iii.org/fact-statistic/facts-statistics-annuities

[5] Gan, G., &amp Lin, X. (2015, March 20). Valuation of large variable annuity portfolios under nested simulation: A functional data approach. https://www.sciencedirect.com/science/article/pii/S0167668715000219

[6] James, G., Witten, D., Hastie, T., & Tibshirani, R. (2017). Moving beyond linearity. In An introduction to statistical learning: With applications in R (pp. 265-297). New York, NY: Springer.

[7] Lin, X., & Yang, S. (2020, March 31). Efficient Dynamic Hedging for Large Variable Annuity Portfolios with Multiple Underlying Assets. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3550106

[8] Liu, Y. (2010). Pricing and Hedging the Guaranteed Minimum Withdrawal Benefits in Variable Annuities. https://uwspace.uwaterloo.ca/bitstream/handle/10012/4990/Liu_Yan.pdf?sequence=1

[9] Longstaff, F. A., & Schwartz, E. S. (2001). Valuing American Options by Simulation: A Simple Least-Squares Approach. Review of Financial Studies, 14(1), 113-147. doi:10.1093/rfs/14.1.113

[10] Marquardt, T. & Platen, Eckhard & Jaschke, Stefan. (2008). Valuing Guaranteed Minimum Death Benefit Options in Variable Annuities Under a Benchmark Approach.

[11] Secure Retirement Institute: 2019 Total Annuity Sales Reach Highest Levels Since 2008. (2020, February 18).

[12] Zhou, S., & Wolfe, D. (2000). ON DERIVATIVE ESTIMATION IN SPLINE REGRESSION. Statistica Sinica, 10(1), 93-108. www.jstor.org/stable/24306706