Fast Valuation of Large Portfolios of Variable Annuities via Transfer Learning

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

[1] Xi'an Shiyou University, Shaanxi 710065, China
xiaojuan.cheng@tulip.org.au
[2] Guangxi Key Laboratory of Trusted Software,
Guilin University of Electronic Technology, Guilin, China
[3] Deakin University, Geelong, VIC 3216, Australia
{wei.luo,gang.li}@deakin.edu.au
[4] University of Connecticut, Storrs, CT, USA
guojun.gan@uconn.edu

Abstract. Variable annuities are important financial products that result in 100 billion sales in 2018. These products contain complex guarantees that are computationally expensive to value, and insurance companies are turning to machine learning for the valuation of large portfolios of variable annuity policies. Although earlier studies, exemplified by the regression modelling approach, have shown promising results, the valuation accuracy is unsatisfying. In this paper, we show that one main cause for the poor valuation accuracy is the inefficient selection of representative policies. To overcome this problem, we propose a novel transfer-learning based portfolio valuation framework. The framework first builds a backbone deep neural network using historical Monte Carlo simulation results. The backbone network provides a valuation-driven representation for selecting the policies that best represent a large portfolio. Furthermore, the transferred network provides a way to adaptively extrapolate from these representative policies to the remaining policies in the portfolio. By overcoming a major difficulty faced by the popular Kriging model, the need of matrix inversion, the transferred network can handle a large number of representative policies to sufficiently cover a diverse portfolio.

Keywords: Variable annuity · Deep representation · Transfer learning

1 Introduction

A variable annuity (VA) is a retirement insurance product. Guarantees embedded in variable annuities have complex risk profiles and many insurance companies manage the risk through dynamic hedging [11], which results in a large portfolio of individual policies. In order to simulate the performance of dynamic hedging and determine the stochastic reserve of VA products, insurance companies rely on nested Monte Carlo (MC) simulations [14]. However, the computation of MC simulations for a large VA portfolio is time-consuming because each VA policy

A. C. Nayak and A. Sharma (Eds.): PRICAI 2019, LNAI 11672, pp. 716–728, 2019.
https://doi.org/10.1007/978-3-030-29894-4_57

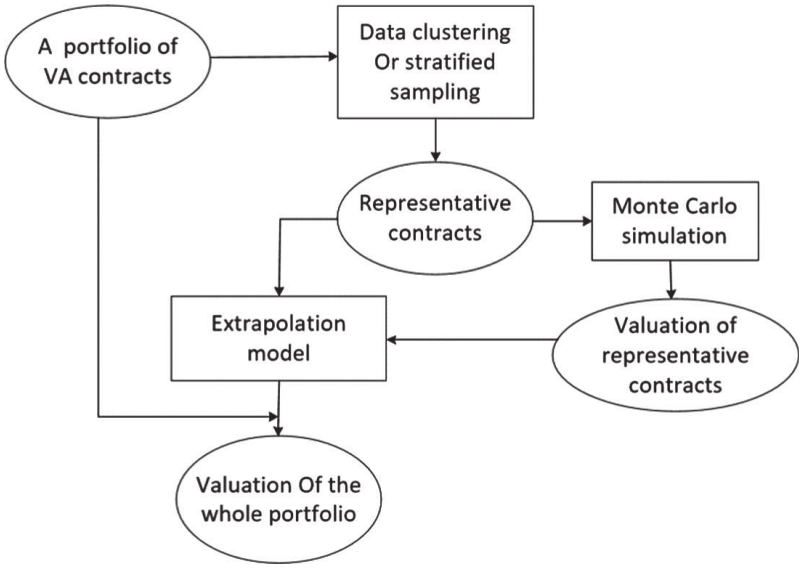


Fig. 1. Meta-modeling approach for VA portfolio valuation.

needs to be projected over many scenarios for a long time horizon. For example, [9] implemented the nested MC simulations in Java, and they calculated the partial dollar deltas along 1,000 real-world paths at annual steps. For a portfolio of 38,000 VA policies, the calculation would take about 2.97 years to complete. In practice, a portfolio needs to be reevaluated under multiple market assumptions. Repeating the MC simulation for each market assumption is simply infeasible.

Recently, meta-modeling approaches [16] have been proposed in the literature [5,10,20] to address the aforementioned computational problem. Figure 1 shows that the meta-modeling approach involves three main steps: First, we select a small number of representative VA policies by a clustering algorithm or a sampling method; Then, we run the MC simulation to generate the valuation of representative VA policies; Finally, we choose an appropriate meta-model (e.g., linear regression) to estimate the valuation of all policies in the large portfolio based on the valuation of representative policies. Meta-modeling approaches can significantly reduce the runtime because only a small number of representative policies are valuated by the high-accuracy MC simulation method, and the whole portfolio of policies are valuated by the meta-model. Although the meta-model may produce less accurate valuation for each policy, the aggregated valuation for the whole portfolio can achieve a low overall error due to the nature of dynamic hedging [5]. This attractive feature makes meta-modeling a popular VA valuation framework.

However, current VA meta-modeling approach relies on efficient selection of policies representing the whole portfolio. When a portfolio is large and with diverse policies, this task is challenging and usually results in poor valuation accuracy (See Sect. 3).

Our Contributions. We propose a new framework for accurate valuation of a large VA portfolio. It builds on the commonly adopted clustering-based

Table 1. Example predictor variables used for clustering VA policies.

Variable	Description
`gender`	Gender of the policyholder
`age`	Age of the policyholder
`productType`	Product type of the VA policy
`gmwbBalance`	Guaranteed minimum withdrawal benefit (GMWB) balance
`gbAmt`	Guaranteed benefit amount
`FundValue`i	Account value of the ith fund, for $i = 1,2,\ldots,n$
`ttm`	Time to maturity in years

approach. At the centre of our framework is the novel idea of clustering not at the predictors themselves, but at a deeper representation guided by the target performance of policies. This is done in a principled approach using deep neural network motivated by the *information bottleneck* [17] principle (see Sect. 4.2 for details).

The proposed method for selecting representative policies greatly improves the coverage of the diverse policies in a dynamically hedged portfolio. This contributes to superior performance in portfolio risk valuation. In addition to an abstract representation to improve clustering, the deep neural network provides a way to quickly re-estimate VA valuations under varying market assumptions. Extensive empirical evaluations have confirmed that our framework provides more accurate VA estimates, which also implies reduced dependency on the computational expensive Monte Carlo simulation. Finally, our framework addresses a major challenge faced by the state-of-the-art Kriging model, the need to compute matrix inversion which inhibits the use of a moderately bigger number of representative policies.

This paper is organized as follows. Section 2 reviews the related work, and Sect. 4 proposes the transfer-learning framework. The details of experiments and result analysis are presented in Sect. 5. Finally, in Sect. 6 we conclude the paper.

2 Related Work

During the past five years, a number of research papers on meta-modeling for VA valuation have been published [2,3,5–8,10,12,13,20]. In [2] and [6], the k-prototype algorithm was used to select representative VA policies and the Kriging model was used as the meta-model. To address the drawback that the k-prototype algorithm is not efficient for selecting a moderate number (e.g., 200) of representative VA policies, [3] proposed the Latin hypercube sampling (LHS) to select representative policies. In [5], a scalable clustering algorithm called the truncated fuzzy c-means (TFMC) algorithm was used to select representative policies. In [7], several methods for selecting representative VA policies were compared. The authors found that the clustering method and the LHS method

produce similar results, and both are better than other methods such as random sampling.

In [12] and [20], neural networks were employed for the valuation of large VA portfolio. In [13], the valuation of large VA portfolios was formulated as a spatial interpolation problem. In [8], the authors studied the use of copula to model the dependency of partial dollar deltas and found that the use of copula does not improve the prediction accuracy of the meta-model. In [10], the authors used the GB2 (generalized beta of the second kind) distribution to model the fair market values. In [4], the author considered interactions between VA policy features in linear models and found that linear models with interaction terms can produce accurate predictions.

Among the meta-models considered in the aforementioned papers, the Kriging model is one of the top performers in terms of accuracy. Therefore in this paper, the Kriging model is used as the baseline for evaluating our proposed framework. To describe the ordinary Kriging model, let $\mathbf{z}_1$, $\mathbf{z}_2$, …, $\mathbf{z}_s$ be the representative VA policies. For every $j = 1, 2, \ldots, k$, let v_j be the fair market value of $\mathbf{z}_j$ that is calculated by Monte Carlo simulation. Then the fair market value of the VA policy $\mathbf{x}_i$ in the portfolio is estimated as follows:

$$\hat{y}_i = \sum_{j=1}^{k} w_{ij} \cdot v_j, \tag{1}$$

where $w_{i1}, w_{i2}, \ldots, w_{ik}$ are the Kriging weights obtained by solving the following linear equation system [15]:

$$\begin{pmatrix} V_{11} & \cdots & V_{1k} & 1 \\ \vdots & \ddots & \vdots & \vdots \\ V_{k1} & \cdots & V_{kk} & 1 \\ 1 & \cdots & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} w_{i1} \\ \vdots \\ w_{ik} \\ \theta_i \end{pmatrix} = \begin{pmatrix} D_{i1} \\ \vdots \\ D_{ik} \\ 1 \end{pmatrix}. \tag{2}$$

In the above equation, θ_i is a control variable used to make sure the sum of the Kriging weights is equal to one,

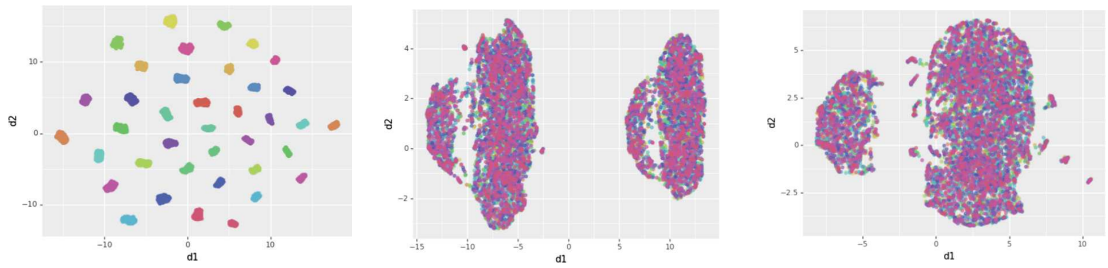
$$V_{rs} = \alpha + \exp\left(-\frac{3}{\beta} D(\mathbf{z}_r, \mathbf{z}_s)\right), \quad r, s = 1, 2, \ldots, k, \tag{3}$$

and

$$D_{ij} = \alpha + \exp\left(-\frac{3}{\beta} D(\mathbf{x}_i, \mathbf{z}_j)\right), \quad j = 1, 2, \ldots, k, \tag{4}$$

where $D(\cdot, \cdot)$ denotes the Euclidean distance, and both $\alpha \geq 0$ and $\beta > 0$ are parameters.

One major drawback of the Kriging model is that the computational cost for large k can be inhibitive, due to the need for matrix inversion.



(a) All variables including `productType` and `Gender`. Distinct clusters are formed for different product types.

(b) With `productType` removed and `Gender` kept. Two genders lead to two nearly identical clusters. But better mixture is achieved in each cluster.

(c) With both `Gender` and `productType` removed. The artefact due to `Gender` is also removed.

Fig. 2. t-SNE visualization of policies in a portfolio, forming clusters from which representative policies are selected.

3 Challenges in Representative Policy Selection

The success of meta-modeling relies on a set of well-balanced representative policies. However, finding such representative policies in a large portfolio remains some challenges. We will illustrate these challenges for clustering-based meta-modeling, which also apply to the sampling-based approach.

In the clustering-based approach, the representative policies are chosen from cluster centroids. The clustering of policies are based on a bag of variables assumed to be potentially predictive of the policy performance. These may include variables related to the policy holder and those related to the products themselves. Some example variables are shown in Table 1. Clearly until the simulation is completed, we do not actually know whether or how much these variables can predict the policy performance. Adding to this indiscriminating use of variables, the clustering also runs on bare categorical variables (e.g., `gender` and `productType`). Such categorical variables, especially those with a large number of levels, can create artificial clusters aligned mostly with their levels. Figure 2 demonstrates the inherent challenge of selecting 'representative' policies solely based on independent variables, which may or may not be predictive of target variable (policy valuation in this case). In other words, noisy or irrelevant policies-level features may impose undue influence on the clustering results, leading to unreliable 'representative' policies. Clearly the above difficulties of finding truly representative policies also apply to the sampling-based approach. The problem needs to be addressed for meta-modeling to achieve more reliable valuation.

4 Method

In this section, we present the transfer-learning (TL) based framework. Figure 3 shows the major steps in our framework, and the difference with Fig. 1

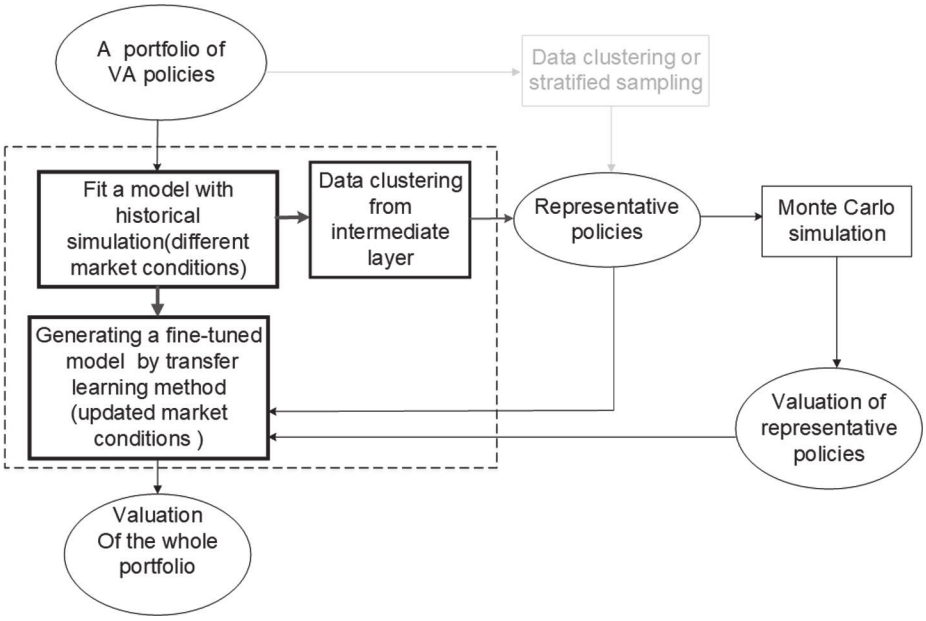


Fig. 3. The proposed model. The dashed line highlights the new components introduced. A deep neural network is trained using historical data or data from an approximate (simplified) simulation (Sect. 4.1). The trained network provides both a regulated space for robust policy clustering (Sect. 4.2) and a base model that can be transferred for extrapolation (Sect. 4.3).

(the traditional approach) has been highlighted with bold arrows in Fig. 3. In general, the proposed transfer-learning (TL) framework consists of the following five major steps.

1. Fit a multi-layered (deep) model based on a large number of historical simulations, under a potentially different market scenarios.
2. Obtain feature representations from an intermediate layer, which also forms a manifold of the portfolio, then use a data clustering algorithm to find a small number of representative policies.
3. With the configurations of the target market, run the Monte Carlo simulation for the valuation of representative policies.
4. Fine-tune the pre-trained model using simulation results of representative policies.
5. Use the transferred model to value all policies in the portfolio.

4.1 Build a Deep Neural Network Using Historical Data

This step builds a deep neural network that provides both a representation for clustering and a base model for transfer learning. Figure 4 shows the network architecture. To train such a network, we exploit available historical simulation data for similar VA products, potentially under a different set of market assumptions. When such historical simulation data is not available, we rely on the fact

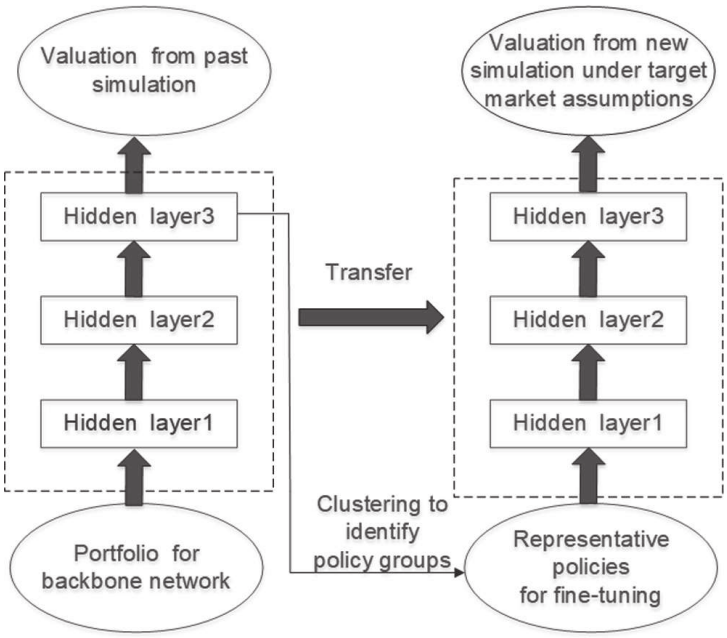


Fig. 4. The transfer learning architecture

that Monte Carlo simulation can be simplified with a much lower resolution to trade simulation accuracy for efficiency. A network trained on such a *proxy* target variable is often sufficient as a base model. With the proxy training labels, we train a dense network with three hidden layers.

4.2 Clustering on Hidden-Layer Representation

With the neural network trained, we perform the following steps to obtain the representative contracts from a portfolio.

1. Feed contacts to the network and obtain the hidden-layer representations.
2. Perform clustering on the obtained representations.
3. From the generated clusters, retrieve the cluster centers, which will become the representative policies.

Performing clustering on deep features can overcome the problem of inefficient clustering as shown in Fig. 2, because performing unsupervised learning on such features is a highly effective technique used by many deep learning practitioners. For example, deep representation was recently used to improve robustness in video anomaly detection [19]. Such representation can be extracted by deep neural network, which can transform input signals through multiple hidden layers to output layer. More specifically, when model training begins, the network receives an input X, and successively processes it through hidden layers, where the output of previous layer is the input of next layer. The closer the hidden layer is to the output layer, the more relevant features can be captured.

In our setup, the deep representation can be viewed as the result of regulating input features using the (proxy) target variable. Formally, it can be

explained via the *information bottleneck* principle. The recent work by Tishby and Zaslavsky [18] provides a formal structure for understanding the latent representations in terms of information processing. The idea of *information bottleneck* principle is that a network rids noisy input data of extraneous details as if by squeezing the information through a bottleneck, preserving information in the data that is relevant to the outputs. As can be seen in Fig. 2, clustering on the original inputs cannot distinguish relevant information from irrelevant information.

More formally, assume an input random variable $X \in \mathcal{X}$, and an output random variable $Y \in \mathcal{Y}$, given a join distribution $p(X, Y)$, the *relevant information* is defined as the mutual information $I(X; Y)$, where we assume statistical dependence between X and Y. In this case, we can capture relevant features by a compressed mapping of input variable X that discards the information irrelevant to Y.

In a multi-layer network, the hidden layer representation H provides an information compression of X guided by Y. In terms of the mutual information, neural network training tries effectively to minimise $I(X; H)$ and maximise $I(H; Y)$.

4.3 Transfer Learning

The previous step produces a deep network that maps each policy to a proxy measurement of its valuation. It forms a base model that can recalibrate using the high-resolution Monte Carlo simulation results on representative policies under the target market condition.

Let (P) be the portfolio of policies as shown in Fig. 1. The pre-trained network can be viewed as a function $f(c; \theta^s)$ minimising $\sum_{c \in \mathcal{P}} L(f(c; \theta^s), y_c^s)$, where y_c^s is the valuation used for pre-training. With the set $\mathcal{R}$ of the representative policies, we fine-tune the network so that θ^s is replaced by θ^t that minimises $\sum_{c \in \mathcal{R}} L(f(c; \theta^t), y_c^t)$, where y_c^t is the valuation generated by high-resolution Monte Carlo simulation.

5 Experiment and Analysis

Due to the demanding computational requirement of Monte Carlo simulation, our experiment will be based on existing simulation results for a large VA portfolio under five different sets of market assumptions. We will treat the first set of simulation results as given and use it to train a deep neural network. From the remaining four sets of results, we will simulate the process of re-valuation of the representative policies under those market assumptions. They also provide the ground truth for evaluating the valuation accuracy of the transferred model.

5.1 Data Description

To evaluate the performance of our transfer-learning framework, we follow [5] and use a synthetic portfolio. The portfolio contains 38,000 synthetic variable

annuity policies, described by 34 features including 2 categorical features (See Table 1).

Five sets of deltas have been generated using Monte Carlo simulation as in [5] under different market assumptions. Figure 5 shows a histogram of the first set of deltas. We use this set of deltas to simulate the available historical data for training the backbone network. The remaining four sets of deltas will be used as the ground-truths for evaluating the transfer-learning model.

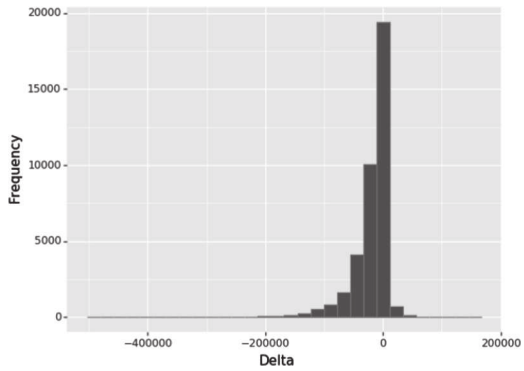


Fig. 5. A histogram of deltas in a portfolio under the first application market. The wide range of deltas reflects the diverse policies in a portfolio with dynamic hedging. It is crucial that the selected representative policies provide sufficient coverage of such a diverse portfolio.

5.2 Performance Metrics

To evaluate the accuracy of the proposed model, we follow the strategy in [5] and use the following two validation measures: the percentage error at the portfolio level and R^2. The percentage error and R^2 is respectively defined as

$$PE(\mathcal{P}) = \frac{\sum_{c_i \in \mathcal{P}} (\hat{y}_i - y_i)}{\sum_{c_i \in \mathcal{P}} y_i}, R^2 = 1 - \frac{\sum_{c_i \in \mathcal{P}} (\hat{y}_i - y_i)^2}{\sum_{c_i \in \mathcal{P}} (y_i - \mu)^2} \tag{5}$$

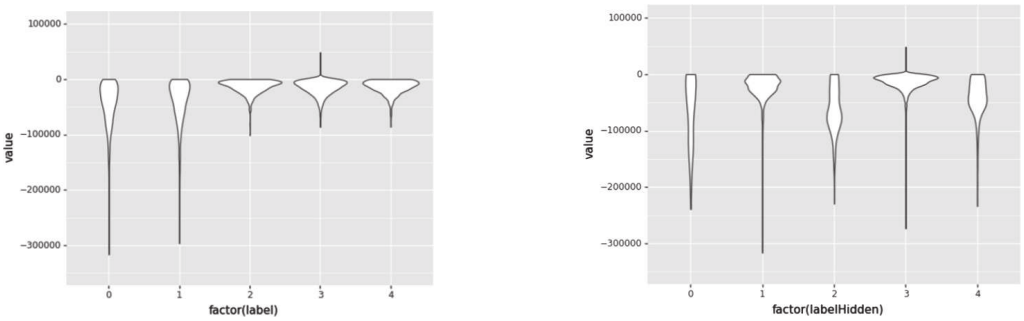
where y_i describes the value of policy c_i in the portfolio $\mathcal{P}$ from the high-resolution Monte Carlo simulation. And $\hat{y}_i$ is the corresponding estimate from the neural network, $\mu = \frac{1}{n} \sum_{c_i \in \mathcal{P}} y_i$ is the average Delta value.

From the above equations we can see, PE and R^2 are complimentary measurements for the valuation accuracy. While R^2 measures the fitness at the policy level, PE directly measures the accuracy at the portfolio level. Therefore minimising PE is our primary objective.

5.3 Baseline Models

To verify the performance of proposed transfer-learning framework, two baseline models are set. One is the meta-model in [5]. In that model, the TFCM++

algorithm is used to obtain k cluster centres as representative policies. After obtaining the Monte Carlo simulated deltas for these k representative policies, Kriging is performed to extrapolate the delta values to other policies in the portfolio. Please see [5] for more details.



(a) clustering on the input space (policy-specific features).

(b) clustering on the hidden-layer representation.

Fig. 6. clustering on the input space (policy-specific features).

To demonstrate the value of transfer learning, we also use the neural network (NN) model trained directly on the representative policies as another baseline. For simplicity, we use the same representative policies selected by the backbone network, and corresponding deltas are acquired via the MC simulation. However, no fine-tuning is applied, the neural network starts with random parameter initialisation and is directly trained using the small number of representative policies under a target market assumption.

5.4 Implementation Details of the Proposed Model

The deep network. Using the first set of deltas, we train a three-layer densely connected network. From the third hidden layer, we obtain the representation of the policies and perform clustering using K-means to obtain the representative policies. The trained network is saved as the base network for transfer learning under a different set of market assumptions.

Transfer learning. For each of the remaining four sets of market assumptions, we obtain the deltas for representative policies. These additional deltas were used to fine-tune the saved basic network. Detailed illustrations are shown in Fig. 4.

5.5 Results

Quality of Representative Policy Selection. To verify that the hidden layer provides a better representation for selecting representative policies. We randomly sample $5,000$ points on the input space and the representation space respectively, and then group them into 5 clusters. The corresponding delta value

range and distribution are shown in Fig. 6. Some similar clusters are presented in Fig. 6a. For example, the leftmost two clusters have nearly identical distributions, and similarly for the remaining three clusters, which suggests that the clustering on the input features will result in redundant representative policies, and consequently inefficient use of the Monte Carlo simulation. In contrast, clusters in Fig. 6b have distinct distributions, and they will likely lead to more distinct representative policies.

Portfolio Estimation Accuracy. To demonstrate the superiority of proposed model, we compare the TL model with Kriging model and NN model when $k = 100$, 200 and 400. When k gets bigger, the baseline Kriging model becomes infeasible due to the need for inverting a large matrix. Table 2 shows the accuracy of three models. As we can see in this table, in each model, as the number of clusters increases, PE reduces but R^2 improves, which indicates that the larger the number of representative policies, the higher the prediction accuracy. On the other hand, in each setting, the accuracy of TL model is always the highest among these models. For example, when $k = 100$, PE of Kriging model is 0.115 whereas TL model's drops to 0.043. Similar trends can be observed in other clusters, too. Moreover, with the increase of k, the advantage of the TL model is more remarkable. The low R^2 for the Kriging model suggests a poor model fit. This is not surprising in views of the redundant clusters shown in Fig. 6a.

Overall, the transfer-learning framework outperforms the Kriging model and the vanilla deep neural network in terms of the valuation accuracy.

Computing Cost. Table 3 shows the runtime of major steps of Kriging model and TL model. The majority of the run time is still spent on generating Monte Carlo simulation for the representative policies. In general, transfer-learning framework does not take longer than the SoTA Kriging model, if a backbone network is available. The fine-tuning step is faster than Kriging, especially when the number of clusters k increases, because it avoids the need for matrix inversion. The training of the backbone network took 54.90 s, a constant that is independent of k.

Overall, the transfer-learning framework achieves improved accuracy (measured by PE) and shorter runtime (due to the avoidance of matrix inversion is more pronounced). The differences are more pronounced as the number of clusters k gets larger. For example, from Tables 2 and 3, when $k = 400$, PE of Kriging model is 0.035, while TL model's surprisedly drops to 0.001. This is achieved with a shorter run-time than Kriging model. Therefore, the proposed framework can have greater advantages in both estimation accuracy and computation time when a portfolio has a greater diversity and requires more representative policies for sufficient coverage.

Table 2. Accuracy comparison of three models. k denotes the number of representative policies.

$k = 100$				$k = 200$			$k = 400$		
	Kriging	NN	TL model	Kriging	NN	TL model	Kriging	NN	TL model
$\downarrow PE$	0.115	0.056	**0.043**	0.074	0.036	**0.024**	0.035	0.003	**0.001**
$\uparrow R^2$	0.324	0.437	**0.445**	0.392	0.452	**0.485**	0.446	0.577	**0.661**

Table 3. Runtime of the proposed TL framework and the baseline Kriging model.

	$k = 100$		$k = 200$		$k = 400$	
	Kriging	TL model	Kriging	TL model	Kriging	TL model
Clustering	1.63	**1.46**	3.35	**3.16**	8.10	**7.82**
Monte Carlo[a]	722.34	722.34	1,444.68	1,444.68	2,889.36	2,889.36
Kriging[b]/Fine-tuning	3.07	**2.65**	7.30	**2.86**	14.79	**3.88**
Total	727.04	**726.45**	1455.33	**1450.70**	2912.25	**2901.06**

[a,b] Estimations derived from the results reported in [5].

6 Conclusions

We have proposed a new framework to address two challenges that current meta-modeling approaches face in a large portfolio of VA policies: inefficient selection of representative policies and the need for matrix inversion in Kriging. Incorporating the principles of information bottleneck and transfer learning, the framework achieves empirically validated improvement on the representative policy selection and the policy re-valuation under varying marketing assumption. Furthermore, by avoiding matrix inversion in the popular Kriging model, the proposed framework is able to handle a large number of representative policies, which is critical for sufficient coverage of a diverse portfolio.

The current work can potentially be extended along several dimensions. In particular in [1], we show that the clustering can be performed in a space of reduced dimension, which can result in further improvement of the valuation accuracy.

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.: Deep neighbor embedding for evaluation of large portfolios of variable annuities. In: Douligeris, C., Karagiannis, D., Apostolou, D. (eds.) KSEM 2019. LNCS (LNAI), vol. 11775, pp. xx–yy. 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). https://doi.org/10.1016/j.insmatheco.2013.09.021
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.: Valuation of large variable annuity portfolios using linear models with interactions. Risks **6**(3), 71 (2018). https://doi.org/10.3390/risks6030071
5. Gan, G., Huang, J.: 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, pp. 1467–1475 (2017). https://doi.org/10.1145/3097983.3098013
6. Gan, G., Lin, X.S.: Valuation of large variable annuity portfolios under nested simulation: a functional data approach. Insur. Math. Econ. **62**, 138–150 (2015)
7. Gan, G., Valdez, E.A.: An empirical comparison of some experimental designs for the valuation of large variable annuity portfolios. Depend. Model. **4**(1), 382–400 (2016). http://ssrn.com/abstract=2830879
8. Gan, G., Valdez, E.A.: Modeling partial greeks of variable annuities with dependence. Insur. Math. Econ. **76**, 118–134 (2017). http://ssrn.com/abstract=2844509
9. Gan, G., Valdez, E.A.: Nested stochastic valuation of large variable annuity portfolios: Monte Carlo simulation and synthetic datasets. Data **3**(3), 31 (2018). https://doi.org/10.3390/data303003110.3390/data3030031
10. Gan, G., Valdez, E.A.: Regression modeling for the valuation of large variable annuity portfolios. North Am. Actuarial J. **22**(1), 40–54 (2018). http://ssrn.com/abstract=2808088
11. Hardy, M.: Investment Guarantees: Modeling and Risk Management for Equity-linked Life Insurance, vol. 215. Wiley, Hoboken (2003)
12. Hejazi, S.A., Jackson, K.R.: A neural network approach to efficient valuation of large portfolios of variable annuities. Insur. Math. Econ. **70**, 169–181 (2016)
13. Hejazi, S.A., Jackson, K.R., Gan, G.: A spatial interpolation framework for efficient valuation of large portfolios of variable annuities. Quant. Finan. Econ. **1**(2), 125–144 (2017). https://doi.org/10.3934/QFE.2017.2.125
14. IAA: Stochastic Modeling: Theory and Reality from an Actuarial Perspective. International Actuarial Association, Ontario, Canada (2010). http://share.actuaries.org/Documentation/StochMod_2nd_Ed_print_quality.pdf
15. Isaaks, E., Srivastava, R.: An Introduction to Applied Geostatistics. Oxford University Press, Oxford (1990)
16. Kleijnen, J.P.C.: A comment on blanning's "metamodel for sensitivity analysis: the regression metamodel in simulation". Interfaces **5**(3), 21–23 (1975)
17. Tishby, N., Pereira, F.C., Bialek, W.: The information bottleneck method. Univ. Illinois **411**(29–30), 368–377 (2000)
18. Tishby, N., Zaslavsky, N.: Deep learning and the information bottleneck principle. In: 2015 IEEE Information Theory Workshop (ITW), pp. 1–5. IEEE (2015)
19. Vu, H., Nguyen, T.D., Le, T., Luo, W., Phung, D.: Robust anomaly detection in videos using multilevel representations. In: Proceedings of Thirty-third AAAI Conference on Artificial Intelligence (AAAI), Honolulu, USA (2019)
20. Xu, W., Chen, Y., Coleman, C., Coleman, T.F.: Moment matching machine learning methods for risk management of large variable annuity portfolios. J. Econ. Dyn. Control **87**, 1–20 (2018)