SAS® GLOBAL FORUM 2020 #SASGF

Paper 4101-2020

Pricing Life Insurance Contracts with Early Exercise Features Using Neural Networks

Helgard Raubenheimer, Neels Geldenhuys, Jason Tappan; Centre for Business Mathematics and Informatics, North-West University

ABSTRACT

This paper describes an algorithm based on neural networks to price life insurance contracts embedding American options. We focus on equity-linked contracts with surrender options and terminal guarantees on benefits payable upon death, survival, and surrender. We use the Monte Carlo approach to generate artificial sample paths of the processes. We then use the least squares neural networks regression estimates to estimate from this data the so-called continuation values, rather than the ordinary least squares (OLS) estimates. For the purpose of fitting the neural network, we use the NEURAL procedure in SAS/IML®. The results from this investigation showed that neural network regression estimates provide adequate improvements to the OLS estimates and are thus useful additions to pricing option embedded insurance products.

INTRODUCTION

According to Honegger and Mathis (2018), life insurance products has increased and given rise to the existence of multiple variations among fundamental policy structures as well as a wide array of add-on features associated with life insurance policies. As innovation within the whole insurance industry grew, so too did the life insurance realm evolve and gave rise to modern policies such as equity-linked contracts which offered insurers exposure to financial indices and consolidating investment performances. The desire to maintain competitiveness brought about policies with surrender and minimum guarantee features which prodded policyholder's perspective that insurance securities could be viewed as possible liquid investments. This induced an increase to use insurance products for saving prospects rather than traditional protection. Unit (equity) linked life insurance policies relate to where the contract benefit is linked directly to the market value of a reference portfolio.

A variant of the Black Scholes option pricing formula is utilized in pricing these contracts. However, a difference in valuation arises when it involves annual rate of return guarantees as well as when American-style sell back features are present. Modelling these contracts have attracted many interests starting in Jorgensen and Grosen (1997) and Jorgensen and Grosen (2001). Jensen et al. (2001) proposed a finite difference approach. The flexible and long maturity nature of life insurance policies prompted multiple exercise dates. Furthermore, surrenders are usually driven by a range of factors, e.g. policyholder's health deterioration or recovery. Policy pay-outs are also contingent on mortality-and-mobility rates, asset classes, interest rates etc. All these aspects, in conjunction with dependencies between asset generated returns and their means of distribution to the policyholder all contribute to the overall complexity of the contract. Bacinello (2003a,b) and Bacinello (2005) study this problem in the context of binomial trees. Bacinello (2008) and Bacinello et al. (2009) study this problem in the context of Monte Carlo simulation and the Least Squares Monte Carlo (LSMC) approach.

The establishment of these modern contracts with these features prompted heavy computational expertise. This warranted an investigation into effective measures to price life insurance contracts with early exercise features.

PROBLEM DISCERPTION

The life insurance contract, as described by Bacinello et al. (2009), consider an individual aged χ at time 0 when entering the contract, with maturity $T > 0$. The contract provides a lump sum benefit F_T^s at time T upon survival or a benefit F_t^d at time $t \in (0, T]$. In case t coincides with the individual's time of death, denoted by τ. The contracts investigated by Bacinello et al. (2009) focus on equity-linked products, meaning that the benefits are linked to the performance of a reference fund. Besides providing death and survival benefits, these policies may allow policyholders to exit the contract before maturity. Policies is said to be lapsed, if no payment is provided upon withdrawal. If the policy is surrendered at time t a lump sum F_t^w is paid.

Depending on whether we consider survival, death or surrender benefits the terminal guarantee at time t has the following form:

$$F_t^e = \max\left(\frac{S_t}{S_0}, exp(\kappa_e t)\right),$$

where $e = s, d, w$ (survival, death or surrender). κ_e is the specified guaranteed return on the different benefit payments and $S = (S_t)_{t\geq 0}$ the market value of the reference fund to which policy payments are linked. As in Bacinello et al. (2009), this paper limits its attention to the case of single premium policies and ignores performance smoothing.

Define θ, a random variable, the time at which the policyholder decides to terminate the contract, with $\theta < t \wedge T$. The value of the random variable θ depends on the evolution of market and demographic conditions, which at any given time, make the surrender value more or less attractive with respect to staying in the contract. The cumulated benefits paid by the contract up to a time t are then given by

$$G_t(\theta) = F_T^s 1_{\tau > T, T \leq T \wedge \theta} + F_\tau^d 1_{\tau \leq t \wedge T \wedge \theta} + F_\theta^w 1_{\theta \leq t, \theta < \tau \wedge T},$$

where T the maturity time and τ individual's time of death as above. The price focused on in this paper is simply the expected present value of terminal guarantee $G_t(\theta)$.

Bacinello et al. (2009) propose an algorithm based on the LSMC method to price the contract. We propose to use the same Monte Carlo approach to generate the artificial sample paths of these price processes, and then we use the least squares neural networks (LSNN) regression estimates to estimate from this data the so-called continuation values instead of using OLS Optimization of the prices were subject to constraints of run time and had to orbit close to the benchmark prices set forth by Bacinello et al. (2009).

VALUATION FRAMEWORK

As in Bacinello et al. (2009), the financial and demographic risk factors are modelled as follow. The term structure of interest rates is modelled by a standard Cox-Ingersoll-Ross model with dynamics

$$dr_t = \zeta_r(\delta_r - r_t)dt + \sigma_r\sqrt{r_t}\, dZ_t^r,$$

where $\zeta_r, \delta_r, \sigma_r > 0$ and Z_t^r a standard Brownian motion. For the market value of the reference fund, the stochastic exponential $S = \exp(Y)$ is considered, where Y evolves according to

$$dY_t = (r_t - 0.5K_t - \lambda_Y\mu_Y)dt + \sqrt{K_t}\left(\rho_{SK}dZ_t^K + \rho_{Sr}dZ_t^r + \sqrt{1-\rho_{Sr} - \rho_{SK}}dZ_t^S\right) + dJ_t^Y,$$

with compound Poisson process dJ_t^Y. The process dJ_t^Y has a jump arrival rate of $\lambda_Y > 0$ and normally distributed jump sizes with mean μ_Y and standard deviation $\sigma_Y > 0$. The correlation coefficients ρ_{SK} and ρ_{S_r} satisfy $\rho_{SK}^2 + \rho_{S_r}^2 \leq 1$. The stochastic volatility are modelled with mean-reverting dynamics

$$dK_t = \zeta_K(\delta_K - K_t)dt + \sigma_K\sqrt{K_t}\,dZ_t^K,$$

where $\zeta_K, \delta_K, \sigma_K > 0$ and Z_t^K a standard Brownian motion. The process (Z_t^K, Z_t^r, Z_t^S) is a three-dimensional Brownian motion independent of the jump process J^Y.

For the intensity of mortality, a left continuous version of the process

$$d\mu_t = \zeta_\mu(m(t) - \mu_t)dt + \sigma_\mu\sqrt{\mu_t}\,dZ_t^\mu + dJ_t^\mu,$$

where $m(.)$, $\zeta_\mu, \sigma_\mu > 0$ and Z_t^μ a standard Brownian motion and J^μ is a compound Poisson process with jump arrival rate of $\lambda_\mu > 0$ and exponential jump sizes with mean $\gamma_\mu > 0$, independent of Z^μ. The processes (Z^r, Z^S, Z^K, J^Y) and process (Z^μ, J^μ) are independent. The function $m(.)$ above is obtained by fitting a Weibull intensity, given by $m(t) = c_1^{c_2}c_2(\chi + t)^{c_2 - 1}$ to implied survival probabilities.

Bacinello et al. (2009) defines the arrival of death is by setting

$$\tau = \inf\{t: \Gamma_t > \xi\},$$

with $(\Gamma_t)_{t>0}$ a nondecreasing process expressed as $\Gamma_t = \int_0^t u_s ds$ and ξ a random variable independent Γ and exponentially distributed with parameter one.

The financial market is characterized by an investment fund S, as introduced above, and a money market account yielding a instantaneous risk-free rate $(r_t)_{t>0}$. $B_t = \exp\left(\int_0^t r_s\, ds\right)$ is defined as the proceeds from investing one unit of money at time 0 in risk-free deposits and rolling over the proceeds until time t.

Bacinello et al. (2009) shows that the time t value $V_t(\theta)$, of the insurance contract terminated at the random time θ under no-arbitrage is given by the usual risk-neutral formula

$$V_t(\theta) = B_t E^{\mathbb{Q}}\left[\int_t^\infty B_u^{-1} dG_u(\theta)\,|\mathcal{F}_t\right],$$

where $G_t(\theta)$ is given as above represents the cumulated benefits provided by the contract up to time u with $\mathbb{Q}$ the risk-neutral probability measure and filtration $\mathcal{F}_t$.

THE ALGORITHM

Below we describe the LSCM algorithm for prising the life insurance contract as proposed by Bacinello et al. (2009) and give some informatization in the implementation in PROC IML. We then describe the LSNN approach and how this is implemented.

LEAST SQUARES MONTE CARLO (LSMC)

The original algorithm used by Bacinello et al. (2009) is described below. Assume that M simulated paths have been generated for the state variables process $X = (\mu, r, Y, K)$ (as defined above) and a unit exponential random variable ξ. We also simulate the state variable process over the time grid $\mathbb{T} = \{t_0, \ldots, t_n\}$, where n the number of periods in which we divide the interval $[0, T]$ and $t_i = \frac{i}{n}T$. For each simulation m $(m = 1, \ldots, M)$, we let ξ^m denote the simulated value of the exponential random variable. Define $\tau^m = \min\{t\epsilon\mathbb{T}: \Gamma_t^m > \zeta^m\}$ the simulated time of death where $\Gamma_t^m = \int_0^t \mu_s^m ds$ and $(\mu_t^m)_{t\in\mathbb{T}}$ the simulated paths of the stochastic intensity. If the insured survives at maturity T we set $\tau^m = \infty$. For each $t \in \mathbb{T}$ such that $t \le \tau^m$, we denote by r_t^m, S_t^m and K_t^m the simulated values of the short rate, stochastic volatility and reference fund. The simulated discount factors are computed as $v_{t,s}^m = B_t^m (B_s^m)^{-1}$, (with $t < s$ and $t,s \in \mathbb{T}$) and the simulated benefits payable on death $\left(F_t^{d,m}\right)$ survival $(F_t^{s,m})$ and surrender $(F_t^{w,m})$. We denote $\theta^{*,m}$ as the optimal stopping time for simulation m.

The valuation algorithm proposed by Bacinello et al. (2009) follows these steps:

Step 1. (Initialization):

For $m = 1, \dots, M$, if $\tau^m \le T$ set $\theta^{*,m} = \tau^m$ and $P_{\theta^{*,m}}^m = F_{\theta^{*,m}}^{d,m}$, otherwise set $\theta^{*,m} = T$ and $P_{\theta^{*,m}}^m = F_{\theta^{*,m}}^{s,m}$.

Step 2. (Backward iteration):

For $j = n-1, n-2, \dots, 1$:

1. (Continuation Values): Let $I_j = \{1 \le m \le M : \tau^m > t_j\}$ and, for each $m \in I_j$ set $C_{t_j} = P_{\theta^{*,m}}^m \times v_{t_j,\theta^{*,m}}^m$.
2. (Regression) Regress the continuation values $C_{t_j} = (C_{t_j}^m)_{m\in I_j}$ against $(e(X_{t_j}^m))_{m\in I_j}$ to obtain $\hat{C}_{t_j}^m = \beta_j^* \cdot e\left(X_{t_j}^m\right)$ for each $m \in I_j$, where $e(.) \doteq \left(e_1(.), \dots, e_H(.)\right)'$ a finite set of functions taken from a suitable basis $\{e_1, \dots, e_H, \dots\}$ and $\beta_j^* \doteq \left(\beta_j^{*1}, \dots, \beta_j^{*1H}\right)'$ is the optimal vector obtained by the least squares regression. If $F_{t_j}^{w,m} > \hat{C}_{t_j}^m$ then set $\theta^{*,m} = t_j$ and $P_{t_j}^m = F_{t_j}^{w,m}$.

Step 3. (Initial value): Compute the single premium of the contract by

$$V_0^* = \frac{1}{M} \sum_{m=1}^{M} P_{\theta^{*,m}}^m \times v_{0,\theta^{*,m}}^m.$$

The objective of the OLS algorithm implemented in the Monte Carlo context is to provide a path-wise approximation to the optimal stopping rule that maximizes the value of the American option.

IMPLEMENTATION OF LSCM IN PROC IML

The LSCM algorithm, as described above, was implemented in PROC IML. The sample code of the implementation is supplied with the paper. An extract of the IML code is given below:

```
Pt = Ft_s[,n+1]#(tau = T);

if ( ( (do(0,T,dt)+J(M,n+1,0) = tau)[,1:n] )[+] >= 1 ) then
      Pt[loc(Pt=0)] = Ft_d[loc((do(0,T,dt)+J(M,n+1,0) = tau)[,1:n] ||
                        J(M,1,0))];

do i = n-1 to 1 by -1;
      tj = i*dt;
      set = loc(tau > tj);
      Ctj = Pt[set] # ( Bt[set,i+1] / ((Bt[set,]) [loc((do(0,T,dt) +
             J(M,n+1,0))[set,] = theta[set])]) );
```

```
    Xtj = rt[set,i+1] || Yt[set,i+1] || Kt[set,i+1] || mut[set,i+1] ;
    Xreg = J(nrow(Xtj),1,1);
    do l = 1 to 4;
         Xreg = xreg || Xtj[,l][,#];
         do j = l to 4;
              Xreg = xreg || Xtj[, (l || j )][,#];
              do k = j to 4;
                   Xreg = xreg || Xtj[, (l || j || k)][,#];
              end;
         end;
    end;
    ares = ginv(Xreg)*Ctj;
    Ctj_h = Xreg*ares;
    tmp = set[loc((Ft_w[set,i+1] > Ctj_h))];
    Pt[tmp] = Ft_w[tmp,i+1];
    theta[tmp] = tj;
end;
```

After setting up the dataset $(e(X_{t_j}^m))_{m\in I_j}$ in the matrix β_j^* is estimated as $ginv\left(e\left(X_{t_j}^m\right)\right)\cdot C_{t_j}$.

LEAST SQUARES NEURAL NETWORKS (LSNN)

Kohler et al. (2010) uses least squares neural networks regression to price high-dimensional American options. As in Kohler et al. (2010), this paper substitutes OLS in Step 2 above with least squares neural networks. The investigation relied on utilising PROC NEURAL to implement the neural networks.

In Step 2-2 in the algorithm above is change to follow Kohler et al. (2010) to estimate the continuation values $C_{t_j} = (C_{t_j}^m)_{m\in I_j}$ by neural networks with $k\in\mathbb{N}$ hidden neurons and a sigmoid function σ. The choice of k the number of hidden neurons will be data-driven by using sample splitting. Let $\beta n > 0$ (which is chosen such that $\beta n \to \infty$ $(n\to\infty)$) and let $F_k(\beta n)$ be a class of neural networks defined by

$$F_k(\beta n) = \left\{\sum_{i=1}^{k} c_i\sigma\left(\alpha_{j,i}^* \cdot e\left(X_{t_j}^m\right)\right) + c_0\right\},$$

where σ a sigmoid function and $\sum_{i=1}^{k}|c_i| < \beta_n$.

To choose of parameter k, of the neural networks regression estimate, fully automatically, the sample is split. Thus the dataset $(e(X_{t_j}^m))_{m\in I_j}$ is subdivided in a learning sample $\left(I_j^l\right)$ and validation sample $\left(I_j^v\right)$ with equal proportions. Define for a given k the regression estimate of $\hat{C}_{t_j}$ as:

$$\hat{C}_{t_j}^{l,k} = \arg\min_{f\in F_k(\beta n)}\left(\frac{1}{\dim(I_j^l)}\sum_{m\in I_j^l}\left|f\left(e(X_{t_j}^m\right) - C_{t_j}^l\right|^2\right).$$

The parameter k is chosen the minimize the empirical L_2 risk on the testing sample:

$$\hat{k} = \arg\min_{k}\left(\frac{1}{\dim(I_j^l)}\sum_{m\in I_j^l}\left|\hat{C}_{t_j}^{l,k} - C_{t_j}^l\right|^2\right).$$

IMPLEMENTATION OF LSNN

The LSNN algorithm, as described above, was also implemented by making use of the SUBMIT statement in PROC IML. The sample code of the implementation is supplied with the paper. The LSMC codes above is only modified by replacing the following line of code:

```
ares = ginv(Xreg)*Ctj;
```

with:

```
samp = J(ncol(set),1,0);
call randgen(samp,"uniform");
call sortndx(ndx,samp,{1});

dev = Ctj[ndx[1:int(ncol(set)/2)],] ||
Xreg[ndx[1:int(ncol(set)/2)],2:35];
val = Ctj[ndx[int(ncol(set)/2)+1:ncol(set)],] ||
Xreg[ndx[int(ncol(set)/2)+1:ncol(set)],2:35];

score = Ctj|| Xreg[,2:35];

create dev from dev;
append from dev;
close dev;

create val from val;
append from val;
close val;

create score from score;
append from score;
close score;

submit;
      %nn_LSCM;
endsubmit;

use netout;
read all var _num_ into Ctj_h;
close netout;
```

Where the dataset $(e(X_{t_j}^m))_{m \in I_j}$ is split into a training (dev) and validation data set (val). The macro definition ***%nn_LSCM*** contains the PROC NEURAL statement:

```
proc neural data=dev
            dmdbcat=cat_dev
            VALIDATA=val
            random=12345 ;

      input col2-col35 / level=interval id=int;
      target col1  / level=interval id=ter;

      hidden &i / id=hu ACT=LOGistic COMBINE=LINEAR;

      netoptions objective = dev;

      train OUTEST= est ESTITER = 1 MAXITER = 50;
```

```
    score data=score outfit=netfit out=netout  ;

run;
```

See the sample code on how the PROC NEURAL statement is looped in the macro definition to find the optimal number of hidden units.

NUMERICAL EXAMPLE

To test the proposed algorithm, we implement the example by Bacinello et al. (2009). The reference insured is a male aged $\chi = 40$ at time 0. The contract has maturity $T = 15$ years and provides terminal guarantees on survival, death and surrender benefits. The valuation algorithm is applied with polynomial basis functions of order 3, yielding $H = 34$. To replace the American claim with a Bermudan claim, the time dimension is discretized by using a time step (in years) which is called Backward Discretization Step (BDS). To simulate the state variable process X, a time grid finer than $\mathbb{T}$ is employed called the Forward Discretization Step (FDS), which is the length in years of each time interval in the finer grid.

Different values for the minimum rates guaranteed upon both death and survival ($\kappa = \kappa_d = \kappa_s$) as well for the surrender guarantee (κ_w) are considered. The case of $\kappa_w > \kappa$ is considered for numerical purposes only. 140 seed iterative process was used to create and store a data frame of final contract value data points. Each iteration generates 19000 artificial sample paths for the risk factors defined above. These datasets were then used for graphical and formal analysis. The results obtained relied on initial values and fixed variable values as provided by Bacinello et al. (2009). These values can be seen in Table 1 below.

Table 1: Parameter values for Monte Carlo simulation

	r	K	S	μ
$BDS = 1$	$r_0 = 0.05$	$K_0 = 0.04$	$S_0 = 100.00$	$\mu_0 = m(0)$
$FDS = 0.01$	$\zeta_r = 0.60$	$\zeta_K = 1.50$	$\rho_{SK} = -0.70$	$\zeta_\mu = 0.50$
	$\delta_r = 0.05$	$\delta_K = 0.04$	$\rho_{Sr} = 0.00$	$\sigma_\mu = 0.03$
	$\sigma_r = 0.03$	$\sigma_K = 0.40$	$\lambda_Y = 0.50$	$\lambda_\mu = 0.10$
			$\mu_Y = 0.00$	$\gamma_\mu = 0.01$
			$\sigma_Y = 0.07$	$c_1 = 83.70$
				$c_2 = 8.30$
				$\chi = 40$

With reference to the study conducted by Bacinello et al. (2009), baseline results of the average price and standard errors for different guarantee scenarios were procured. By contextualising the problem statement to an argumentative setting, we aim to conclude which technique between the OLS and neural network processes yields the most optimal and parsimonious results.

The results show adequate accuracy in line with previously obtained results for both OLS and neural network processes. The neural network also resulted in a standard error (see Table 3 below) substantially lower than that obtained by using OLS (see Table 2 below) in most of the scenarios tested. It is worth noting that the OLS results differ in terms of the standard error since Bacinello et al. (2009) used variance reduction techniques not included in this investigation.

Table 2: Contract values using OLS

		κ_s / κ_d		
		0%	2%	4%
κ_w	0%	114.703 (0.062)	115.833 (0.071)	122.701 (0.067)
	2%	118.144 (0.065)	118.396 (0.069)	122.842 (0.078)
	4%	123.730 (0.067)	123.852 (0.067)	124.548 (0.065)
	6%	135.286 (0.070)	135.417 (0.072)	135.336(0.078)

Table 3: Contract values using a neural network

		κ_s / κ_d		
		0%	2%	4%
κ_w	0%	113.202 (0.059)	115.115 (0.064)	122.109 (0.077)
	2%	117.154 (0.062)	117.485 (0.060)	122.399 (0.074)
	4%	123.278 (0.067)	123.583 (0.065)	124.228 (0.080)
	6%	136.877 (0.062)	137.159 (0.062)	137.693 (0.054)

CONCLUSION

In this paper, we have briefly discussed the application of neural networks to pricing equity-linked contracts with surrender options and terminal guarantees on benefits payable upon death, survival, and surrender. The results showed that neural network regression estimates provide adequate improvements to the OLS estimates and are thus useful addition to pricing option embedded insurance products.

Throughout the analysis, however, we noted a violation of one of the above-stated optimality prerequisites where the neural network procedure had a much greater run time. The authors investigated the implementation of the algorithm in SAS VIYA using DATA steps to simulate the state variable process X and employing PROC NNET to implement the neural networks. By taking advantage of the CAS server, a considerable decrease in run time is observed. Further areas of research could focus on the employment of other machine learning techniques such as gradient boosting.

REFERENCES

BACINELLO. A.R. (2003a). Fair valuation of a guaranteed life insurance participating contract embedding a surrender option. The Journal of Risk and Insurance. 70(3). 461-487.

BACINELLO. A.R. (2003b). Pricing guaranteed life insurance participating policies with annual premiums and surrender option. North American Actuarial Journal. 7(3). 1-17.

BACINELLO. A.R. (2005). Endogenous model of surrender conditions in equity-linked life insurance. Insurance: Mathematics & Economics. 37(2). 270-296.

BACINELLO. A.R. (2008). A full Monte Carlo approach to the valuation of the surrender option embedded in life insurance contracts. in: C. Perna. M. Sibillo (Eds.). Mathematical and Statistical Methods for Insurance and Finance. Springer. 19-26.

BACINELLO. A.R. BIFFIS. E. & MILLOSSOVICH P. (2009). Pricing life insurance contracts with early exercise features. Journal of Computational and Applied Mathematics. 233. 27-35.

GROSEN. A. & JORGENSEN. P.L. (1997). Valuation of early exercisable interest rate guarantees. The Journal of Risk and Insurance. 64 (3). 481-503.

GROSEN. A. & JORGENSEN. P.L. (2000). Fair Valuation of Life Insurance Liabilities: The Impact of Interest Rate Guarantees. Surrender Options. and Bonus Policies. Insurance: Mathematics and Economics. 26(1). 37–57.

JENSEN. B.. JORGENSEN. P.L. & GROSEN. A. (2001). A Finite Difference Approach to the Valuation of Path Dependent Life Insurance Liabilities. The Geneva Papers on Risk and Insurance Theory. 26:57-84.

HONEGGER. R. & MATHIS. C. (2008). Duration of life insurance liabilities and asset liability management. 17th International AFIR Colloquium. Rome. Italy - October 1-3. 2008. Available at: http://www.actuaries.org/AFIR/Colloquia/Rome/Honegger_Mathis.pdf [accessed 2019-01-17].

KOHLER. M.. KRZYZAK. A. & TODOROVIC. N. (2010): Pricing of High-Dimensional American Options by Neural Networks. Math. Finance. 20(3). 383–410.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Helgard Raubenheimer
Centre for Business Mathematics and Informatics, North-West University
Helgard.Raubenheimer@nwu.ac.za
http://natural-sciences.nwu.ac.za/bmi