Model Bridging: To Interpretable Simulation Model From Neural Network

Keiichi Kisamori[1,2], **Keisuke Yamazaki**[2], **Yuto Komori**[2], **Hiroshi Tokieda**[2]

[1] Data Science Laboratories
NEC Coopration
Nakahara-ku, Kawasaki, Japan

[2] AI Research Center
National Institute of Advanced Industrial Science and Technology
Koto-ku, Tokyo, Japan

Abstract

The interpretability of machine learning, particularly for deep neural networks, is strongly required when performing decision-making in a real-world application. There are several studies that show that interpretability is obtained by replacing a non-explainable neural network with an explainable simplified surrogate model. Meanwhile, another approach to understanding the target system is simulation modeled by human knowledge with interpretable simulation parameters. Recently developed simulation learning based on approximate Bayesian computation is a method used to estimate simulation parameters as posterior distributions. However, there was no relation between the machine learning model and the simulation model. Furthermore, the computational cost of simulation learning is very expensive because of the complexity of the simulation model. To address these difficulties, we propose a "model bridging" framework to bridge machine learning models with simulation models by a series of kernel mean embeddings. The proposed framework enables us to obtain predictions and interpretable simulation parameters simultaneously without the computationally expensive calculations associated with simulations. This framework can provide insights from the simulation based model of physical sciences to the study of machine learning models.

1 Introduction

The interpretability of machine learning, especially for deep neural networks, is strongly required when decision-making is required in a real-world application. In recent years, there are many studies that have addressed the interpretability of neural networks [5, 3, 11]. One of the approaches is to replace a un-interpretable machine learning model with a simplified surrogate model. This approach is considered to be a type of model compression. For example, Hara et al. [6] introduced a method to replace a un-interpretable random forest model with a simple decision tree model, with information criterion as a model selection problem; however, there is no method for neural networks. As another example, "distillation" of a neural network model [7] is one of the representative methods for model compression to replace a complex model with a simplified model; meanwhile, there is no interpretability for a small surrogate neural network model. These methods do not provide a clear pathway toward obtaining interpretability of a neural network.

Another approach to understanding the target system is conducting a simulation that may be outside the scope of conventional machine learning. Here, we assume a simulation such as multi-agent simulation, traffic simulation, production simulation, or simulation of the dynamics of a physical system,

Second Workshop on Machine Learning and the Physical Sciences (NeurIPS 2019), Vancouver, Canada.

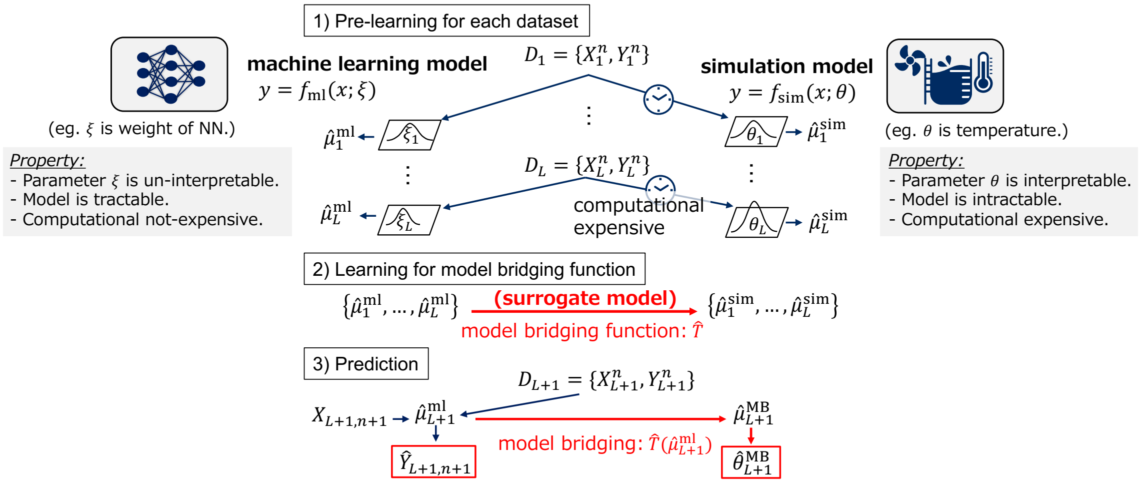
1) Pre-learning for each dataset
machine learning model
$y = f_{\mathrm{ml}}(x;\xi)$
(eg. ξ is weight of NN.)
Property:
- Parameter ξ is un-interpretable.
- Model is tractable.
- Computational not-expensive.
$D_1 = \{X_1^n, Y_1^n\}$
simulation model
$y = f_{\mathrm{sim}}(x;\theta)$
(eg. θ is temperature.)
Property:
- Parameter θ is interpretable.
- Model is intractable.
- Computational expensive.
$\hat{\mu}_1^{\mathrm{ml}}$
ξ_1
θ_1
$\hat{\mu}_1^{\mathrm{sim}}$
$D_L = \{X_L^n, Y_L^n\}$
$\hat{\mu}_L^{\mathrm{ml}}$
ξ_L
computational expensive
θ_L
$\hat{\mu}_L^{\mathrm{sim}}$
2) Learning for model bridging function
$\{\hat{\mu}_1^{\mathrm{ml}}, \dots, \hat{\mu}_L^{\mathrm{ml}}\}$
(surrogate model)
$\{\hat{\mu}_1^{\mathrm{sim}}, \dots, \hat{\mu}_L^{\mathrm{sim}}\}$
model bridging function: $\hat{T}$
3) Prediction
$D_{L+1} = \{X_{L+1}^n, Y_{L+1}^n\}$
$X_{L+1,n+1}$
$\hat{\mu}_{L+1}^{\mathrm{ml}}$
$\hat{\mu}_{L+1}^{\mathrm{MB}}$
model bridging: $\hat{T}(\hat{\mu}_{L+1}^{\mathrm{ml}})$
$\hat{Y}_{L+1,n+1}$
$\hat{\theta}_{L+1}^{\mathrm{MB}}$

Figure 1: Illustration of the algorithm of model-bridging framework.

which are widely used in social and industrial society. Simulation modeling is implemented to describe the basic law of the objective system, using human knowledge with interpretable simulation parameters. Recently developed "simulation learning" [8] is a method in which simulation parameters are estimated as posterior distributions in the context of machine learning. The simulation model is treated as an intractable or non-differentiable regression function in simulation learning. The challenge of simulation learning is a computational cost that is often more expensive than that of the machine learning model because of the complexity of the simulation model. Thus, if we have a simulation model for the objective system, we now have two ways to reproduce real data: machine learning with a statistical model and simulation learning with a simulation model. However, before, there was no way to relate a simulation model with a machine learning model, such as a neural network model.

We propose a "model bridging" framework to bridge the un-interpretable aspect of the machine learning model and the interpretable aspect of the simulation model (Fig. 1). A model-bridging framework enables us to not only predict a new dataset with high accuracy using a machine learning model but also obtain interpretable simulation parameters simultaneously without the expensive calculation of a simulation model.

2 Related Works

We briefly review a series of applications of kernel mean embedding [12] as the building blocks of the proposed framework. Kernel mean embedding is a framework to map distributions into a reproducing kernel Hilbert space (RKHS) $\mathcal{H}$ as a feature space.

Simulation Learning: "Simulation learning" [8] is a method in which the simulation model is treated as a regression function $f_{\mathrm{sim}}(x;\theta)$ by combining a series of kernel mean embedding methods. Conventional statistical methods of parameter estimation are not applicable owing to the properties of the likelihood function: intractable or nondifferentiable.

Kernel ABC: Kernel ABC [13, 4] is a method to compute the kernel mean of the posterior distribution from a sample of parameter θ, generated by the prior distribution $\pi(\theta)$. The assumption is that the explicit form of the likelihood function is intractable, while the sample from the likelihood is available. The kernel ABC allows us to calculate the kernel mean of the posterior distribution as follows: First, sample $\{\theta_1, ..., \theta_m\}$ is generated from prior distribution $\pi(\theta)$ and pseudo-data $\{\bar{Y}_1^n, ..., \bar{Y}_m^n\}$ as a sample from $p(y|x, \theta_j)$ for $j = 1, ..., m$. Next, the empirical kernel mean of the posterior distribution $\hat{\mu}_{\theta|YX} = \sum_{j=1}^{m} w_j k_\theta(\cdot, \theta_j)$ is calculated, where k_θ is a kernel of θ. Weight w_j is calculated by kernel of y [4].

Kernel Herding: Kernel herding [1] is a method used to sample data from the kernel mean representation of a distribution, which is an element of the RKHS. Kernel herding can be considered as an opposite operation to that of kernel ABC. Kernel herding greedily obtains samples $\{\theta_1, ..., \theta_m\}$ by updating Eqs.(1) and (2) in Chen et al. [1].

Distribution Regression: Distribution regression is a regression for d_x-dimensional “distributions” represented by samples. Meanwhile normal regression is regression for d_x-dimensional “point”. There are several studies of distribution regression, including distribution-to-distribution regression [15] and distribution-to-point regression [16, 10]. Oliva et al. [15] employ the idea of approximating a density function by kernel density estimation, rather than using RKHS. Szabó et al. [16] propose the distribution-to-point with kernel ridge regression method on RKHS; however, there are no methods for distribution-to-distribution regression.

3 Proposed Framework: Model Bridging

We propose a new framework to bridge the un-interpretable machine learning model and the interpretable simulation model. In this study, we assume a machine learning model, such as a Bayesian neural network (BNN) [14] with a few hidden layers. Meanwhile, this proposed framework is applicable to any model. Figure 1 shows an overview of the framework.

3.1 Problem Setting, Assumption, and Usage of Model Bridging

We define the problem setting of the model-bridging framework. Let L be dataset $\{X_1^n, Y_1^n, ..., X_L^n, Y_L^n\}$ $(X_l^n \in \mathbb{R}^{n\times d_x}, Y_l^n \in \mathbb{R}^{n\times d_y})$, given in the pre-learning phase. The purpose is to predict $\hat{Y}_{L+1,n+1}$ and simultaneously obtain interpretable simulation parameter $\hat{\theta}_{L+1}^{\mathrm{MB}}$ to reproduce $Y_{L+1,n+1} = f_{\mathrm{sim}}(X_{L+1,n+1}; \hat{\theta}_{L+1}^{\mathrm{MB}})$ without the expensive calculation of simulation model $f_{\mathrm{sim}}(x;\theta)$ when we obtain new dataset $\{X_{L+1}^n, Y_{L+1}^n\}$. The assumptions of the problem setting are as follows. These assumptions are a typical setting for use case of a simulation.

- Existing simulation model $f_{\mathrm{sim}}(x;\theta)$ with interpretable simulation parameter $\theta \in \mathbb{R}^{d_\theta}$ and a machine learning model $f_{\mathrm{ml}}(x;\xi)$ that is sufficiently accurate to predict a typical regression problem while having un-interpretable parameter $\xi \in \mathbb{R}^{d_\xi}$.
- Cost of simulation learning is much higher than that of learning from the machine learning model. For example, it takes more than 1 day for simulation learning of one dataset $\{X_l^n, Y_l^n\}$ while learning of BNN takes less than minute.
- Dataset $\{X_l^n, Y_l^n\}$ has dependency of parameter θ_l for each $l = 1, ..., L$. Let us assume the following situation: $\{X_l^n, Y_l^n\}$ is obtained in one time period with the same conditions, described as parameter θ_l, while conditions are changed for the following time period, described as θ_{l+1}.
- Time for off-line calculation of simulation learning is sufficient, while time for prediction is restricted.

Once we obtain model-bridging function $\hat{T}$ as a mapping from the machine learning model to the simulation model, we can obtain an accurate prediction for $\hat{Y}_{L+1,n+1}$ by both the machine learning model and interpretable $\hat{\theta}_{L+1}^{\mathrm{MB}}$ by the simulation model for new dataset $\{X_{L+1}^n, Y_{L+1}^n\}$ without an expensive calculation from the simulation model.

3.2 Distribution-to-Distribution Regression Based on Kernel Ridge Regression

We present the regression algorithm between the conditional kernel mean of the machine learning model $\mu^{\mathrm{ml}} \in \mathcal{H}$ and that of the simulation model $\mu^{\mathrm{sim}} \in \mathcal{H}$, as a model-bridging function $\mu_l^{\mathrm{sim}} = T(\mu_l^{\mathrm{ml}})$. We develop the algorithm based on kernel ridge regression which is suitable for kernel mean input and output on RKHS. This is the extension of the distribution-to-point regression method proposed by Szabó et al. [16] for the distribution output. The formulation to be solved is as follows as an analogy of normal kernel ridge regression:

$$\hat{T} = \arg\max_{T\in\mathcal{F}} \frac{1}{L}\sum_{l=1}^{L} \|\hat{\mu}_l^{\mathrm{sim}} - T(\hat{\mu}_l^{\mathrm{ml}})\|_{\mathcal{F}}^2 + \lambda\|T\|_{\mathcal{F}}^2, \tag{1}$$

where $\lambda > 0$ is a regularization constant. $\mathcal{F}$ is a function space of kernel mean embeddings following Christmann et al. [2] and $\|\cdot\|_{\mathcal{F}}$ is its norm. The difference from normal kernel ridge regression is

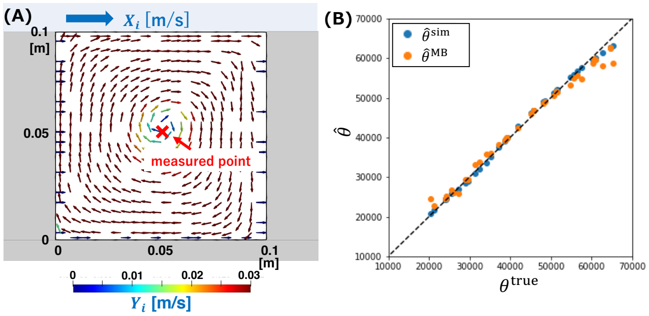

Figure 2: (A) Experiment of "cavity", which is a two-dimensional square space surrounded by walls (gray) on three sides while moving material (light blue) is located on top of the space. (B) The estimated result of Reynolds number by simulation learning ($\hat{\theta}^{\mathrm{sim}}$) and model-bridging ($\hat{\theta}^{\mathrm{MB}}$) as a function of true θ (θ^{true}).

that the inputs and outputs are kernel means. Therefore, we define kernel $\kappa \in \mathcal{F}$ as a function of kernel mean $\mu \in \mathcal{H}$. We employ a Gaussian-like kernel as $\kappa(\mu, \mu') = \exp\left\{-\frac{1}{2\sigma_\mu^2}\|\mu - \mu'\|_{\mathcal{H}}^2\right\} \in \mathcal{F}$, where constant $\sigma_\mu > 0$ is the width of kernel κ and $\|\cdot\|_{\mathcal{H}}$ is RKHS norm. The kernel κ is also a positive definite kernel [2]. Following the representer theorem of kernel ridge regression [9], the estimated model-bridging function $\hat{T}$ for new $\hat{\mu}_{L+1}^{\mathrm{ml}}$ is described as $\hat{\mu}_{L+1}^{\mathrm{MB}} = \hat{T}(\hat{\mu}_{L+1}^{\mathrm{ml}}) = \sum_{l=1}^{L} v_l \hat{\mu}_l^{\mathrm{sim}} \in \mathcal{F}$, where $\mathbf{v} = (v_1, ..., v_L)^T = (G_\mu + \lambda L I)^{-1}\mathbf{k}_\mu(\hat{\mu}_{L+1}^{\mathrm{ml}}) \in \mathbb{R}^L$. Gram matrix G_μ and the vector $\mathbf{k}_\mu(\hat{\mu}_{L+1}^{\mathrm{ml}})$ are described as the function of κ.

We assume BNN model $f_{\mathrm{ml}}(x; \xi)$ with a few hidden layers, where ξ is parameter such as weights for each node and bias terms of each layer. We can obtain the posterior distribution of ξ_l for $l = 1, ..., L$ by the Markov Chain Monte Carlo (MCMC) method or variational approximation. The $j = 1, ..., m$ is the number of parameter samples. Then, the empirical kernel mean of the posterior distribution is represented as $\hat{\mu}_l^{\mathrm{ml}} = \sum_{j=1}^{m} k_\xi(\cdot, \xi_{l,j}) \in \mathcal{H}$ for $l = 1, ..., L$ dataset where k_ξ is kernel of ξ.

After obtaining the kernel mean of $\hat{\mu}_{L+1}^{\mathrm{MB}}$, kernel herding can be applied to sample $\hat{\theta}_{L+1}^{\mathrm{MB}} = \{\hat{\theta}_{L+1,1}, ..., \hat{\theta}_{L+1,m}\}$ where $\hat{\theta}_{L+1,j} \in \mathbb{R}^m$. The explicit form of the update equation for sample $j = 1, ..., m$ iteration of kernel herding with kernel mean $\hat{\mu}_{L+1}^{\mathrm{MB}}$ is as follows:

$$\hat{\theta}_{L+1,j} = \arg\max_{\theta} \sum_{l=1}^{L} \sum_{j'=1}^{m} v_l w_{l,j'} k_\theta(\theta, \theta_{l,j'}) - \frac{1}{j} \sum_{j'=1}^{j-1} k_\theta(\theta, \theta_{j'}) \in \mathbb{R}^{d_\theta}, \tag{2}$$

for $j = 2, ..., m$. For initial state $j = 1$, the update equation is only the first term of Eq. (2). The weight of $w_{l,j}$ is calculated by kernel ABC for dataset $\{X_l^n, Y_l^n\}$.

4 Experiment

Through computer aided engineering (CAE) simulations, we confirm that our model-bridging algorithm is applicable to the simulation of fluid-dynamics systems. We employ the typical benchmark in this field, named "cavity flow experiment", which is shown in Fig. 2 (A). We consider a two-dimensional squared space called "cavity" fulfilled with fluid having unknown Reynolds number. The Reynolds number is used to help predict flow patterns and velocities in fluid dynamics. Turbulent flow, in particular, is somewhat difficult to predict, even though it is very common in real-world situations. In this experiment, input $X_i \in \mathbb{R}$ is the velocity of the material on top of the cavity; output $Y_i \in \mathbb{R}$ is velocity at the particular point (see Fig. 2 (A)); and parameter $\theta \in \mathbb{R}$ is the Reynolds number. The number of data $n = 50$; the number of samples $m = 41$; and the number of dataset $L = 41$ are generated by different true $\theta_l (= \theta_l^{\mathrm{true}})$. The hyperparameter of regularization is $\lambda = 1.0^{-5}$.

Figure 2 (B) shows the estimated result of $\hat{\theta}^{\mathrm{sim}}$ by simulation learning and $\hat{\theta}^{\mathrm{MB}}$ by model bridging as a function of true θ for $L = 41$ dataset with one-leave-out cross-validation. Dashed line shows $\theta^{\mathrm{true}} = \hat{\theta}^{\mathrm{MB}} (= \hat{\theta}^{\mathrm{sim}})$, so that estimation is accurate if the result is on the dashed line. We can see reasonable estimation of $\hat{\theta}^{\mathrm{MB}}$. Human experts can understand why such flow of fluid is caused by the Reynolds number.

5 Conclusion

We propose a novel framework named “model-bridging” to bridge from the un-interpretable machine learning model to the simulation model with interpretable parameters. The model-bridging framework enables us to not only obtain precise prediction from the machine learning model but also obtain the interpretable simulation parameter simultaneously without the expensive calculations involved in a simulation. We confirm the effectiveness of the model-bridging framework and accuracy of the estimated simulation parameter simulation of fluid-dynamics.

Acknowledgement

This work was supported by JST-Mirai Program Grant Number JPMJMI18B9, Japan.

References

[1] Yutian Chen, Max Welling, and Alex Smola. 2010. Super-samples from kernel herding. *Proceedings of the Twenty-Sixth Conference Annual Conference on Uncertainty in Artificial Intelligence* (2010), 109–116.

[2] Andreas Christmann and Ingo Steinwart. 2010. Universal Kernels on Non-Standard Input Spaces. *Advances in Neural Information Processing Systems* (2010).

[3] Finale Doshi-Velez and Been Kim. 2017. Towards A Rigorous Science of Interpretable Machine Learning. (2017). `https://doi.org/10.1016/j.intell.2013.05.008`

[4] Kenji Fukumizu, Le Song, and Arthur Gretton. 2013. Kernel Bayes' Rule: Bayesian Inference with Positive Definite Kernels. *Journal of Machine Learning Research* 14 (2013), 3753–3783.

[5] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Dino Pedreschi, and Fosca Giannotti. 2018. A Survey Of Methods For Explaining Black Box Models. (2018). `https://doi.org/10.1145/3236009`

[6] Satoshi Hara and Kohei Hayashi. 2018. Making Tree Ensembles Interpretable: A Bayesian Model Selection Approach. *Proceedings of the 21 International Conference on Artificial Intelligence and Statistics* (2018). `https://doi.org/10.1002/hbm.24063`

[7] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the Knowledge in a Neural Network. *arXiv:1503.02531v1* (2015).

[8] Keiichi Kisamori and Keisuke Yamazaki. 2018. Intractable Likelihood Regression for Covariate Shift by Kernel Mean Embedding. *arXiv:1809.08159* (2018).

[9] S. Y. Kung. 2014. *Kernel Methods and Machine Learning*. Cambridge University Press. `https://doi.org/10.1017/CBO9781139176224`

[10] Ho Chung Leon Law, Dougal J. Sutherland, Dino Sejdinovic, and Seth Flaxman. 2017. Bayesian Approaches to Distribution Regression. *Proceedings of the 21st International Conference on Artificial Intelligence and Statistics (AISTATS)* (2017).

[11] Christoph Molnar. 2012. *Interpretable Machine Learning*. Christoph Molnar.

[12] Krikamol Muandet, Kenji Fukumizu, Bharath Sriperumbudur, and Bernhard Schölkopf. 2016. Kernel Mean Embedding of Distributions: A Review and Beyonds. *arXiv:1605.09522* (2016), 133. `https://doi.org/10.1561/2200000060`

[13] Shigeki Nakagome, Kenji Fukumizu, and Shuhei Mano. 2013. Kernel approximate Bayesian computation in population genetic inferences. *Statistical Applications in Genetics and Molecular Biology* 12, 6 (2013), 667–678. `https://doi.org/10.1515/sagmb-2012-0050`

[14] Radford M. Neal. 1996. *Bayesian Learning for Neural Networks*. Springer. `https://doi.org/10.1007/978-1-4612-0745-0`

[15] Junier B Oliva and Jeff Schneider. 2013. Distribution to Distribution Regression. *Proceedings of The 30th International Conference on Machine Learning* (2013).

[16] Zoltan Szabo, Bharath Sriperumbudur, Barnabas Poczos, and Arthur Gretton. 2016. Learning Theory for Distribution Regression. *Journal of Machine Learning Research* 17 (2016), 1–40.