Deep Neighbor Embedding for Evaluation of Large Portfolios of Variable Annuities

Xiaojuan Cheng[1,2], Wei Luo[3](✉), Guojun Gan[4], and Gang Li[3]

[1] Xi'an Shiyou University, Shaanxi, China
xiaojuan.cheng@tulip.org.au
[2] Xinjiang Technical Institute of Physics and Chemistry,
Chinese Academy of Sciences, Urumqi, China
[3] Deakin University, Geelong, Australia
{wei.luo,gang.li}@deakin.edu.au
[4] University of Connecticut, Storrs, CT, USA
guojun.gan@uconn.edu

Abstract. Variable annuities are very profitable financial products that pose unique challenges in risk prediction. Metamodeling techniques are popular due to the significant saving in computation time. However, the current metamodeling techniques still have a low valuation accuracy. One key difficulty is the selection of a small number of contracts that optimally represent the whole portfolio. In this paper, we propose a novel and highly effective method for selecting representative contracts. At the center of this method is a deep neighbor embedding that supports robust clustering of the contracts in a portfolio. The embedding is a low-dimensional representation that respects similarities among contracts in both contract-specific features and their historical performance, achieved through abstract representation in a deep neural network. Empirical results show that the proposed model achieves significant improvement in valuation accuracy, often 10 times or more accurate compared with the popular Kriging-based model.

Keywords: Variable annuity · Neighbor embedding · Deep transfer learning

1 Introduction

A variable annuity (VA) is a retirement insurance product that in the U.S. alone generated 92.9 billion sales in 2018, according to *Insured Retirement Institute* (IRI). With the product's popularity, comes the significant and complex risk due to the return guarantees embedded in variable annuities. The traditional risk estimation method based on Monte-Carlo simulation involves the prohibitive computational cost and fails to handle large policy portfolios in most insurance companies. Insurance companies are turning to machine learning for the valuation of large portfolios of variable annuity policies. In particular, metamodeling

C. Douligeris et al. (Eds.): KSEM 2019, LNAI 11775, pp. 472–480, 2019.
https://doi.org/10.1007/978-3-030-29551-6_42

techniques [4,7] have been proposed to provide a portfolio-level risk estimate with a minimal amount of simulation.

With metamodeling, Monte Carlo simulation [8] is conducted only on a small number of representative policies in a large portfolio. The risk estimates on these representative policies are then used to infer the overall risk profile of the whole portfolio. Selecting the truly representative policies is crucial for successful metamodeling. Today, the most popular approach is to partition a portfolio into a number of policy clusters and then select the centroid policy from each cluster as a representative policy [4]. In the past, clustering is usually performed on a set of predefined features describing each policy and the policyholder. This choice of feature space has a number of potential issues, ranging from the presence of irrelevant features, mixed variable types, to the so-called "curse of dimensionality" when the feature set is large. In order to improve the accuracy and robustness of risk estimation, researchers have been actively searching for better ways to select representative policies [2,3,5]. Our earlier work [1] suggests that clustering on the deep-representation level greatly improves the accuracy.

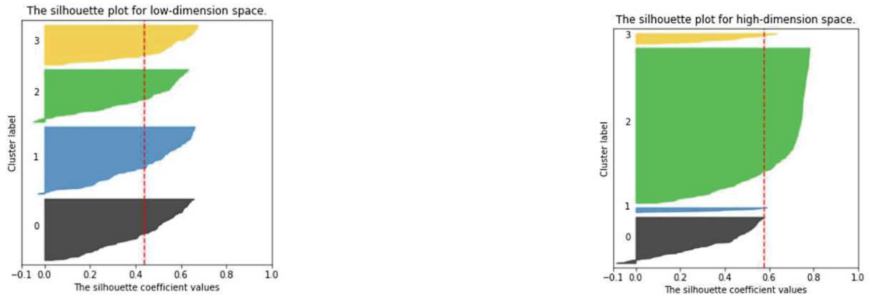
Metamodeling and the Challenges. The metamodeling of a large portfolio consists of two steps. First a small number of representative policies are selected from the portfolio using clustering-based approaches or sampling-based approaches and then valued via computationally intensive MCMC. Next, a regression model extrapolate the MCMC valuation to all policies in the portfolio. Selecting the truly representative policies is the key to the success of metamodeling.

The aforementioned clustering-based approaches perform clustering on a high-dimension space, which can result imbalanced clusters (see Fig. 1b). Figure 1 illustrates clearly a low-dimensional embedding of the deep representation can generate robust clusters. We can see that the silhouette value almost surpasses the average level in a low-dimensional embedding. However in the high-dimensional representation space, it's extremely nonuniform for each cluster. In this paper, we present a novel framework integrating neighbor embedding with deep transfer learning, which achieves a significant performance improvement.

Our Contributions. In this work, we propose a novel approach to select representative policies for variable annuities, extending our work in [1]. This approach significantly improves the quality of selected representative policies, via clustering over a low-dimensional embedding of the deep representation of the portfolio. To our knowledge, it is the first time that dimension reduction has been applied to metamodeling for variable annuities. Although the idea looks deceivingly simple, we show that a naive application of dimension reduction on the original feature space does not work. It is the integration of transfer learning and local-distance preserving embedding that results in the significant performance improvement shown in this paper.

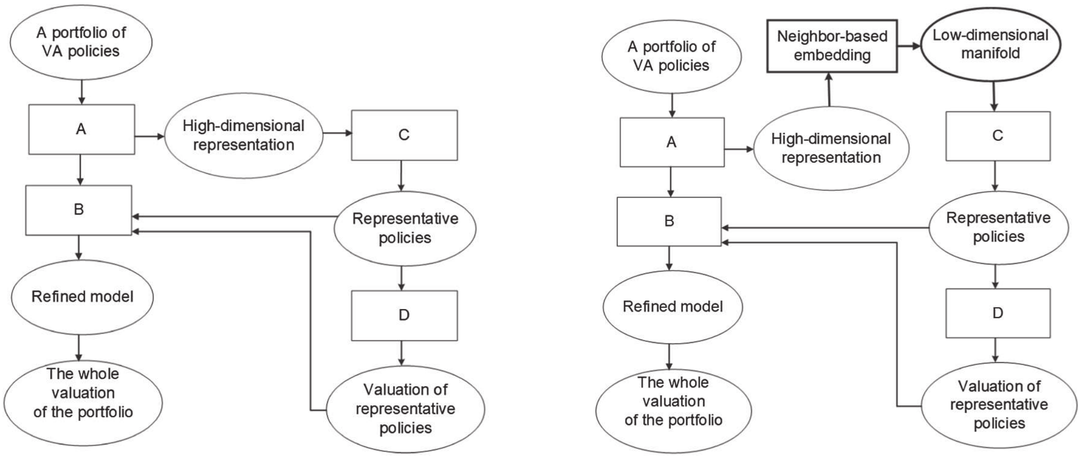
2 The Method: DR-TL

In this section, we present a *dimension-reduced transfer learning* (DR-TL) framework. Unlike TL framework in [1] performs clustering in a high-dimension space, Fig. 2 shows DR-TL framework performs clustering in a low-dimension embedding. The process consists of major steps as follows: (1) Fit a backbone deep neural network using historical Monte Carlo simulation results, then get a high-dimensional latent representation of policies from intermediate layer. (2) Apply the neighbor-based embedding to the deep representation to get a low-dimension manifold. (3) Clustering via the k-means algorithm in the low-dimension space to find representative contracts. (4) Run the Monte Carlo simulation for the valuation of representative contracts under the target market. (5) Fine-tuning



(a) Clustering in a low-dimensional space. (b) Clustering in a high-dimensional space.

Fig. 1. The silhouette plot for four clusters respectively in low-dimensional embedding and in high-dimensional representation space. The dotted red line reflects the average silhouette score of all the values. (Color figure online)


A portfolio of VA policies
A
High-dimensional representation
C
B
Representative policies
Refined model
D
The whole valuation of the portfolio
Valuation of representative policies
A portfolio of VA policies
Neighbor-based embedding
Low-dimensional manifold
A
High-dimensional representation
C
B
Representative policies
Refined model
D
The whole valuation of the portfolio
Valuation of representative policies

(a) TL framework (b) DR-TL framework

Fig. 2. The TL framework in [1] and the proposed DR-TL framework, with the additional neighbor embedding step to produce a low-dimensional manifold. **A**: Fit a model with historical simulation (different market conditions). **B**: Fine-tuning the fitted model by transfer learning method (updated market conditions). **C**: Data clustering. **D**: Monte Carlo simulation.

the backbone deep neural network model using representative policies and the corresponding valuation under the target market. (6) Use the tuned model to estimate the valuation of all policies in the portfolio.

2.1 Problem Formulation

Policy and Portfolio. A VA portfolio $\mathcal{P}$ often contains a large number N of individual policies c_i sold to different customers: $\mathcal{P} = \{c_i : 1 \le i \le N\}$. For each policy c_i, a feature vector x_i captures the characteristics of the policy and the customer. At the same time, each policy c has an expected risk y_i. Although y_i is unknown in general, it can be very well estimated by Monte Carlo simulation, though the simulation is usually very computationally extensive and can only be performed on a small sample from $\mathcal{P}$.

Metamodeling. Metamodeling attempts to estimate the aggregated risk $\overline{Y}$ of the whole portfolio $\mathcal{P}$ using only regression models to compensate the lack of complete Monte Carlo results. It involves two subtasks: selecting a subset $\mathcal{R}$ of K representative policies of the portfolio $\mathcal{P}$, and inferring the overall risk $\overline{Y}$ from the simulated results $\{y_i : c_i \in \mathcal{R}\}$ on the representative subset.

The second subtask is relatively straightforward: instead of directly predicting $\overline{Y} = \text{mean}\left(\{y_i : 1 \le i \le N\}\right)$, we can pursue the unbiased estimate of each y_i. This can be achieved through fitting a regression model on $\{(x_i, y_i) : c_i \in \mathcal{R}\}$. Our prior work has shown that a deep neural network pre-trained using historical data achieved the best prediction performance so far.

The first subtask of selecting an optimal subset remains largely unsolved, hence it is the focus of this work. One natural solution is to first cluster $\mathcal{P}$ based on $\{x_i\}$ and then select only the cluster centers (similar to medoids) [4]. The prior work shows that clustering directly on the features $\{x_i\}$ can be problematic, especially when $\{x_i\}$ contains discrete features and/or high-variance features irrelevant to y_i. Instead, we have shown that transfer learning which solves the second (regression) problem also provides an opportunity to better address the first (clustering) problem, as better clustering can be obtained on the deep representation of a pre-trained model.

Although deep representation in a pre-trained model leads to better clustering results, it also brings a new challenge: the high dimensionality of the deep representation may cause sensitive clusters and cluster centers. Therefore, this paper address the following research problem: with a limited computation budget on the Monte Carlo simulation for K policies, how to find the robust K clusters and cluster centers $\{c_{ij} : 1 \le j \le K\}$ that allow the most accurate estimation of $\overline{Y}$ given $\{(x_{ij}, y_{ij}) : 1 \le j \le K\}$.

2.2 Neighbour Based Embedding

As the hidden layer representation has a high dimension, directly performing clustering algorithm may result in unstable clusters as well as sensitive cluster centers. We use neighbor based embedding to map the high-dimension representations into a lower dimensional space without losing the local structures among policies.

There are many mature neighbor based embedding methods, such as t-SNE [9], ISOMap [11] and UMAP [10]. In this paper, we use UMAP due to its superiority in embedding stability and computational efficiency. The embedding aims to preserve the local structures among observations. Many nonlinear embedding methods follow a similar two-step process: generating weighted K-neighbor graphs followed by laying out the graph in a low-dimensional space.

In UMAP, the graph weights are derived from two weighted directed graphs defined by the similarity measure $w\left(\left(x_i, x_{i_j}\right)\right) = \exp\left(\frac{-\max\left(0, d\left(x_i, x_{i_j}\right) - \rho_i\right)}{\sigma_i}\right)$, where $d\left(x_i, x_{i_j}\right)$ is the pairwise distance in the feature space, and ρ_i, σ_i are normalising constants. The layout algorithm in UMAP is the widely used force-directed graph layout algorithm. Negative sampling is used for computational efficiency. In this paper, we set the embedding space dimension as 2. At the end of the embedding step, the hidden representations are all mapped to a 2-d space on which clustering will be carried out.

3 Experiment and Analysis

In this section, we evaluate the performance of the proposed DR-TL framework using both simulated datasets and standard benchmark datasets. Then we present the numerical results to show the superiority of low-dimensional embedding of the deep representation including both simulated data and benchmark dataset.

3.1 Data Description

Simulated Dataset. The simulated dataset is generated by `make_blobs` function in `sklearn` Python library. This data is used to simulate a portfolio of $1,000$ policies with 5 features from 20 clusters. The corresponding $Y = f(x)$ is made up of a random noisy in linear and nonlinear form. In the linear form, Y_1 and Y_2 simulate respectively two different market scenarios. Forming the dependent variable coefficients are $\{2, 1, 1, 3, 5\}$ and $\{5, 2, 3, 4, 2\}$. In the nonlinear form, $\hat{Y_2}$ simulating the target market consists of a quadratic polynomial and the coefficients are $\{2, 1, 3, 1, 2\}$.

VA Benchmark Dataset. The second dataset [6] is a large synthetic benchmark dataset used extensively in metamodeling [8]. The runtime used to create these synthetic datasets would be about 3 years if only a single CPU was used. The dataset contains $1,000$ risk-neutral paths, and each path contains $38,000$ VA policies. Each policy contains a feature vector of 34 predictors, a Greeks value and a central risk metric that is the target variable for metamodeling.

3.2 Performance Metrics

To evaluate the accuracy of the proposed model, we follow the strategy in [4] and use the following two validation measures: the *percentage error* (PE) at the portfolio level and R^2. Among which, the percentage error is defined as

$$PE(\mathcal{P}) = \frac{\sum_{c_i \in \mathcal{P}} (\hat{y}_i - y_i)}{\sum_{c_i \in \mathcal{P}} y_i}, \tag{1}$$

Table 1. PE of Kriging model and DR-Kriging model on simulated data.

# clusters	$PE \downarrow$ (linear)		$PE \downarrow$ (nonlinear)	
	Kriging model	DR-Kriging model	Kriging model	DR-Kriging model
$k = 10$	−0.402	**−0.303**	**0.167**	0.178
$k = 20$	**−0.084**	−0.292	**−0.091**	0.169
$k = 30$	**0.036**	0.152	0.191	**0.069**

Table 2. R^2 of Kriging model and DR-Kriging model on simulated data.

# clusters	$R^2 \uparrow$ (linear)		$R^2 \uparrow$ (nonlinear)	
	Kriging model	DR-Kriging model	Kriging model	DR-Kriging model
$k = 10$	0.604	**0.824**	0.252	**0.289**
$k = 20$	0.832	**0.904**	0.642	**0.693**
$k = 30$	**0.954**	0.830	**0.778**	**0.778**

where y_i describes the value of policy c_i in the portfolio $\mathcal{P}$ from the high-resolution Monte Carlo simulation. and $\hat{y}_i$ is the corresponding estimate from the neural network. On the other hand, R^2 is defined as

$$R^2 = 1 - \frac{\sum_{c_i \in \mathcal{P}} (\hat{y}_i - y_i)^2}{\sum_{c_i \in \mathcal{P}} (y_i - \mu)^2}, \tag{2}$$

where $\mu = \frac{1}{n} \sum_{c_i \in \mathcal{P}} y_i$ is the average Delta value.

From the above equations we can see, PE and R^2 are complimentary measurements for the valuation performance. While R^2 indicates the fitness at the policy level, PE directly reflects the accuracy at the portfolio level. For insurance companies with a large portfolio, the hedging is commonly used to manage the portfolio level risk. Therefore minimising PE, or equivalently maximising the portfolio accuracy, is the primary objective in this work.

3.3 Results

Performance of DR-Kriging Model on Simulated Data. First, we try a *dimension-reduced Kriging* (DR-Kriging) model, which applies dimension reduction directly on the feature space, instead of the deep representation space. On the simulated data, we compared the baseline Kriging model with the DR-Kriging model. Tables 1 and 2 show PE and R^2 achieved by the Kriging model and the DR-Kriging model with different number of clusters k. These results show that dimension reduction directly applied on the feature space doesn't improve the performance.

DR-TL on Simulated Data. Unlike DR-Kriging, DR-TL applies dimension reduction on the deep represenation. Table 3 shows PE achieved by the Kriging model and the DR-TL model on the simulated data. For different numbers of clusters, DR-TL model achieves smaller absolute PE values than the baseline kriging model. the superiority of DR-TL is particularly evident on the nonlinear data. For example, with 30 clusters, DR-TL reduced the PE from 0.038 to 0.009.

Table 3. PE of Kriging model and DR-TL model on simulated data.

# clusters	$PE \downarrow$ (linear)		$PE \downarrow$ (nonlinear)	
	Kriging model	DR-TL model	Kriging model	DR-TL model
$k = 10$	0.425	**0.035**	1.268	**−0.214**
$k = 20$	0.100	**0.024**	0.529	**0.017**
$k = 30$	0.048	**0.013**	0.038	**0.009**

Table 4. R^2 of Kriging model and DR-TL model on simulated data.

# clusters	$R^2 \uparrow$ (linear)		$R^2 \uparrow$ (nonlinear)	
	Kriging model	DR-TL model	Kriging model	DR-TL model
$k = 10$	0.604	**0.987**	**0.222**	0.004
$k = 20$	0.831	**0.999**	0.725	**0.804**
$k = 30$	0.953	**0.999**	0.816	**0.926**

Table 5. PE on benchmark data

# clusters	$PE \downarrow$		
	Kriging	TL	DR-TL
50	−0.168	0.152	**0.075**
80	−0.050	0.088	**0.032**
90	0.098	0.053	**0.006**
100	−0.108	0.043	**0.003**
200	−0.032	0.024	**0.003**

Table 6. R^2 on benchmark data

$R^2 \uparrow$		
Kriging	TL	DR-TL
0.187	0.139	0.162
0.273	**0.508**	0.245
0.304	**0.504**	0.281
0.222	**0.445**	0.430
0.365	0.485	**0.507**

Table 4 shows the R^2 results on simulated data. While the PE shows the accuracy at the portfolio level, the R^2 shows the accuracy at the individual policy level, which is not a primary concern for portfolio risk valuation.

DR-TL on Benchmark Data. Tables 5 and 6 show the results on the benchmark data [4]. In addition to the Kriging model, TL model from [1] is used as another baseline. Although Table 6 shows that DR-TL does not optimise the policy-level accuracy, Table 5 shows the proposed DR-TL method significantly outperformed the two baselines on PE, the primary optimisation target of portfolio metamodeling. First, a superior PE of 0.003 is achieved with as few as 100 representative policies. This is 30 times better than what can be achieved through the mainstream Kriging method. This is a new level of accuracy that, to our knowledge, has never been achieved through metamodeling with such a small training sample. Second, the high accuracy translates to great saving in terms of simulation time. If 90 representative policies are sufficient to reach a PE less than 1%, the predicted risk can potentially be recalculated in near real-time when an insurance company needs to adjust its product portfolio.

4 Conclusions

In this work, we have proposed a novel metamodeling framework integrating neighbor embedding with deep transfer learning. Although dimension reduction seems to be an obvious option to improve clustering stability, we have shown that when directly applied to policy-level features, dimension reduction fails to improve the valuation accuracy, and sometimes even results in higher PE. Building on the transfer-learning metamodels, the new framework can locate representative policies that are not only stable, but also relevant to the risk profile.

The proposed framework enables a large portfolio to be valued in high accuracy with fewer than 50 training examples in metamodeling. This translates to 10 times or higher saving in the computational cost on simulation, and we believe that this significant improvement of the valuation accuracy can potentially bring disruptive change to how the insurance industry performs the risk evaluation.

Acknowledgement. This work was supported by the International Cooperation Project of Institute of Information Engineering, Chinese Academy of Sciences under Grant No. Y7Z0511101, and Guangxi Key Laboratory of Trusted Software (No KX201528).

References

1. Cheng, X., Luo, W., Gan, G., Li, G.: Fast valuation of large portfolios of variable annuities via transfer learning. In: Nayak, A.C., Sharma, A. (eds.) PRICAI 2019. LNCS, vol. 11672, pp. 716–728. Springer, Cham (2019)
2. Gan, G.: Application of data clustering and machine learning in variable annuity valuation. Insur. Math. Econ. **53**(3), 795–801 (2013)
3. Gan, G.: Application of metamodeling to the valuation of large variable annuity portfolios. In: Proceedings of the Winter Simulation Conference, pp. 1103–1114 (2015)
4. Gan, G., Huang, J.X.: A data mining framework for valuing large portfolios of variable annuities. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD 2017, pp. 1467–1475 (2017)

5. Gan, G., Valdez, E.: An empirical comparison of some experimental designs for the valuation of large variable annuity portfolios. Dep. Model. **4**, 382–400 (2016)
6. Gan, G., Valdez, E.: Nested stochastic valuation of large variable annuity portfolios: Monte Carlo simulation and synthetic datasets. Data **3**(3), 31 (2018)
7. Gan, G., Valdez, E.A.: Regression modeling for the valuation of large variable annuity portfolios. North Am. Actuarial J. **22**(1), 40–54 (2018)
8. Kleijnen, J.P.C.: A comment on blanning's "metamodel for sensitivity analysis: The regression metamodel in simulation". Interfaces **5**(3), 21–23 (1975)
9. van der Maaten, L., Hinton, G.: Visualizing data using t-SNE. JMLR **9**, 2579–2605 (2008)
10. McInnes, L., Healy, J., Melville, J.: UMAP: uniform manifold approximation and projection for dimension reduction. arXiv preprint arXiv:1802.03426 (2018)
11. Tenenbaum, J.B., De Silva, V., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. Science **290**(5500), 2319–2323 (2000)