A New Framework of Prediction Error Decomposition for the Machine Learning Era

Kazuki Kuriyama[1], Masafumi Suzuki[2], Hirokazu Iwasawa[3]

ABSTRACT

Research on predictive modeling methods has been remarkable in recent years, and models for actuaries, including machine learning ones, are advancing day by day. However, actuaries are interested not only in prediction accuracy, but also in evaluating the prediction error and interpreting what the source of the error is. Thus, a concept used in the actuarial field is prediction error decomposition, which means prediction error is decomposed into three components: process error, parameter error, and model error. But how can the error be decomposed when a machine learning model is used?

A lot of previous research of error decomposition has been conducted for traditional actuarial models such as GLM. However, there seems to be little research of general-purpose error decomposition methodology applicable even to machine learning models.

As an outcome of the research by the Data Science Related Basic Research Working Group of the Institute of Actuaries of Japan (IAJ), this paper proposes a new framework for decomposing prediction errors into process errors, parameter errors, and other errors, which is widely applicable to varieties of predictive modeling methods.

In this framework, the basic concept of process error is reconsidered in order to accommodate error decomposition for machine learning models. We also demonstrate the usefulness of this new framework through numerical experiments. We hope the method will greatly help practicing actuaries, no matter what kind of predictive modeling method they use.

Key words

Predictive modeling, Error estimation, Error decomposition, Process error, Parameter error, Model error, Machine learning, Computational statistics, Bootstrapping, Cross validation

[1] The Institute of Actuaries of Japan
[2] Munich Re, Japan Branch
[3] Waseda University, Japan

1. Introduction

1.1 Why do we need to discuss Decomposition of Prediction Error?

Actuaries have been making efforts to decompose prediction error, which is the difference between predicted values and actual values, into process error, parameter error, and model error. For example, ASOP43 explicitly states that uncertainty is composed of process risk, parameter risk, and model risk. The definitions of each are as follows.

- 2.10 Process Risk
  The risk associated with the projection of future contingencies that are inherently variable, even when the parameters are known with certainty.
- 2.8 Parameter Risk
  The risk that the parameters used in the methods or models are not representative of future outcomes.
- 2.7 Model Risk
  The risk that the methods are not appropriate to the circumstances or the models are not representative of the specified phenomenon.

Gutterman (2017) (Chapter 17 of the IAA Risk Book) is a more detailed reference, which says that classical actuarial decomposition of risk and uncertainty is based on the following three categories: process risk (due to stochastic processes), parameter risk (if the variables chosen in a model don't have the right values) and model risk (if the model structure is appropriate and the variables in it have been selected and whose relationships are recognized properly).

The Data Science Related Basic Research Working Group of The Institute of Actuaries of Japan is studying the general framework of the prediction error decomposition. In this method, prediction error can be evaluated regardless of the type of model, and applicable not only to traditional methods but also to modern data science techniques. The WG is also working on developing a tool by which actuaries can calculate the errors easily. This paper is based on the results of the study by the WG.

We believe that it is worthwhile to construct the general framework of the prediction error decomposition and develop a general tool due to the following three reasons.

The first reason is that the framework and the tool can be used for model selection. In recent years, there has been a lot of research on predictive modeling methods, including machine learning methods. In other words, a very wide range of methods can be incorporated into practice. It is not an exaggeration to say that there are an infinite number of options for predictive modeling methods if hyperparameter selections are included. In this situation, the framework and the tool will promote the practitioners' understanding of predictive modeling methods and help them make more appropriate choices.

The second reason is that this decomposition method can be used to enhance the interpretability of the predictive model by measuring model risks. Recent research has led to the development of

various machine learning methods, including random forests, gradient boosting, and neural networks, which are used in a variety of fields. However, in machine learning methods, the priority is given to improving prediction accuracy rather than interpretability so that they are even regarded as black boxes. Therefore, a separate method to measure model risk is needed for risk management and we believe that developing the general framework of the prediction error decomposition with useful tools will meet this need. It would also be beneficial from a governance perspective, as there is a growing demand for model governance in the financial industry.

The third reason is that this is a mission of actuaries. Actuaries care not only about the predicted value but also about its uncertainty. Thus, actuaries who use modern data science techniques, often referred to as the fifth generation, also need to assess the uncertainty of the model they work with. We believe that developing the general framework of the prediction error decomposition can help accomplish this mission.

1.2 Previous research

The decomposition of prediction error into process error, parameter error, and model error is a classic idea. Cairns (2000) considers the uncertainty from three sources below and presents an inference approach based on Bayesian methods.

- uncertainty due to the stochastic nature of a given model;
- uncertainty in the values of the parameters in a given model;
- uncertainty in the model underlying what we are able to observe and determining the quantity of interest.

Cairns (2000) is also a great reference, albeit an old study, enumerating much of the previous research up to that point on risk decomposition.

Recent examples of this idea can be found in Casualty Actuarial Society (2015), McGuire et al. (2021), Taylor et al. (2016), and others. An overview follows.

Let y be the random variable to be predicted and let $\hat{y}$ be the predicted value. The prediction error $\hat{\boldsymbol{y}} - \boldsymbol{y}$ in the literature is typically decomposed as

$$\hat{y} - y = (\mathrm{E}[\hat{y}] - \mathrm{E}[y]) + (\hat{y} - \mathrm{E}[\hat{y}]) + (\mathrm{E}[y] - y). \tag{1.1}$$

Here, $(\mathrm{E}[\hat{y}] - \mathrm{E}[y])$ is called model error, $(\hat{y} - \mathrm{E}[\hat{y}])$ is called parameter error, and $(\mathrm{E}[y] - y)$ is called process error.

Then, the mean square error of prediction (MSEP) of the predicted value $\hat{y}$ is

$$\begin{aligned}\mathrm{MSEP}[\hat{y}] &= \mathrm{E}[(\hat{y} - y)^2] \\ &= \mathrm{E}\left[\left((\mathrm{E}[\hat{y}] - \mathrm{E}[y]) + (\hat{y} - \mathrm{E}[\hat{y}]) + (\mathrm{E}[y] - y)\right)^2\right].\end{aligned}$$

A series of expanding terms and subsequent simplification yields

$$\mathrm{MSEP}[\hat{y}] = \mathrm{E}[(\hat{y} - \mathrm{E}[\hat{y}])^2] + \mathrm{E}[(\mathrm{E}[y] - y)^2] - 2\mathrm{E}[y\hat{y}] + \mathrm{E}[y]^2 + \mathrm{E}[\hat{y}]^2.$$

If we assume y and $\hat{y}$ are independent, then $\mathrm{E}[y\hat{y}] = \mathrm{E}[y]\mathrm{E}[\hat{y}]$ and

$$\mathrm{MSEP}[\hat{y}] = \mathrm{E}[(\hat{y} - \mathrm{E}[\hat{y}])^2] + \mathrm{E}[(\mathrm{E}[y] - y)^2] + (\mathrm{E}[\hat{y}] - \mathrm{E}[y])^2.$$

Reordering yields

$$\mathrm{MSEP}[\hat{y}] = (\mathrm{E}[\hat{y}] - \mathrm{E}[y])^2 + \mathrm{E}[(\hat{y} - \mathrm{E}[\hat{y}])^2] + \mathrm{E}[(\mathrm{E}[y] - y)^2]. \quad (1.2)$$

The first term corresponds to the model error, the second to the parameter error, and the third to the process error.

A lot of previous research on a prediction error decomposition deal with loss reserving. For example, Taylor (2020) discusses the prediction errors of GLM used for loss reserving, and the mean square error is decomposed into three errors. McGuire et al. (2021) decomposes the prediction error in lasso-based loss reserving into three errors and then propose to calculate the sum of the model error and parameter error by using a bootstrap method, assuming unbiasedness in the model. Also, a number of other previous literature on loss reserving can be found in Hindley (2018).

Attempts to decompose prediction errors are not only in the field of loss reserving. In a previous study on mortality, Richards (2009), for example, discusses the uncertainty of the Lee-carter model's predictions in terms of model risk, parameter uncertainty, parameter stability, and stochastic variation. A recent example that addresses model errors in the machine learning era is Richman et al. (2019), whose perspective is different from our paper but noteworthy.

1.3 Outline of the general framework of the prediction error decomposition

The objective of our study is to develop a framework and tools for a general prediction error decomposition. In order to achieve the goal, we seek to develop a "framework of the prediction error decomposition for the machine learning era" that can be applied to predictive models including machine learning ones. We consider this framework to have the following features.

#1: Applicable to various predictive models, including machine learning ones

In most previous research, e.g., McGuire et al. (2021), the objective of the discussion is limited to the specific predictive model. We propose a new framework for decomposing prediction errors into process errors, parameter errors, and model errors, which is widely applicable to a variety of predictive modeling methods, including machine learning ones.

To achieve this, it is necessary to assume a prediction function whose argument is a feature vector consisting of a large number of features. For example, in previous research focusing on loss reserving, prediction error is often discussed, but feature vectors are almost never discussed. Also, when considering insurance premium rates, it is not necessary to assume a prediction function of a feature vector since only a finite number of classes are typically assumed by risk classification. On the other hand, considering machine learning methods, the output value can vary depending on the feature vector. Therefore, in formulation, it is necessary to define a prediction function that takes a feature vector as an argument.

#2: Estimating process errors without presuming the error distribution implied by the predictive model

In the previous research, the process error is determined by the predictive modeling method. In our framework, however, the basic concept of process error is reconsidered in order to accommodate error decomposition for machine learning models. The process errors measured by this proposed method do not presume the error distribution implied by the predictive model.

#3: Clarifying the independence between y and $\hat{y}$

The prediction error decomposition in the previous research is derived on the assumption of independence between y and $\hat{y}$. The formula that decomposes the mean square error into the process error, the parameter error, and the model error is also under this assumption. When considering loss reserving and insurance premium rates, such as those discussed in the previous research, it can be reasonable to formulate them as "independent" even when they are in fact “conditionally independent”. That is because, as mentioned in #1, it is assumed that y has no feature vector, or a feature vector if any can be considered fixed, rather than a covariate, as there are at most a finite number of classes.

However, under the framework for the machine learning era, it is inappropriate to formulate y and $\hat{y}$ as independent because they both depend on the common feature vector. Therefore, in order to consider the method of a prediction error decomposition that can be used for general purposes, it should be argued that y and $\hat{y}$ are explicitly conditionally independent.

#4: Model error estimation method applicable to various predictive modeling method

All three features described above relate to the formulation, but in addition, the estimation method of each error must also be considered. In most of the previous studies, model error is calculated for a single predictive model, but in our concept of predictive error decomposition, we consider estimating model errors for different predictive models. For this purpose, it is necessary to consider a model error estimation method that can be applied to various kinds of the predictive models[4].

1.4 Organization of this paper

The rest of the paper is organized as follows. In Section 2, New Framework of Prediction Error Decomposition, we propose a general concept of prediction error decomposition that satisfies the features described in Section 1.3. We also show examples of tentative estimation methods for the process error, the parameter error, and the model error. In Section 3, Development of Prediction Error Decomposition Tool, we introduce the contents of the prediction error decomposition tool prototyped based on the idea proposed in this paper. In Section 4, Next Steps, we summarize this research and present next steps.

It should be noted that the content in this paper is based on the contribution of Data Science Related Basic Research Working Group of The Institute of Actuaries of Japan.

[4] In this paper, we do not consider errors due to external factors such as "structural break in the future" mentioned by McGuire et al. (2021).

2. New Framework of Prediction Error Decomposition

2.1 Definition of terms in this paper

This section provides definitions of terms (predictive models, predictive modeling methods, various parameters) used in this paper.

2.1.1 Predictive model

In this paper, we define "a predictive model" as a model that gives the distribution of the variable y to be predicted with respect to for a feature vector $\mathbf{x}$. This distribution is assumed to be the distribution of target variable y according to the given $\mathbf{x}$, and the predicted value $\mu(\mathbf{x})$ is given based on this distribution, for example, as the expected value of this distribution.

2.1.2 Predictive modeling method

In this paper, we define "a predictive modeling method" as a procedure for uniquely determining a predictive model from given training data $\mathcal{T}_n = \{(\mathbf{x}_i, y_i); i = 1, \dots, n\}$ (n is a sample size). An algorithm that determines the parameters of the predictive model from the training data (automatically calculated by a computer as a solution to some optimization problem) is an element of a predictive modeling method. If the model has hyperparameters (described later), the algorithm to determine them (e.g., specifying a set of candidate values and, if cross-validation is used, specifying the number of divisions for that purpose) can be also included in the predictive modeling method. If it has nuisance parameters (described later), the determining procedure can also be included. However, a process that is not automatically calculated through the algorithm (such as actuarial judgment on choosing some hyperparameters) is not included in the "predictive modeling method" defined in this paper[5].

2.1.3 Parameters

In this paper, we define "a parameter" without any special modifying word as a parameter in the specifications of the predictive model, which is automatically calculated as a solution to some optimization problems defined in the predictive modeling method. For example, regression coefficients in a regression model are parameters. Parameters in this sense are referred to as “a model parameter” when it is necessary to distinguish them from other types of parameters defined below.

2.1.4 Nuisance parameter

In this paper, we define "a nuisance parameter” as a parameter of the model determined by using training data that is not used to predict the future value, for example, the variance of the normal error

[5] By defining so, when considering the tool of the prediction error decomposition in Section 3, errors related to the automatically calculated part of the model are categorized as parameter errors, and errors related to the rest of the model are categorized as model errors.

distribution in a linear regression model. It is not necessarily specified during the automatic calculation determining the model parameters so that it is to be calculated separately as needed.

2.1.5 Hyperparameter

In this paper, we define "a hyperparameter" as a value determined by the user to define the behavior of the predictive modeling method. It is determined by another procedure (using information criterion, cross-validation, etc.) than the automatic calculation determining the model parameters. For example, the regularization parameter in Lasso regression, and tuning parameters used in random forest (e.g., the number of trees, the depth, and the number of candidates of features) are hyperparameters.

2.2 Process to build the general framework of prediction error decomposition

As we mentioned in Section 1.3, we intend to provide the following features for the general framework.

#In Formulation
#1: Applicable to various predictive models, including machine learning ones
#2: Estimating process errors without presuming the error distribution implied by the predictive model
#3: Clarifying the independence between y and $\hat{y}$
#In Estimation
#4: Model error estimation method applicable to various predictive modeling method

First, in formulation, we take the following items into consideration. These will be discussed in Section 2.3.

- As for #1, we assume predictive models based on various methods, including machine learning methods. To achieve this, we formulate the prediction function as a function of a feature vector.
- As for #2, we assume process errors are not determined by the error distribution implied by the predictive model. To achieve this, we formulate it in a data-driven manner instead.
- As for #3, we assume the condition which naturally implies that y and $\hat{y}$ are independent in some sense. To achieve this, we formulate it as conditional independence under the condition of the feature vector of the object to be predicted.

Second, in estimation, we do not presume specific predictive models to estimate model errors but consider estimating model errors based on data. This will be discussed in Section 2.4.3.

2.3 General formula of prediction error decomposition

2.3.1 Definition of variables and functions

Based on the above, some definitions are given as follows.

- Feature vector: $\mathbf{x}_i = (x_{i1}, \ldots, x_{ip})$
- Target variable: $y_i = f(\mathbf{x}_i) + \varepsilon_i$
  Here, $\mathrm{E}[\varepsilon_i|\mathbf{x}_i] = 0$, $\mathrm{V}[\varepsilon_i|\mathbf{x}_i] = \sigma_\varepsilon^2(\mathbf{x}_i)$, $y_1, y_2, \ldots$ are mutually independent, and $\mathrm{E}[y_i|\mathbf{x}_i] = f(\mathbf{x}_i)$.

- Training data: $\mathcal{T}_n = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$ (Stationarity assumption)
- Feature vector of the object to be predicted: $\mathbf{x}_{\mathrm{new}}$
- Target variable of the object to be predicted: y_{new}
- Error in prediction: $\varepsilon_{\mathrm{new}} = y_{\mathrm{new}} - f(\mathbf{x}_{\mathrm{new}})$
  Here, $\varepsilon_{\mathrm{new}} | \mathbf{x}_{\mathrm{new}} \perp\!\!\!\perp \mathcal{T}_n$.
- Prediction function (returns the predicted value when a feature vector is input): $\hat{f}_n$
  Here, prediction is based on the model created by $\mathcal{T}_n$. Mathematically, $\hat{f}_n(\mathbf{x})$ is a $\sigma(\mathcal{T}_n)$-measurable random variable for fixed $\mathbf{x} \in \mathbb{R}^{\mathrm{p}}$.
- Conditional expectation of the predicted value under the condition of a feature vector $\mathbf{x}_{\mathrm{new}}$: $\mathcal{E}_n(\mathbf{x}_{\mathrm{new}}) \coloneqq \mathrm{E}\left[\hat{f}_n(\mathbf{x}_{\mathrm{new}}) \middle| \mathbf{x}_{\mathrm{new}}\right]$
  Note that it depends on the sample size n but is independent of the training data $\mathcal{T}_n$. It is the same as $\mathrm{E}\left[\hat{f}_n(\mathbf{x}_{\mathrm{new}})\right]$ if $\mathbf{x}_{\mathrm{new}}$ is fixed.

2.3.2 Derivation of general formula of prediction error decomposition

Given a feature vector of new data $\mathbf{x}_{\mathrm{new}}$, the square loss $\mathrm{Err}_n(\mathbf{x}_{\mathrm{new}})$ with respect to the target variable of new data y_{new} of the prediction target can be written as

$$\mathrm{Err}_n(\mathbf{x}_{\mathrm{new}}) \coloneqq \mathrm{E}\left[\left(y_{\mathrm{new}} - \hat{f}_n(\mathbf{x}_{\mathrm{new}})\right)^2 | \mathbf{x}_{\mathrm{new}}\right].$$

This can be decomposed as follows.

$$\mathrm{Err}_n(\mathbf{x}_{\mathrm{new}}) = \mathrm{E}\left[\left\{(y_{\mathrm{new}} - f(\mathbf{x}_{\mathrm{new}})) + (f(\mathbf{x}_{\mathrm{new}}) - \mathcal{E}_n(\mathbf{x}_{\mathrm{new}})) + (\mathcal{E}_n(\mathbf{x}_{\mathrm{new}}) - \hat{f}_n(\mathbf{x}_{\mathrm{new}}))\right\}^2 | \mathbf{x}_{\mathrm{new}}\right]$$

$$= \mathrm{E}\left[\left(y_{\mathrm{new}} - f(\mathbf{x}_{\mathrm{new}})\right)^2 | \mathbf{x}_{\mathrm{new}}\right] + \left(f(\mathbf{x}_{\mathrm{new}}) - \mathcal{E}_n(\mathbf{x}_{\mathrm{new}})\right)^2 + \mathrm{E}\left[\left(\mathcal{E}_n(\mathbf{x}_{\mathrm{new}}) - \hat{f}_n(\mathbf{x}_{\mathrm{new}})\right)^2 | \mathbf{x}_{\mathrm{new}}\right]$$

$$= \sigma_\varepsilon^2(\mathbf{x}_{\mathrm{new}}) + \left(\mathcal{E}_n(\mathbf{x}_{\mathrm{new}}) - f(\mathbf{x}_{\mathrm{new}})\right)^2 + \mathrm{V}\left[\hat{f}_n(\mathbf{x}_{\mathrm{new}}) | \mathbf{x}_{\mathrm{new}}\right].$$

The first term corresponds to the process error, the second to the model error, and the third to the parameter error. To summarize, this equation can be written with three error components as below:

$$\begin{gathered} \mathrm{Err}_n(\mathbf{x}_{\mathrm{new}}) = \mathrm{Err}_n^{\mathrm{proc}}(\mathbf{x}_{\mathrm{new}}) + \mathrm{Err}_n^{\mathrm{model}}(\mathbf{x}_{\mathrm{new}}) + \mathrm{Err}_n^{\mathrm{param}}(\mathbf{x}_{\mathrm{new}}) \\ \mathrm{Err}_n^{\mathrm{proc}}(\mathbf{x}_{\mathrm{new}}) \coloneqq \sigma_\varepsilon^2(\mathbf{x}_{\mathrm{new}}) \\ \mathrm{Err}_n^{\mathrm{model}}(\mathbf{x}_{\mathrm{new}}) \coloneqq \left(\mathcal{E}_n(\mathbf{x}_{\mathrm{new}}) - f(\mathbf{x}_{\mathrm{new}})\right)^2 \\ \mathrm{Err}_n^{\mathrm{param}}(\mathbf{x}_{\mathrm{new}}) \coloneqq \mathrm{V}\left[\hat{f}_n(\mathbf{x}_{\mathrm{new}}) | \mathbf{x}_{\mathrm{new}}\right] \end{gathered} \tag{2.2}$$

This is the formula we propose in this paper as the general formula of prediction error decomposition.

The general formula is consistent with the previous research shown above. It can be proved by considering a condition that $\hat{y}$ and y are "conditionally independent" where y is the future observation value and $\hat{y}$ is the predicted value. See Appendix for details.

2.4 Attempt of the estimation method of each error based on the general formula

Based on the general formula given above, we will consider methods for estimating the process error, the parameter error, and the model error, respectively.

2.4.1 Parameter error estimation method

The parameter error is by definition the variance of the predicted value and can be estimated using, for example, the bootstrap method (Efron (1979). See, e.g., Efron et al. (2016) for detail). At first, B bootstrap samples $\mathcal{T}_n^{*1}, \cdots, \mathcal{T}_n^{*B}$ (here, B denotes the number of bootstrap samples) are taken from the training data $\mathcal{T}_n$. Next, B prediction functions $\hat{f}_n^{*1}, \cdots, \hat{f}_n^{*B}$ are created by using respective bootstrap samples as training data. Finally, the parameter error is estimated by the following formula[6]:

$$\widehat{\mathrm{Err}_n^{\mathrm{param}}}(\mathbf{x}_{\mathrm{new}}) = \frac{1}{B-1}\sum_{b=1}^{B}\left(\hat{f}_n^{*b}(\mathbf{x}_{\mathrm{new}}) - \frac{1}{B}\sum_{b=1}^{B}\hat{f}_n^{*b}(\mathbf{x}_{\mathrm{new}})\right)^2 \tag{2.3}$$

This estimator is rather straightforward, as there can be ways to reduce the bias. However, we believe it is already useful in practice to a certain extent.

2.4.2 Process error estimation method

Estimating process error is quite challenging. According to the general formula,

$$\mathrm{Err}_n^{\mathrm{proc}}(\boldsymbol{x}_{\mathrm{new}}) = \sigma_\epsilon^2(\mathbf{x}_{\mathrm{new}}) = \mathrm{E}\left[\left(y_{new} - f(\mathbf{x}_{\mathrm{new}})\right)^2 | \mathbf{x}_{\mathrm{new}}\right]. \tag{2.4}$$

In other words, it represents "intrinsic stochastic variability that would remain even if the resulting predictive model were true". Since the true model itself cannot be identified, we have to use some estimation method to estimate process errors and, indeed, our Working Group is continuing intensive research on the method. Here, we give a very tentative estimation method for the process error which still has significant shortcomings for practical use, but is valuable in that it can be applied to any predictive model.

Here, **we assume that the predictive model created based on the training data $\mathcal{T}_n$ is "a model that can predict accurately enough" and the process error is estimated based on the actual and predicted values by the model.** Under this assumption, the process error can be estimated by substituting $\hat{f}_n(\mathbf{x}_{\mathrm{new}})$ for $f(\mathbf{x}_{\mathrm{new}})$.

We can obtain the estimated value, assuming additionally that **"the value of the process error does not change depending on the feature vector $\mathbf{x}_i$"** (equal variance assumption), by calculating (and correcting for the bias if necessary):

$$\widehat{\mathrm{Err}_n^{\mathrm{proc}}}(\mathbf{x}_{\mathrm{new}}) = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{f}_n(\mathbf{x}_i)\right)^2 =: \widehat{\sigma^2}. \tag{2.5}$$

[6] When hyperparameter tuning is also included in the predictive modeling method, an error derived from the variation of the hyperparameter value is also included in the parameter error.

Here, the right side of the formula is the average over the training data, that is, $(\mathbf{x}_i, y_i) \in \mathcal{T}_n$[7].

This estimation method is too simple and has problems that will be discussed shortly. However, the method is indeed generic, and it serves the purpose of this paper, which is to provide a framework for a generic tool. Indeed, despite its shortcomings, the estimation of process errors makes it possible to estimate model errors, as we will see later.

As mentioned above, this estimator has the following problems.

- Incompatible with the idea that process errors are inherently independent of predictive models, since estimated values of process errors can vary widely depending on predictive models.
- If the predictive model deviates significantly from the true model, the process error estimate obtained from the equation (2.5) is likely to be overestimated (especially when the process error exceeds the estimated prediction error as explained below in (iii), it is strongly suggested that the process error estimate is inadequate).
- On the other hand, if the predictive model is overfitted to the training data, the process error in the equation. (2.5) is likely to be underestimated (in extreme cases, the process error estimate is zero).
- As a matter of fact, process errors can vary widely depending on the feature vector.

Therefore, more research is needed on the method of estimating the process error. This point will be discussed again in Section 4.

2.4.3 Model error estimation method

Of course, it is impossible to directly estimate the model error because we cannot identify the true model $f(\mathbf{x})$. On the other hand, it is possible to estimate the average prediction error (expressed by AVErr_n). For example, it can be done by performing k-fold cross-validation with appropriate k. Therefore, the average model error (expressed by $\mathrm{AVErr}_{\mathrm{n}}^{\mathrm{model}}$) can be estimated as the average prediction error estimate minus the sum of the average of process error estimates and the average of parameter error estimates as follows:

$$\widehat{\mathrm{AVErr}_n^{\mathrm{model}}} = \widehat{\mathrm{AVErr}}_n - \frac{1}{n}\sum_i \widehat{\mathrm{Err}_n^{\mathrm{proc}}}(\mathbf{x}_i) - \frac{1}{n}\sum_i \widehat{\mathrm{Err}_n^{\mathrm{param}}}(\mathbf{x}_i). \tag{2.6}$$

Here, $\widehat{\mathrm{AVErr}_n^{\mathrm{model}}}$ and $\widehat{\mathrm{AVErr}}_n$ denote the estimated average model error and the estimated average prediction error, respectively.

In practice, the right-hand side of equation (2.6) may be negative due to errors in the estimated values of each term. In such cases, it is reasonable to assume that the model error is relatively small enough and estimate its value as zero.

[7] If not so, some errors other than process errors (e.g., model errors) will obviously be included.

Further, we can estimate the model error for each observation target by making adjustments for each observation target based on average model error and some "simple assumption". For example, assuming that **"model error is proportional to the sum of process error and parameter error"**, it can be estimated by introducing the adjustment factor γ as follows[8]:

$$\gamma \coloneqq \frac{n\widehat{\mathrm{AVErr}}_n}{\sum_i \widehat{\mathrm{Err}}_n^{\mathrm{proc}}(\mathbf{x}_i) + \sum_i \widehat{\mathrm{Err}}_n^{\mathrm{param}}(\mathbf{x}_i)}. \tag{2.7}$$

$$\widehat{\mathrm{Err}}_n^{\mathrm{model}}(\mathbf{x}_{\mathrm{new}}) = (\gamma - 1) \times \left(\widehat{\mathrm{Err}}_n^{\mathrm{proc}}(\mathbf{x}_{\mathrm{new}}) + \widehat{\mathrm{Err}}_n^{\mathrm{param}}(\mathbf{x}_{\mathrm{new}}) \right). \tag{2.8}$$

This estimator is also fairly simple. However, we believe it is already of some practical use.

[8] Other possible assumptions include "model error is constant regardless of the observation target" and "model error is proportional to process error". It will be needed to carefully examine what kind of estimation is better in practice by taking more theoretical consideration and numerical experiments.

3. Development of prediction error decomposition tool

3.1 Calculation by a prototype tool

In the previous section, we proposed the general formula of the prediction error decomposition that can be applied to any predictive model and provided examples of how to estimate each error. In this section, we present the computational flow of a prototype tool of prediction error decomposition for estimating each error in the manner described in Section 2.4. This tool first estimates the process error, the parameter error, and the model error for the objects in the training data. After that, the information obtained from the training data is used to estimate prediction error for the new data.

The explanation will be divided into two parts.

- Decomposition of the prediction error for training data
- Estimating the prediction error for new data

Before going into the details of the tool, we show the flow of the calculation of the tool as below.

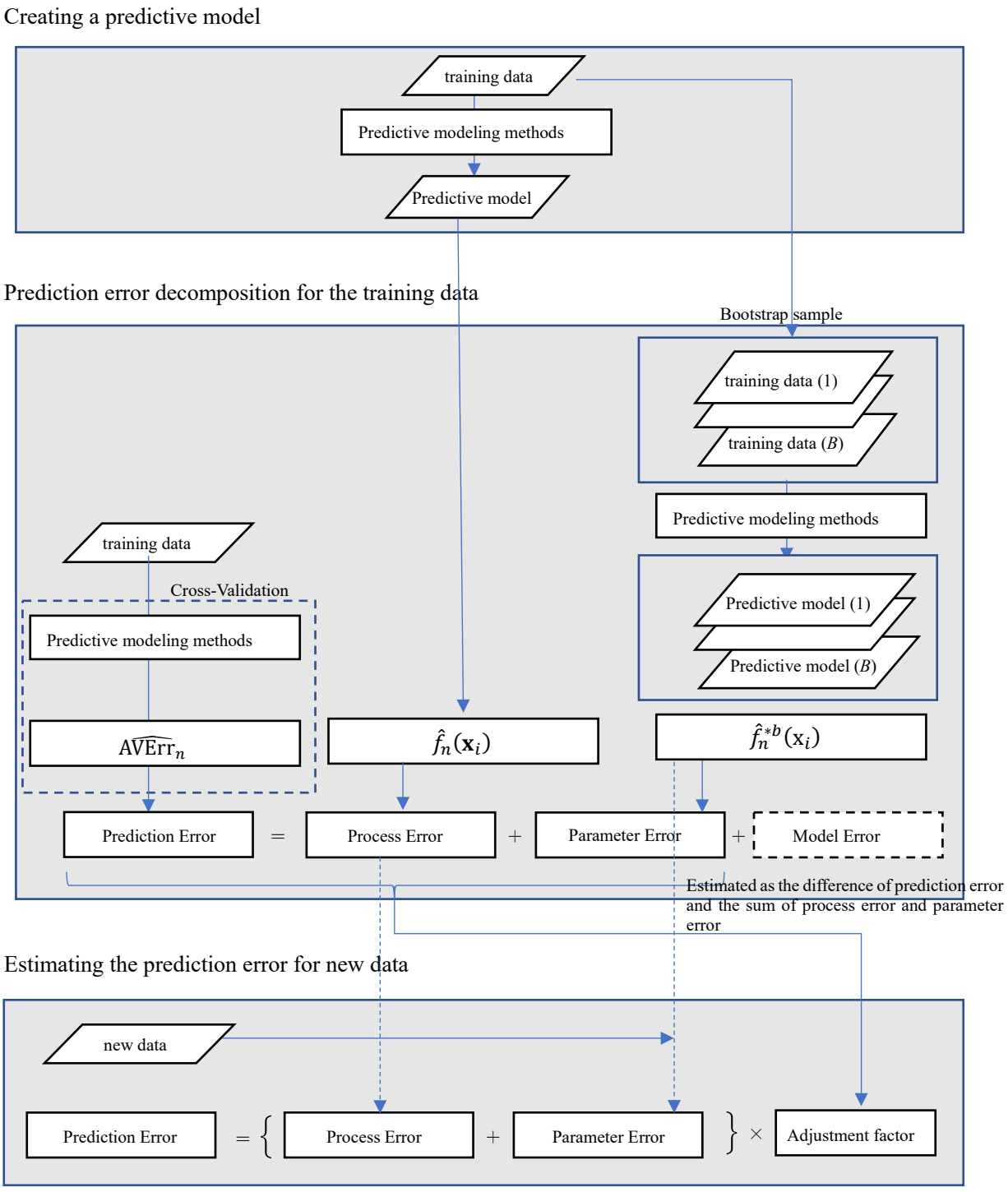

Figure 1. the computational flow of the prototype of the tool of the prediction error decomposition

3.1.1 Decomposition of prediction error for training data

3.1.1.1 Process Error

In our tentative tool, the process error is estimated based on the equation (2.4). The following is an example code when applied to a linear model.

Example: Process Error (linear model)

In the following R code, `train` is the data frame of the training data and `X` is the data frame in which each row is the feature vector of each object in the training data.

```
# Predictive model
model_LM <- lm(y ~., data = train)
pred_LM <- function(model, newx){
  predict(model, newdata = newx, type = "response")
  }

# Process Error
proc_err_LM <- mean((y - pred_LM(model_LM, X)) ^ 2)
```

3.1.1.2 Parameter Error

Our tool uses the bootstrap method to estimate the parameter error based on the equation (2.3).

Example: Parameter Error (linear model)

The following R code prepares `B` bootstrap samples, creates a linear model for each of them, and estimates the parameter errors based on them.

```
#Creating a bootstrap specimen
BS <- as.list(NULL)
for(b in 1:B){
  set.seed(b)
  BS[[b]] <- train[sample(1:n, n, replace = TRUE), ]
  }

#Create a linear model for each bootstrap sample
model_BS_LM <- as.list(NULL)
for(b in 1:B){
  model_BS_LM[[b]] <- lm(y ~., data = BS[[b]])
```

```
}

#Create a function that outputs the Parameter Error
paramErr_LM <- function(x){
  mu <- 0
  for (b in 1:B){
  mu <- mu + pred_LM(model_BS_LM[[b]], x) / B
  err <- rep(0, nrow(x))
  for(b in 1:B){
    err <- err + (pred_LM(model_BS_LM[[b]], x) - mu) ^ 2 / (B-1) }
  err
}

#Calculation of Parameter Error for training data
mean(paramErr_LM(X))
```

3.1.1.3 Model Error

In our tentative tool, the average model error is estimated based on the equation (2.6).

Example: Model Error (linear model)

The following R code estimates the model errors. At first, the average of the prediction error is estimated by k-fold Cross-Validation. Then the average model error is estimated as the average prediction error estimate minus the sum of the average of process error estimates and the average of parameter error estimates.

```
#Calculate the average Prediction Error for each model using Cross-
Validation
cv_no <- split(sample(1:n), 1:K)
pred_err_mean_LM <- 0
for(k in 1:K){
  pred_err_mean_LM <- pred_err_mean_LM +
    sum((y[cv_no[[k]]] - pred_LM(method_LM(train[-cv_no[[k]], ]),
                                 train[cv_no[[k]], ])) ^ 2) / n

#Calculate Model Error
model_err_LM <- pred_err_mean_LM - (proc_err_LM + mean(paramErr_LM(X)))
```

3.1.2 Estimating the prediction error for new data

3.1.2.1 Calculation of the adjustment factor

To estimate the prediction error for new data, we first calculate the adjustment factor based on the equation (2.7). The following R code shows how to calculate it.

```
# Calculate the adjustment factor
gamm_LM <- pred_err_mean_LM / (proc_err_LM + mean(paramErr_LM(X)))
```

3.1.2.2 Estimating the prediction error for new data

Based on the equation (2.8), the prediction error in new data is estimated by multiplying the total of the process error and the parameter error in new data by the adjustment factor. We can calculate the estimation of the process error and the parameter error in new data in the same manner as we do for the training data. As a result, the estimator of the prediction error for new data is as follows:

$$\widehat{\mathrm{Err}_n}(\mathbf{x}_{\mathrm{new}}) = \left\{ \widehat{\sigma^2} + \frac{1}{B-1} \sum_{b=1}^{B} \left(\hat{f}_n^{*b}(\mathbf{x}_{\mathrm{new}}) - \frac{1}{B} \sum_{b=1}^{B} \hat{f}_n^{*b}(\mathbf{x}_{\mathrm{new}}) \right)^2 \right\} \times \gamma. \qquad (3.1)$$

Example: Prediction Error for New Data (linear model)

The following R code gives the function `predErr` that calculates the estimated value of the prediction error.

```
#Define a function that estimates the Prediction Error using the
adjustment factor
predErr <- function(proc_err, paramErr, x, gamm){
  (proc_err + paramErr(x)) * gamm
}
#Calculate Prediction Error for new data newx
predErr(proc_err_LM, paramErr_LM, newx, gamm_LM)
```

3.2 Numerical example of the tool to housing value data

3.2.1 Prerequisites for application

In this section, we will show a numerical example of applying the tool to Boston data (housing values in suburbs of Boston) in R's MASS package. (The following content is a partial modification of the content introduced in the Actuary Journal published by the Institute of Actuaries of Japan).

The items considered for use are as follows.

- Prediction error decomposition when using the following three predictive modeling methods

- Linear model (without interaction)
- Linear model (with interaction)
- Random forest

- Out of the total 506 records of the Boston data, 10 records are used as holdout data, and the process error, the parameter error, and the model error are calculated for all records.
- The target variable *y* is `medv` (median housing value), and the feature vector `X` is configured by the feature quantities other than `medv` (however, the variables `chas` and `rad` are not used to prevent overfitting when considering the interaction term).
- If hyperparameters are used in the predictive modeling method, they are regarded as fixed values (that is, errors related to hyperparameters are not included in the parameter error).

3.2.2 Tool application results

3.2.2.1 Results for training data

First, results of application for each training data are shown in the figure below.

- The horizontal axis represents the predicted values, and the vertical axis represents the differences between the actual and predicted values.
- The squared root of the average process error is indicated by the red dotted line.
- The squared root of (the average process error + the average parameter error) is indicated by the green dotted line.
- The squared root of the average prediction error, which is interpreted as (the average process error + the average parameter error + the average model error), is indicated by the blue dotted line.

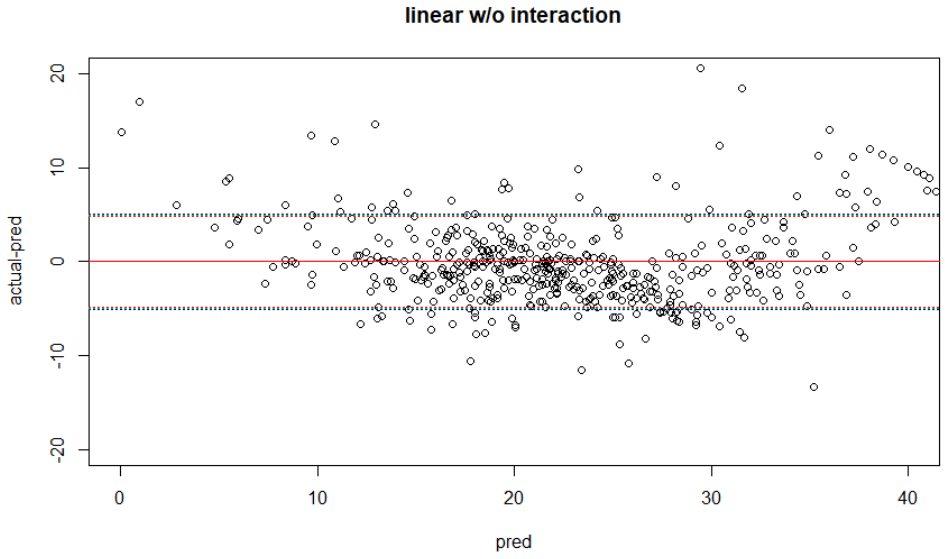

Figure 2-1: Results of application when the predictive model is a linear model without interaction

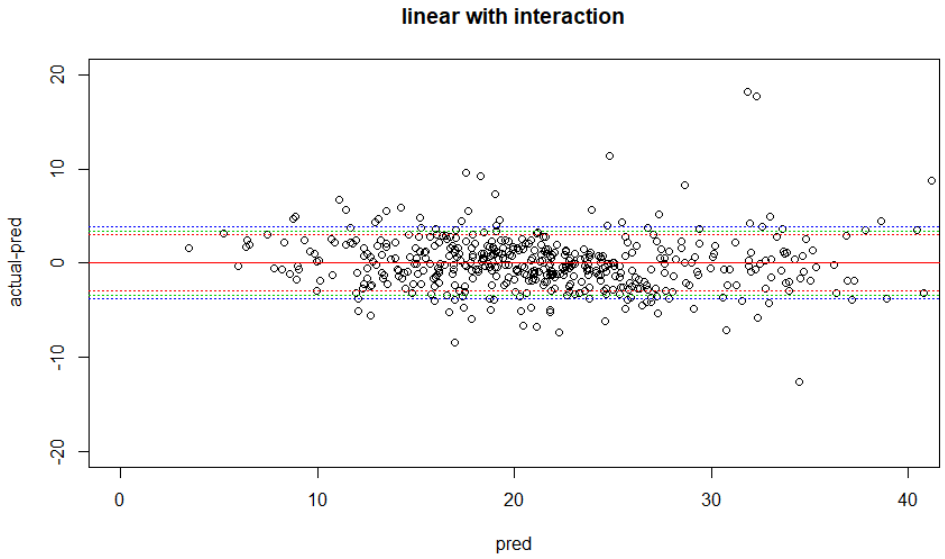

Figure 2-2: Results of application when the predictive model is a linear model with interaction

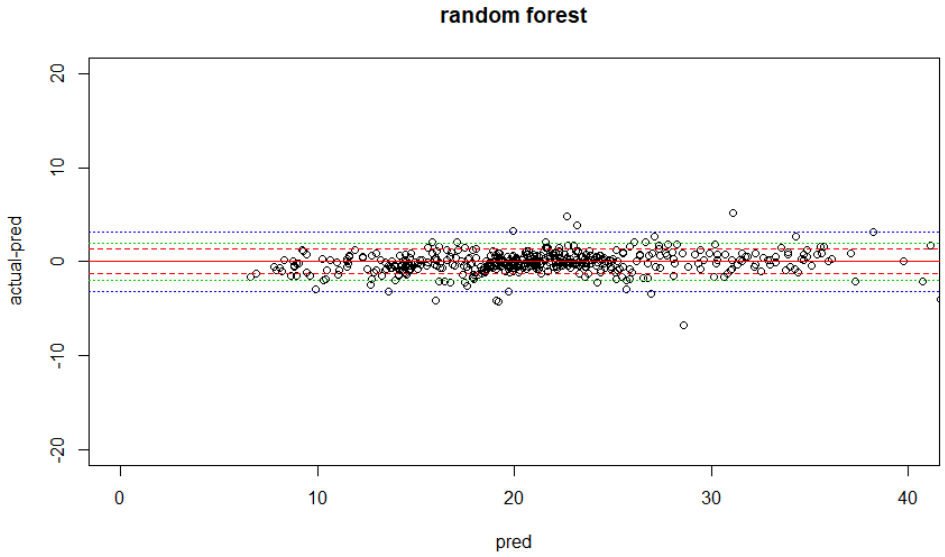

Figure 2-3: Results of application when the predictive model is a random forest

From these figures, the following can be said.

- In the case of the linear model without interaction, the average process error is large, and there are many points that are not within the range of the prediction error due to the large differences between the actual and predicted values.
- In the case of the linear model with interaction, each error is not as large as in the linear model without interaction, and the differences between the actual and predicted values are often within the range of the prediction error.

- In the case of random forest, many points are within the range of process error, and most other points are still within the range of prediction error.

The respective averages of the process error, the parameter error, and the model error for each predictive modeling method are as follows.

Table 1: The averages of each error in the training data

Predictive modeling method	Prediction Error	Breakdown		
		Process Error	Parameter Error	Model Error
Linear model without interaction	25.7811	23.58959	0.84544	1.34607
	100.0%	91.5%	3.3%	5.2%
Linear model with interaction	14.4787	8.87786	2.99085	2.61000
	100.0%	61.3%	20.7%	18.0%
Random forest	10.1715	1.74307	2.09529	6.33313
	100.0%	17.1%	20.6%	62.3%

3.2.2.2 Applying the tool of the prediction error decomposition to holdout data

The range of error for the holdout data by the tool is shown below. Red dots indicate the predicted values and black dots the actual values. Each pair of dotted lines corresponds to the process error, dashed lines to (the process error + the parameter error), and solid lines to the entire prediction error. However, each range is indicated by the square root of the original error.

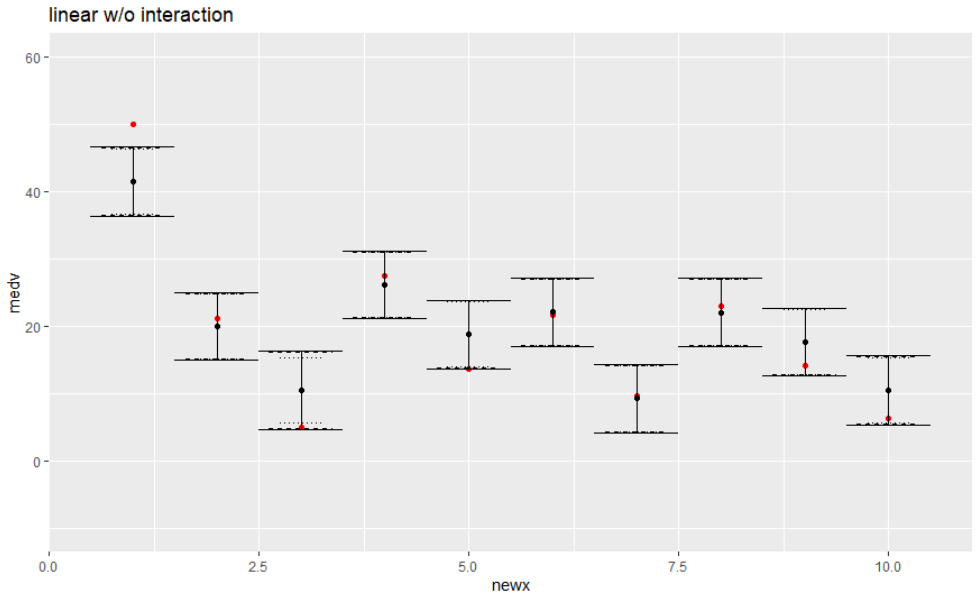

Figure 3-1: Results of error decomposition of holdout data when the predictive model is a linear model without interaction

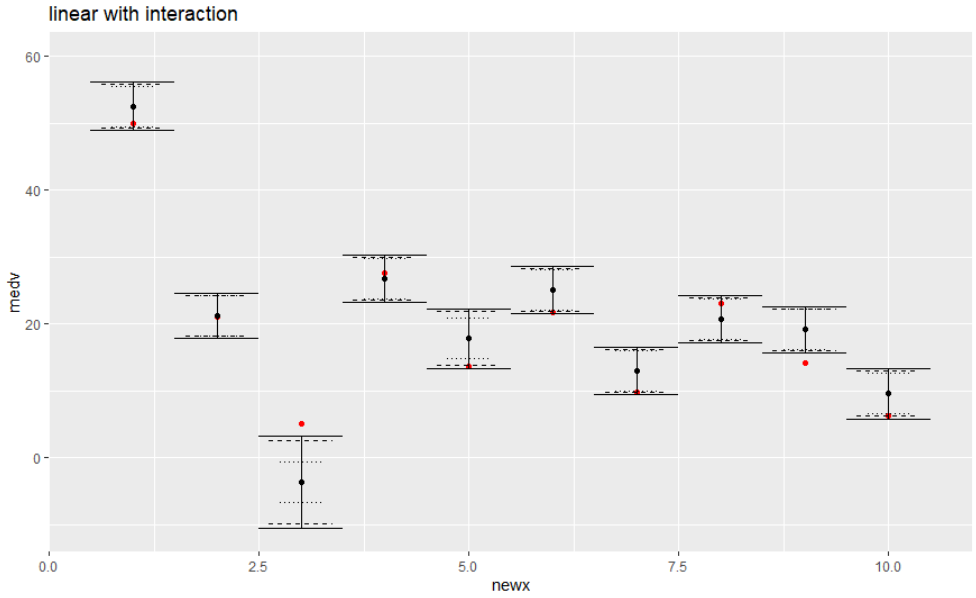

Figure 3-2: Results of error decomposition of holdout data when the predictive model is a linear model with interaction

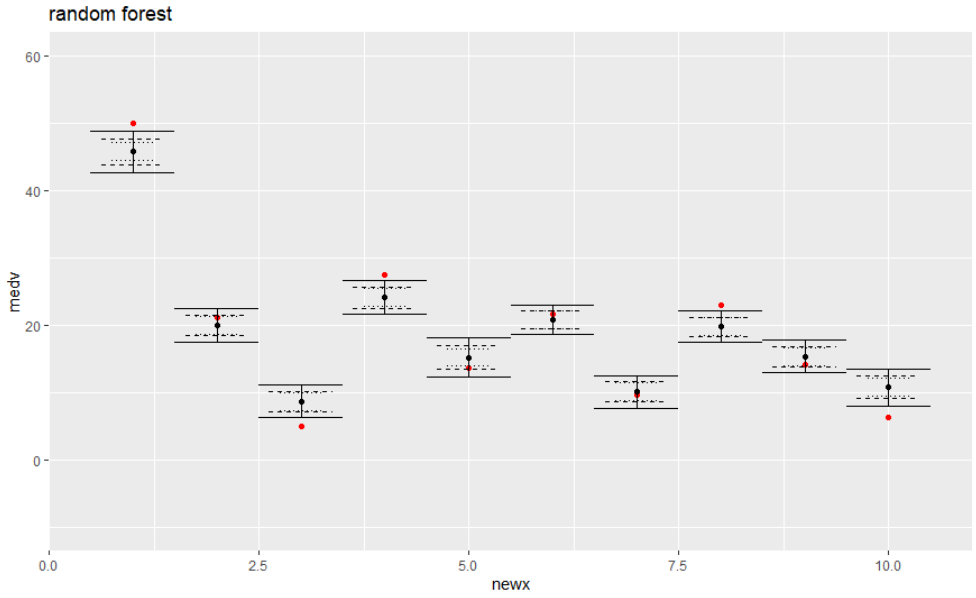

Figure 3-3: Results of error decomposition of holdout data when the predictive model is a random forest

4. Next Steps

Up to this point, we have presented a general framework of the prediction error decomposition, illustrated, albeit tentatively, how to estimate process, parameter, and model errors, respectively, and provided specific R code and numerical examples. We believe that the proposed framework is appropriate and applicable to actuarial practice. However, as discussed below, the tentative estimation methods we have presented are open to various discussion.

4.1 Enhancement of the estimation method of process errors

The estimation method of process errors is a significant issue in this study. In this paper, we set the following assumptions when estimating the process error:

- the predictive model created based on the training data $\mathcal{T}_n$ is "a model that can predict accurately enough" and the process error is estimated based on the actual and predicted values by the model, and
- the value of the process error does not change depending on the feature vector $\mathbf{x}_i$.

The process error represents "the intrinsic stochastic variability that would remain even if the resulting predictive model were true". In contrast, the first assumption relies on the idea that the process error is variability that can be estimated by assuming that the resulting predictive model is true. However, this assumption is inappropriate in that the process error is intrinsic and therefore should not differ depending on the predictive model. In addition, with the method in this paper, if the predictive model is poorly fitted, the errors caused by the inaccuracy of the model are also included in the process error. In fact, in our numerical examples, the process error values differed greatly among the predictive models. According to these results, the approach based on the assumption **that the predictive model created based on the training data $\mathcal{T}_n$ is "a model that can predict accurately enough"** is by no means satisfactory.

The second assumption was made to simplify our discussion. However, in our framework, the process error of course should depend on the feature vector. Therefore, there is room for improvement in this respect as well.

By the way, it is worth pointing out that when a linear model is adopted as the predictive model, the estimation method based on the above two assumptions is almost equivalent to the method conventionally used for linear models. It is important to note this because the fact that the tentative methods in this tool do not work well implies that the conventional methods do not work well either.

Since the current tentative assumptions are not satisfactory, our Working Group is seeking a more appropriate method to estimate the process errors for individual observation targets. Specifically, we are developing a new method for consistently estimating process errors using a random forest, regardless of the predictive model for which the prediction error is to be estimated. By estimating the process error using this method, we have obtained the results shown in the figure below.

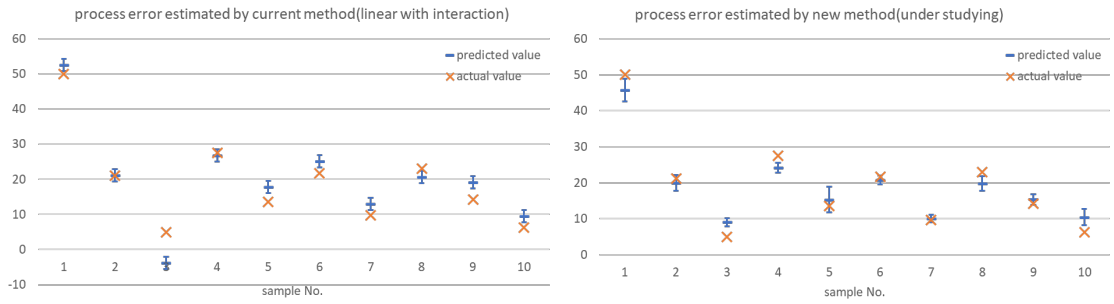

Figure 4: The result of process error estimation by the current method (left) and the new method (right). The width of the error bar shows the square root of the estimated process error.

The new method (under development) can be used to calculate the process error for each object. In the above figure 4, all the error bars for the objects have the same width in the current method (right), but in the new method, the widths for each error bar are all different (left).

Since this method is still under development and is outside the main purpose of this paper, further details are omitted here. Therefore, although it cannot be regarded as a result of this paper, we already have an idea of the approach to the estimation of process errors, and we are more confident about the feasibility of developing a more appropriate error estimation method based on the general formula of the prediction error decomposition proposed in this paper.

4.2 Examination of model error estimation method

In our tentative approach, when estimating the model error, we use an estimation method that determines the adjustment factor based on the training data $\mathcal{T}_n$, but this method has room for further study.

For example, regarding the adjustment factor, we assume that the model error is proportional to the total of the process error and the parameter error. However, other assumptions are also possible, such as "the model error is constant regardless of the observation target" or "the model error is proportional only to process error". It will be needed to carefully consider what kind of estimation is better in practice by taking theoretical consideration and numerical experiments.

4.3 Examination of other challenges for developing the general frameworks and tools of the prediction error decomposition

In the above, the issues regarding the estimation method of each error are shown, but there are various other challenges that need to be considered in advancing the efforts.

For example, the general formula and tool of the error prediction decomposition currently under consideration assume "squared loss" as the loss function. This is a measure to simplify the discussion,

but it seems that generalization of the loss function will be necessary in order to proceed with the efforts of the prediction error decomposition that can be used more universally in the future. For example, a life insurance company often deals with binary classification tasks such as whether to die or not, or whether to accept or decline an insurance application. Although we had the regression problem in mind, if we deal with the classification problems, we could take the log loss instead of the square loss assumed in Section 2.

In addition, although only very simple numerical examples were taken in this paper, needless to say, more various case studies of this tool are needed. In particular, we would like not only to decompose the prediction error but also to consider various uses of this tool.

An idea is to use this tool for risk management. For example, assuming that the only change in the number of policies is a decrease due to deaths, lapses, or surrenders, consider the situation where a model is constructed to predict the number of remaining policies of a certain life insurance company one year later from now based on past experience data. Then, of course there will be prediction errors due to uncertainty of deaths, lapses, and surrenders. In such a case, by applying the tool to decompose the prediction error, it is expected that the understanding and accountability of the predictive model will be improved, and it will be utilized for risk management.

Our working group will continue to make efforts to refine the general framework for the prediction error decomposition, including the estimation method of each error, and to enhance the usefulness and effectiveness of the tool of the prediction error decomposition, so that it will be useful for actuaries in the machine learning era.

5. Conclusion

In this paper, we discussed the general framework of the prediction error decomposition that is applicable to predictive models, including machine learning methods.

In Section 1, we explained the significance of our general prediction error decomposition. In addition, we pointed out the feature which the general framework for the machine learning era should have.

In Section 2, we proposed the general formula of the prediction error decomposition and discussed the estimation methods for each error, thereby constructed a new framework of the prediction error decomposition.

In Section 3, based on the framework discussed in Section 2, we specifically implemented the tool of the prediction error decomposition under the assumption of equal variances. Moreover, by applying the tool to the Boston data, we showed that results of the prediction error decomposition could be obtained for each predictive model.

In Section 4, we discussed several issues that should be tackled in the future. In particular, we recognized that, as pointed out in Section 2.4, the estimation method of process errors is a major issue, and our Working Group is studying a general estimation method for process errors to solve this problem. Although there are even other items to be further studied, we believe that the general prediction error framework presented in this paper is a useful method.

Through this paper, we have presented the framework of the prediction error decomposition to enable the development of a general tool applicable to various predictive models. Our working group will continue to undertake basic research for tool development and case studies.

Appendix: Comparison with previous research

According to the discussion in Taylor (2020), given the future observation value y, the prediction value $\hat{y}$, and $\mathrm{E}[y] = \mu$**,** the prediction error can be decomposed as follows:

$$\hat{y} - y = (\mathrm{E}[\hat{y}] - \mu) + (\hat{y} - \mathrm{E}[\hat{y}]) + (\mu - y).$$

Assuming that all three components are independent, the mean square error of prediction (MSEP) of the predicted value $\hat{y}$ is:

$$\mathrm{MSEP}[\hat{Z}] \coloneqq \mathrm{E}[(\hat{y} - y)^2] = \mathrm{E}[(\mathrm{E}[\hat{y}] - \mu)^2] + \mathrm{E}[(\hat{y} - \mathrm{E}[\hat{y}])^2] + \mathrm{E}[(\mu - y)^2].$$

The first term can be called model error, the second term parameter error, and the third term process error.

On the other hand, future observations y and predicted random variables $\hat{y}$ are generally considered to be highly dependent. In other words, the decomposition of the expected value $\mathrm{E}[(\hat{y} - y)^2]$ in the above equation is essentially a conditional expected value, and it is regarded that $\hat{y}$ and y are conditionally independent.

The natural requirements that imply conditional independence between $\hat{y}$ and y are:

1. the prediction target is not included in the training data, and
2. the condition is the feature vector $\mathbf{x}_{\mathrm{new}}$ of the prediction target.

Then, the source of conditional independence is that, while $\hat{y}_{\mathrm{new}}$ depends on the training data $\mathcal{T}_n$, y_{new} is independent of $\mathcal{T}_n$ under the condition of $\mathbf{x}_{\mathrm{new}}$. Considering the condition of $\mathbf{x}_{\mathrm{new}}$ and using expression $\hat{f}_n(\mathbf{x}_{\mathrm{new}})$ instead of $\hat{y}_{\mathrm{new}}$, the left side of the decomposition, $\mathrm{E}[(\hat{y} - y)^2]$, can be replaced by

$$\mathrm{Err}_n(\mathbf{x}_{\mathrm{new}}) \coloneqq \mathrm{E}\left[\left(y_{\mathrm{new}} - \hat{f}_n(\mathbf{x}_{\mathrm{new}})\right)^2 | \mathbf{x}_{\mathrm{new}}\right]$$

if the target is specified. If the target is unspecified, it will be the expected value $\mathrm{E}[\mathrm{Err}_n(\mathbf{x}_{\mathrm{new}})]]$.

Replaced as such, it is shown that the concept of the general formula proposed in this paper includes the traditional prediction error decomposition by Taylor (2020).

References

Actuarial Standards Board. (2007). Actuarial Standard of Practice No. 43. (ASOP 43)

Cairns, A. J. (2000). A discussion of parameter and model uncertainty in insurance. *Insurance: Mathematics and Economics*, *27*(3), 313-330.

Casualty Actuarial Society. (2015). Incorporating Model Error into the Actuary's Estimate of Uncertainty. *Casualty Actuarial Society E-Forum*, *Summer 2015*.

Data Science Related Basic Research Working Group of The Institute of Actuaries of Japan. (2020). Yosokumoderingu ni okeru gosa hyouka ni kansuru kenkyuu houkoku (Report on Error Evaluation in Predictive Modeling). *Akuchuarii jaanaru(Actuary Journal), 110.*

Efron, B. (1979). Bootstrap Methods: Another Look at the Jackknife. *The Annals of Statistics, 7*(1), 1-26.

Efron, B. & Hastie, T. (2016). *Computer Age Statistical Inference:Algorithms, Evidence, and Data Science*. Cambridge University Press.

Gutterman, S. (2017). IAA Risk Book Chapter 17—Risk and Uncertainty.

Hindley, D. (2018). *Claims Reserving in General Insurance*. Cambridge University Press.

McGuire, G., Taylor, G., & Miller, H. (2021). Self-Assembling Insurance Claim Models Using Regularized Regression and Machine Learning. *Variance, 14 (1)*.

O'Dowd, C., Smith, A., & Hardy, P. (2005). A framework for estimating uncertainty in insurance claims cost. In *Institute of Actuaries of Australia (Ed.), XVth General Insurance Seminar*.

Richards, S. J., & Currie, I. D. (2009). Longevity risk and annuity pricing with the Lee-Carter model. *British Actuarial Journal*, 317-365.

Richman, R., von Rummell, N., & Wuthrich, M. V. (2019). Believing the Bot— Model Risk in the Era of Deep Learning. *Available at SSRN 3444833*.

Risk Margins Task Force (2008). A Framework for Assessing Risk Margins. In *Institute of Actuaries Australia, 16th General Insurance Seminar*.

Taylor, G. (2019). Loss reserving models: Granular and machine learning forms. *Risks*, *7*(3), 82.

Taylor, G. (2020). Loss reserving prediction error with special reference to a Tweedie sub-

family. *Available at SSRN 3642378*.

Taylor, G., & McGuire, G. (2016). Stochastic Loss Reserving Using Generalized Linear Models. *CAS Monograph, 3*.