Solvency II

Methods

Numerical Experiments

Future works

Accuracy and Robustness of Machine Learning Methods in SCR Estimation

Gilberto Castellani[3] Ugo Fiore[2] Zelda Marino[2] Luca Passalacqua[3] Francesca Perla[2] Salvatore Scognamiglio[1] Paolo Zanetti[2]

[1]Department of Economic and Legal Studies, Parthenope University of Naples

[2]Department of Management and Quantitative Studies, Parthenope University of Naples

[3]Department of Statistical Sciences, Sapienza University of Rome

Insurance Data Science Conference
(ETH Zurich - 14 June 2019)

Nested Monte Carlo Simulation

Let $t \in [0, \infty)$ be the evaluation date and let:

- A_t, L_t: market-consistent values of assets and liabilities
- $NAV_t = A_t - L_t$: Net Asset Value (NAV)

Solvency Capital Requirement

The Solvency Capital Requirement (SCR) determines the amount of capital that ensures that an undertaking will be able to meet its obligations over 1 year with a probability of at least 99.5 %.

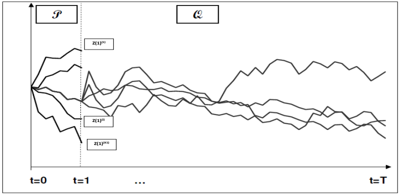
$$SCR = (\mathbf{E}[NAV_1] - NAV_1^{0.5\%})\, v(0,1)$$

Let $\mathbf{z}^{(i)}$ the i-th *state process* realisation at time $t = 1$ under $\mathbb{P}$, $i = 1, \ldots, N$, then

$$L^{(i)}(1) = \mathbf{E}^{\mathbb{Q}}\left[\sum_{t=1}^{T} \frac{Y_t}{B_t} \,\middle|\, \mathbf{z} = \mathbf{z}^{(i)}\right]$$

with Y_t the future liabilities cash flows, B_t the numéraire process.
The random variables $L^{(i)}(1)$ can be used in order to estimate the NAV_1 (PDF) and the SCR that involve **Nested Monte Carlo Simulation**

$$\hat{L}^{(i)}(1) = \frac{1}{K}\sum_{k=1}^{K}\sum_{t=1}^{T} \frac{Y_t^{(i,k)}}{B_t^{(i,k)}}$$



Source: Bauer, D., Reuss, A. and Singer, D. (2012). ASTIN Bulletin, 42(2), 453-499.

PROBLEM: **high computational cost** ($N \times K$ total simulations)

Methods

We investigate the potential of **Machine Learning based models** - *Support Vector Regression*, *Deep Learning Networks* - to reduce the computational burden, comparing with the benchmark *Least Square Monte Carlo* method.

1. **Least Squares Monte Carlo** (LSMC):

$$\hat{L}^{(i)}_{LSMC} = \sum_{j=1}^{m} \alpha_j \pi(\mathbf{z}^{(i)})$$

Issues:

- choice of type of the orthogonal polynomials used, choice of degree of polynomial, the curse of dimensionality.

2. **Support Vector Regression** (SVR):

$$\hat{L}^{(i)}_{SVR} = \mathbf{w}^T \varphi(\mathbf{z}^{(i)}) + b$$

Issues:

- choice of kernel used, hyperparameters tuning.

3. **Deep Learning Networks** (DLN):

$$\hat{L}^{(i)}_{DLN} = \phi(\mathbf{v}^T \mathbf{z}^{(i)} + b)$$

Isuues:

- choice of architecture used, hyperparameters tuning.

COMPUTATIONAL COST: $(\mathbf{N}' \times \mathbf{K}') << (\mathbf{N} \times \mathbf{K})$.

The analysis has been carried out on data that mimic typical profit sharing life insurance policy with **seven risk sources** (inter alia: interest rate, exchange rate, inflation and equity risk).
Using DISAR®[1] a full nested MC approach was carried out and 2 datasets were produced:
Calibration Dataset with N' = 10.000 and K' = 1.000;
Testing Dataset with N = 100.000 and K = 10.000.

About the methods:

- LMSC: Hermite, Legendre and Laguerre orthogonal polynomials are tested. On ground of parsimony we select **Laguerre polynomials with degree equal to 3**.
- SVR: Linear, polynomial and Radial basis kernel are tested. The best performance was obtained using **Radial Basis kernel**.
- DLN: we test several feed-forward architectures. On ground of parsimony we select a **feed-forward with three hidden layers**.

All experiments have been run on a 8-core Linux server with R, using **keras**, **mlr**, **parallel**, **parallelMap**, **orthopolynom**, **e1071**

[1] see Castellani, G., Passalacqua, L., 2010. Applications of distributed and parallel computing in the Solvency II framework, Euro-Par 2010.

Results: accuracy

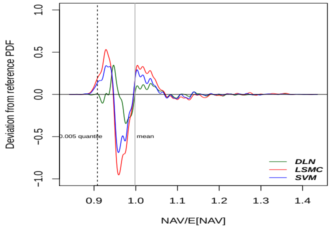


Table: Normalised Root Mean Square Error (NRMSE) from testing distribution.

Model	NRMSE
LMSC	0.002443
SVR	0.002394
DLN	0.002277

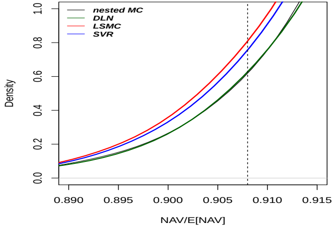


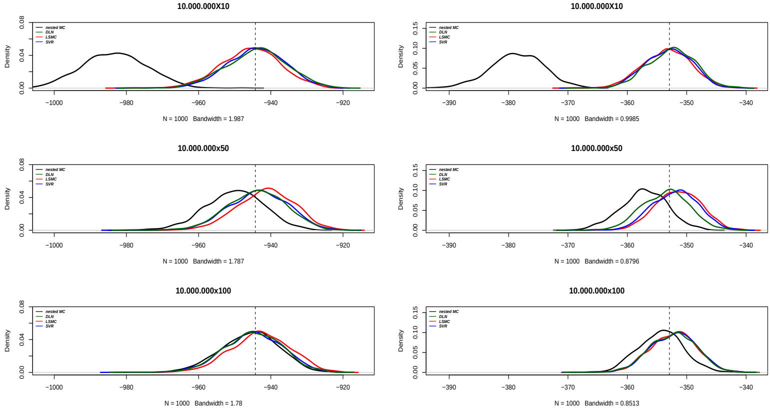
Table: Relative Error in SCR/(Best Estimate) estimation.

Model	Rel. error
LMSC	2.34%
SVR	1.88%
DLN	0.04%

Castellani, G., Fiore, U., Marino, Z., Passalacqua, L., Perla, F., Scognamiglio, S., and Zanetti, P. (2018). An investigation of Machine Learning Approaches in the Solvency II Valuation Framework. Available at SSRN 3303296

Results: robustness in quantile estimation

The analysis of the variability of quantile estimation has been carried out considering two synthetic insurance portfolios mimic a single-premium policy (left) and a recurring-premium policy (right), respectively.



The results show that when few risk-neutral simulations are used, the nested MC estimator is biased, instead LSMC, SVR and DLN give better results.

Future works

In future works we will deal with:

1. increasing number of risk sources;
2. exploring different neural networks;
3. introducing stochastic models for technical risks such as mortality risk and lapse risk;
4. investigating the convergence of all three approaches increasing the number of risk-neutral simulations in the training set;
5. using High Performance Computing in the training of ML-based methods.

For advice or comments:

salvatore.scognamiglio@uniparthenope.it