



## Cross-Validation for Correlated Data

Assaf Rabinowicz & Saharon Rosset

To cite this article: Assaf Rabinowicz & Saharon Rosset (2020): Cross-Validation for Correlated Data, Journal of the American Statistical Association, DOI: [10.1080/01621459.2020.1801451](https://doi.org/10.1080/01621459.2020.1801451)

To link to this article: <https://doi.org/10.1080/01621459.2020.1801451>



View supplementary material [↗](#)



Accepted author version posted online: 13 Aug 2020.



Submit your article to this journal [↗](#)



View related articles [↗](#)



View Crossmark data [↗](#)

# Cross-Validation for Correlated Data

Assaf Rabinowicz\*

Department of Statistics, Tel-Aviv University, Tel-Aviv, Israel, 69978

Saharon Rosset

Department of Statistics, Tel-Aviv University, Tel-Aviv, Israel, 69978

\*The authors gratefully acknowledge *Israel Science Foundation, grant 1804/16*

Corresponding author Assaf Rabinowicz assafrab@gmail.com

## Abstract

??????

K-fold cross-validation (CV) with squared error loss is widely used for evaluating predictive models, especially when **strong distributional** assumptions cannot be taken. However, CV with squared error loss is not free from distributional assumptions, in particular in cases involving non-i.i.d. data. This paper analyzes CV for correlated data. We present a criterion for suitability of standard CV in presence of correlations. When this criterion does not hold, we introduce a bias corrected cross-validation estimator which we term  $CV_c$ , that yields an unbiased estimate of prediction error in many settings where standard CV is invalid. We also demonstrate our results numerically, and find that introducing our correction substantially improves both, model evaluation and model selection in simulations and real data studies.

**Keywords:** *Prediction Error Estimation; Model Selection; Dependent Data; Linear Mixed Model; Gaussian Process Regression*

## 1 INTRODUCTION

Datasets with correlation structures are common in modern statistical applications in various fields, such as geostatistics (Goovaerts 1999), genetics (Maddison 1990) and ecology (Roberts et al. 2017). **Different modeling methods address the correlation structure differently. Some modeling methods, such as**

Gaussian process regression ([Rasmussen and Williams 2006](#), GPR) and generalized least squares ([Hansen 2007](#), GLS), utilize explicitly the correlation structure for achieving better prediction accuracy. Other predictive models, like random forest ([Breiman 2001](#), RF), gradient boosting machines ([Friedman 2002](#), GBM) and other machine learning models, do not consider explicitly the correlation structure but are still potentially able to utilize the correlation implicitly.

The analysis in this paper mainly focuses on correlation that appears due to latent objects, such as random effects and random fields as appear in generalized linear mixed models ([Verbeke 1997](#), GLMM) and generalized Gaussian process regression ([Rasmussen and Williams 2006](#), GGPR) in clustered, temporal and spatial datasets. A simple example that demonstrates the way that latent variable realizations affect the correlation structure is linear mixed models ([Verbeke 1997](#), LMM) with random intercept for clustered data:

$$y_{i,j} = \phi_{i,j}^T \boldsymbol{\beta} + b_i + \epsilon_{i,j}, i \in \{1, \dots, I\}, j \in \{1, \dots, n_i\}, \quad (1)$$

where  $y_{i,j} \in \mathbb{R}$  is the  $j^{th}$  observation for cluster  $i$ ,  $\phi_{i,j} \in \mathbb{R}^p$  are the observed fixed effects covariates,  $\boldsymbol{\beta} \in \mathbb{R}^p$  is the fixed effects coefficients vector,  $b_i \sim N(0, \sigma_b^2)$  are independent random effects and  $\epsilon_{i,j} \sim N(0, \sigma_\epsilon^2)$  are the i.i.d. errors. Since all the observations in cluster  $i$  share the same random effect realization, they are correlated. For more information see [Verbeke \(1997\)](#).

When it comes to prediction, the question of whether there is correlation between the observations from the *training set* and the *prediction set* — the sample that is used for the model's parameter estimation, and the set of points whose response is predicted based on the trained model, respectively — plays an important role. For example, in Eq. (1), if the training and prediction sets are sampled from the same clusters, then once the random effect realizations are estimated in the model training, they can be utilized for better prediction accuracy of the dependent variable in the prediction set. Under some conditions, a predictor that uses the estimated random effect realizations is the best linear unbiased

predictor (BLUP), for more information see [Harville et al. \(1976\)](#). Another scenario is when the training and prediction sets are sampled from different clusters. In this scenario, observations of the prediction set are not correlated with observations of the training set, and therefore estimating the random effect realizations of the training set cannot be utilized for achieving better prediction accuracy. Of course, there are other correlation settings, e.g., when the random effect realizations of the training and the prediction sets are not the same but correlated. In addition to the distributional settings that are covered by GLMM and GGPR, explicit and implicit utilization of the correlation between the training and prediction sets is common in other distributional settings in various applications, including applications involving spatial datasets ([Ward and Gleditsch 2018](#)), longitudinal datasets ([Hand 2017](#)) and datasets with hierarchical clustering structure ([Raudenbush and Bryk 2002](#)).

The correlation structure of the training and prediction sets, and in particular the correlation *between* the training and prediction sets, can affect the model's prediction error and therefore should be carefully addressed when estimating the prediction error. Since many model selection procedures are based on prediction error estimation, ignoring the correlation setting may also affect model selection decision.

Various prediction error measures have been used and analyzed under different correlation settings, e.g.,  $AIC$ -type ([Vaida and Blanchard 2005](#)) and  $C_p$ -type ([Hodges and Sargent 2001](#)). This paper focuses on K-fold cross-validation prediction error estimator ([Stone 1974](#), CV) which is the most widely used method for estimating prediction error ([Hastie et al. 2009](#)). We introduce a new perspective on CV and propose an applicable framework for analyzing how CV is affected by the correlation structure in various distributional settings. In addition, a bias corrected cross-validation measure,  $CV_c$ , is introduced.  $CV_c$  is suitable for many scenarios where CV is biased due to correlations.

Section 2 presents the setting of the problem, the theoretical results of the proposed approach, and comparison with other methods. Section 3 presents numerical analyses that support the theoretical results of Section 2.

## 2 CV FOR CORRELATED DATA

### 2.1 K-fold Estimator and Generalization Error

Let  $\{\mathbf{x}_i\}_{i=1}^n, \mathbf{x}_i \in \mathbb{R}^p$ , be an independent and identically distributed (i.i.d.) sample from the probability distribution function  $P_x$ . Let  $\{y_i\}_{i=1}^n, y_i \in \mathbb{R}$ , be a sample that is drawn independently from the distribution  $P_{y|\mathbf{x}=\mathbf{x}_i, \mathbf{s}=\mathbf{s}_0}$ , where  $\mathbf{s}_0 \in \mathbb{R}^q$  is the latent variable realization that induces correlation structure between  $y_i$ 's.

Denote  $T = \{y_i, \mathbf{x}_i\}_{i=1}^n$  as the training sample for fitting a predictive model. Also

denote  $X = \begin{bmatrix} -\mathbf{x}_1^t - \\ \vdots \\ -\mathbf{x}_n^t - \end{bmatrix}$  and  $\mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}$ .

One example of this setting is linear mixed models (LMM):

$$\mathbf{y} = \Phi\boldsymbol{\beta} + Z\mathbf{s} + \boldsymbol{\delta},$$

where  $\Phi \in \mathbb{R}^{n \times (p-q)}$  contains the fixed effects covariates,  $Z \in \mathbb{R}^{n \times q}$  contains the random effects covariates,  $\mathbf{s} \in \mathbb{R}^q$  is a normally distributed random effects vector with a general covariance matrix and  $\boldsymbol{\delta} \in \mathbb{R}^n$  is the normally distributed error term with a general covariance matrix. In this setting  $X = \{\Phi, Z\}$ . Eq. (1) is a special case of this setting. Extensions of this model are generalized linear mixed models (Breslow and Clayton 1993, GLMM), where  $\mathbf{y} = g^{-1}(\Phi\boldsymbol{\beta} + Z\mathbf{s} + \boldsymbol{\delta})$  and  $g$  is the link function, as well as hierarchical generalized linear models (Lee and Nelder 1996, HGLM) where  $\mathbf{s}$  and  $\boldsymbol{\delta}$  do not necessarily follow the normal distribution. Other examples of this setting that are commonly analyzed and represented using graphical probabilistic models tools are hidden Markov models and mixture models (Jordan 2004).

Once a model is fitted, it is natural to evaluate the prediction ability of the model. In CV prediction error estimator,  $T$  is randomly partitioned into  $K$  folds,

$T_1 = \{y_i, \mathbf{x}_i\}_{i=1}^{n_1}, T_2 = \{y_i, \mathbf{x}_i\}_{i=(n_1+1)}^{n_2}, \dots, T_K = \{y_i, \mathbf{x}_i\}_{i=(n_{K-1}+1)}^{n_K}$ , <sup>1</sup> where  $n_k - n_{k-1}$  is the sample size in fold  $k$  and  $n_K = n$ . For each  $k \in \{1, \dots, K\}$ , the model is trained on the entire data except the  $k^{th}$  fold, denoted by  $T_{-k} = \bigcup_{j \neq k} T_j = \{y_{-k}, X_{-k}\}$ . The prediction error of

the trained model is then measured on the holdout fold,  $T_k = \{y_k, X_k\}$ , with respect to some loss function  $L(\cdot, \cdot): \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$ . Thus, each fold does not train the model used for predicting its outcome. The CV prediction error estimator is calculated by averaging out the estimated prediction error across all the folds, i.e.,

$$CV = \frac{1}{n} \sum_{k=1}^K \sum_{i \in k^{th} \text{ fold}} L(y_i, \hat{y}(\mathbf{x}_i; T_{-k})),$$

where  $\hat{y}(\mathbf{x}_i; T_{-k})$  is the predictor of  $y_i$ , constructed by training with  $T_{-k}$  and predicting on  $\mathbf{x}_i$ . A special case of CV is leave one out CV (Stone 1974, LOO), which is defined by setting  $K = n$ , i.e., each observation defines a fold and therefore the model is trained on  $n - 1$  observations and its prediction error is evaluated on a single observation. Under some conditions, LOO is superior to other CV variants in terms of minimizing asymptotic bias and instability (for more information see [Burman \(1989\)](#); [Arlot et al. \(2010\)](#)), but can be computationally prohibitive in some settings. A comprehensive comparison of CV variants in terms of prediction error estimation and models selection can be found in [Zhang and Yang \(2015\)](#).

In case of i.i.d. sampling, CV is considered to be an estimator of the generalization error, which is defined here in a wide sense that also covers settings with correlated data:

$$generalization \text{ error} = E_{T_{te}, T_{tr}} \frac{1}{n_{te}} \sum_{i=1}^{n_{te}} L(y_{te,i}, y(\mathbf{x}_{te,i}; T_{tr})), \quad (2)$$

where

- $T_{tr} = \{\mathbf{y}_{tr}, X_{tr}\} = \{y_{tr,i}, \mathbf{x}_{tr,i}\}_{i=1}^{n_{tr}}$ 
  - $\mathbf{x}_{tr,i}$  is an i.i.d. sample from  $P_x$
  - $y_{tr,i}$  is sampled from  $P_{y|\mathbf{x}=\mathbf{x}_{tr,i}, S=S_0}$
- $T_{te} = \{\mathbf{y}_{te}, X_{te}\} = \{y_{te,i}, \mathbf{x}_{te,i}\}_{i=1}^{n_{te}}$ 
  - $\mathbf{x}_{te,i}$  is an i.i.d. sample from  $P_x$
  - $y_{te,i}$  is sampled from  $P_{y|\mathbf{x}=\mathbf{x}_{te,i}, S=S_{te}}$

The reason that CV is commonly considered to be an estimator of the generalization error is the random mechanism that is embedded in the CV procedure. In this perspective,  $T_k$  and  $T_{-k}$  are equivalent to  $T_{te}$  and  $T_{tr}$ , respectively. Then, averaging the prediction error,

$$\sum_{i \in k^{th} \text{ fold}} L(y_i, \hat{y}(\mathbf{x}_i; T_{-k})) / (n_k - n_{k-1}), \text{ over the different folds, estimates Eq. (2).}$$

It is important to emphasize that unlike  $T$ , which is the available dataset for modeling,  $\{T_{tr}, T_{te}\}$  are used in the paper for demonstrating different prediction goals, rather than actual datasets.

**Remark 2.1.** *Note, it is typically assumed that  $T_{tr}$  is distributed as  $T$  and therefore of size  $n$ . In this case, the size of  $T_{-k}$  and  $T_{tr}$  is obviously different, and additional bias in CV evaluation is introduced. This is typically ignored, especially when considering LOO and assuming training with  $n - 1$  or  $n$  observations carries little difference. In what follows we also ignore this and implicitly assume that  $T_{tr}$  is of the same size as  $T_{-k}$  for all  $k$ .*

When

$$\{\mathbf{y}_k, X_k\} \perp\!\!\!\perp \{\mathbf{y}_{-k}, X_{-k}\}, \forall k \in \{1, \dots, K\} \quad (3)$$

and

$$\{\mathbf{y}_{te}, X_{te}\} \perp\!\!\!\perp \{\mathbf{y}_{tr}, X_{tr}\},$$

it is clear that CV is an unbiased estimator of the generalization error, however a careful analysis is required for the case when the folds are dependent.

Next we investigate how deviating from the condition in Eq. (3) contributes a bias to CV with respect to the generalization error. Based on it, a bias corrected CV estimator,  $CV_c$ , will be presented.

In what follows we limit the discussion to squared error loss function. Also note, from now on LOO setting will be assumed, and consequently  $n_{te} = 1, n_{tr} = n - 1$ , however the results are valid to other CV partitioning settings. In particular, it can easily be seen that the size of the test set,  $n_{te}$ , has no bearing on generalization error definition.

## 2.2 A General Formulation of CV Bias

Let

$$w_{cv} = E_{T_{te}, T_{tr}} (\mathbf{y}_{te} - \mathbf{y}(\mathbf{x}_{te}; T_{tr}))^2 - \frac{1}{n} E_T \|\mathbf{y} - \mathbf{y}_{cv}(T)\|^2,$$

where  $\mathbf{y}_{cv}(T) = (\hat{\mathbf{y}}(\mathbf{x}_1; T_{-1}), \dots, \hat{\mathbf{y}}(\mathbf{x}_n; T_{-n}))^t$  is the CV predictor of  $\mathbf{y}$ .

An unbiased estimator of the generalization error is:

$$CV_c = CV + w_{cv}. \quad (4)$$

Before analyzing different correlation settings, we derive a more explicit expression for  $w_{cv}$ . For simplicity, subscript notations in the expectation operator are frequently omitted. Therefore, unless a specific object is specified,  $E$  averages all the random variables that it operates on.



Since

$$\begin{aligned} E(y_{te} - y(x_{te}; T_{tr}))^2 &= E(y_{te} - E y_{te})^2 + (E y_{te} - E y(x_{te}; T_{tr}))^2 \\ &+ E(E y(x_{te}; T_{tr}) - y(x_{te}; T_{tr}))^2 \\ &+ 2E(y_{te} - E y_{te})(E y(x_{te}; T_{tr}) - y(x_{te}; T_{tr})), \end{aligned}$$

$$\begin{aligned} E \|y - y_{cv}(T)\|^2 &= E \|y - E y\|^2 + \|E y - E y_{cv}(T)\|^2 + E \|E y_{cv}(T) - y_{cv}(T)\|^2 \\ &+ 2E(y - E y)'(E y_{cv}(T) - y_{cv}(T)), \end{aligned}$$

and  $y_i$  and  $y_{te}$  have the same marginal distribution, which gives

$$E \|y - E y\|^2 / n - E(y_{te} - E y_{te})^2 = 0, \text{ then:}$$

$$\begin{aligned} w_{cv} &= (E y(x_{te}; T_{tr}) - E y_{te})^2 - \frac{1}{n} \|E y_{cv}(T) - E y\|^2 \\ E(y(x_{te}; T_{tr}) - E y(x_{te}; T_{tr}))^2 &- \frac{1}{n} E \|y_{cv}(T) - E y_{cv}(T)\|^2 \quad (5) \\ -2E \text{Cov}(y(x_{te}; T_{tr}), y_{te}) &+ \frac{2}{n} E \text{tr}[\text{Cov}(y_{cv}(T), y)], \end{aligned}$$

where  $\text{tr}$  is the trace operator and  $\text{Cov}$  is the covariance operator which contains conditional expectation of the dependent variables  $y$ ,  $y_{te}$  and  $y_{tr}$  given their covariates,  $X$ ,  $x_{te}$  and  $x_{tr}$  e.g.,  $E \text{tr}[\text{Cov}(y_{cv}(T), y)] := E_X \text{tr}[\text{Cov}(y_{cv}(T), y | X)]$ .

The first two lines in Eq. (5) are the differences between the bias and the variance of  $y_{cv}(T)$  and  $y(x_{te}; T_{tr})$ , where the expectation is taken also over the covariates. The third line relates to the covariances between the response and its predictor in each scheme — CV prediction error and generalization error.

### 2.3 Criterion for CV Unbiasedness

Let  $P_{T_{te}, T_{tr}}$  and  $P_{T_k, T_{-k}}$  be the joint distributions of  $\{T_{te}, T_{tr}\}$  and  $\{T_k, T_{-k}\}$ , respectively. Theorem 2.1 describes a simple generic condition when no correction is required for CV.

**Theorem 2.1.** *If  $P_{T_{te}, T_{tr}} = P_{T_k, T_{-k}} \forall k \in \{1, \dots, n\}$ , then  $w_{cv} = 0$ .*

Proof. Since  $\{T_k, T_{-k}\}$  were drawn from the same distribution as  $\{T_{te}, T_{tr}\}$ , then an expectation over any transformation of them is equal, in particular:

$$(\mathbb{E} y_{te} - \mathbb{E} y(\mathbf{x}_{te}; T_{tr}))^2 = \frac{1}{n} \|\mathbb{E} \mathbf{y} - \mathbb{E} \mathbf{y}_{cv}(T)\|^2.$$

Similarly

$$\begin{aligned} \mathbb{E}(y(\mathbf{x}_{te}; T_{tr}) - \mathbb{E} y(\mathbf{x}_{te}; T_{tr}))^2 &= \frac{1}{n} \mathbb{E} \|\mathbf{y}_{cv}(T) - \mathbb{E} \mathbf{y}_{cv}(T)\|^2 \\ 2\mathbb{E} \text{Cov}(y(\mathbf{x}_{te}; T_{tr}), y_{te}) &= \frac{2}{n} \mathbb{E} \text{tr}[\text{Cov}(\mathbf{y}_{cv}(T), \mathbf{y})]. \end{aligned}$$

□

Theorem 2.1 states a very basic and intuitive condition of CV unbiasedness —  
 When the CV partitioning preserves the distributional relation between the  
 prediction set to the training set, then CV is unbiased.

Let  $P_{y_k | \mathbf{x}_k, T_{-k}}$ ,  $P_{y_{te} | \mathbf{x}_{te}, T_{tr}}$  and  $P_{y_{te} | \mathbf{x}_{te}, T_{-k}}$  denote the respective conditional distribution.

The condition in Theorem 2.1 can be compacted to the following one:

$$P_{y_k | \mathbf{x}_k, T_{-k}} = P_{y_{te} | \mathbf{x}_{te}, T_{tr}}.$$

Moreover, since  $T_{tr}$  is assumed to be distributed as  $T_{-k}$ , then this condition can be rewritten as follows:

$$P_{y_k | \mathbf{x}_k, T_{-k}} = P_{y_{te} | \mathbf{x}_{te}, T_{-k}}. \quad (6)$$

Of course, when  $w_{cv} = 0$  the CV is unbiased and therefore suitable.

There is a confusion in the literature regarding the condition for CV unbiasedness. Commonly, i.i.d. or exchangeability of  $\{y_k, \mathbf{x}_k\}_{k=1}^n$  are specified as the necessary conditions for CV unbiasedness (Roberts et al. 2017; Anderson

et al. 2018). However, as can be seen in Theorem 2.1, the biasedness of CV relates to the question whether  $T_{-k}$  contributes more information for predicting  $y_k$  than  $T_{tr}$  contributes for predicting  $y_{te}$ .

We can demonstrate the use of Theorem 2.1 for a simple application — using LMM for predicting new observations from the *same* clusters that appear in the training set. In the case  $s_{te} = s_0$ , i.e.,

$$\begin{aligned} y &= \Phi\beta + Zs_0 + \dot{\mathbf{o}} \\ y_{tr} &= \Phi_{tr}\beta + Z_{tr}s_0 + \dot{\mathbf{o}}_{tr} \\ y_{te} &= \phi_{te}\beta + z_{te}s_0 + \dot{\mathbf{o}}_{te}, \end{aligned} \quad (7)$$

and

$$\begin{aligned} X &= \{\Phi, Z\}, \Phi \in \mathbb{R}^{n \times (p-q)}, Z \in \mathbb{R}^{n \times q} \\ X_{tr} &= \{\Phi_{tr}, Z_{tr}\}, \Phi_{tr} \in \mathbb{R}^{(n-1) \times (p-q)}, Z_{tr} \in \mathbb{R}^{(n-1) \times q} \\ \mathbf{x}_{te} &= \{\phi_{te}, z_{te}\}, \phi_{te} \in \mathbb{R}^{1 \times (p-q)}, z_{te} \in \mathbb{R}^{1 \times q}, \end{aligned}$$

$s_0 \in \mathbb{R}^q$  is the random effect realization vector, where each entry is a random effect realization for a different cluster and  $\dot{\mathbf{o}} \in \mathbb{R}^n, \dot{\mathbf{o}}_{tr} \in \mathbb{R}^{(n-1)}, \dot{\mathbf{o}}_{te} \in \mathbb{R}$  are i.i.d. normal errors terms.

As was mentioned previously, the observations in  $X, X_{tr}$  and  $\mathbf{x}_{te}$  are i.i.d.. Also, in this example  $y, y_{tr}$  and  $y_{te}$  were drawn given the same latent variable realization,  $s_0$ , therefore Theorem 2.1's condition —  $P_{y_k | \mathbf{x}_k, T_{-k}} = P_{y_{te} | \mathbf{x}_{te}, T_{-k}}$  — is satisfied and  $w_{cv} = 0$ . This use case of predicting new points from the same clusters that were used in the training data is common, for example see [Gelman \(2006\)](#).

The principle of CV suitability for the setting in Eq. (7) is discussed in the LMM literature ([Fang 2011](#); [Little et al. 2017](#)), however we did not find any general mathematical formalization of it. Commonly, CV is avoided in applications involving correlated data based on the wrong perception that CV is always

unsuitable for these cases (Anderson et al. 2018; Roberts et al. 2017). It is also important to stress that since the condition in Theorem 2.1 only relates to the distributional relation between  $\{T_{te}, T_{tr}\}$  and  $\{T_k, T_{-k}\}$  rather than specifying a distribution, Theorem 2.1 can be implemented in applications where the distributional settings are not fully specified, as is common when implementing machine learning algorithms.

## 2.4 CV Correction

Now, consider the setting where Theorem 2.1's condition is not satisfied, i.e., when  $P_{T_{te}, T_{tr}} \neq P_{T_k, T_{-k}}$ . A simple scenario for this setting can again be taken from LMM for clustered data, where  $T_{te}$  contains a *new* latent random effects realization i.e.,

$$\begin{aligned} \mathbf{y} &= \Phi\boldsymbol{\beta} + \mathbf{Z}\mathbf{s}_0 + \dot{\mathbf{o}} \\ \mathbf{y}_{tr} &= \Phi_{tr}\boldsymbol{\beta} + \mathbf{Z}_{tr}\mathbf{s}_0 + \dot{\mathbf{o}}_{tr} \\ \mathbf{y}_{te} &= \phi_{te}\boldsymbol{\beta} + \mathbf{z}_{te}\mathbf{s}_{te} + \dot{\mathbf{o}}_{te}, \end{aligned} \quad (8)$$

where  $\mathbf{s}_{te} \neq \mathbf{s}_0$ . Other relevant components are defined in the same way as in Eq. (7). In this scenario the correlation between  $\mathbf{y}_{tr}$  and  $\mathbf{y}_{te}$  is different than the correlation between  $\mathbf{y}_{-k}$  and  $\mathbf{y}_k$ , and a further analysis of  $w_{cv}$  is required. This can occur for example when the random effects are intercepts for clusters (e.g., cities), and the data was collected at one point of time, but the prediction task is performed on the same clusters at another point of time.

Here, first we will find an estimator of  $w_{cv}$  when  $\mathbf{y}_{cv}(T)$  is linear in  $\mathbf{y}$ , then  $w_{cv}$  for nonlinear predictors will be discussed as well. However, before that, let us demonstrate how  $\mathbf{y}_{cv}(T)$  can be formalized linearly in  $\mathbf{y}$  for some models.

**Definition 2.1.**  $\mathbf{y}_{cv}(T)$  is linear in  $\mathbf{y}$  if:

$$\mathbf{y}_{cv}(T) = H_{cv}\mathbf{y}, \quad (9)$$

where  $H_{cv} \in \mathbb{R}^{n \times n}$  does not contain  $\mathbf{y}$  and is constructed as follows:

$$H_{cv} = \begin{bmatrix} 0 & h_{1,2} & \dots & h_{1,n} \\ h_{2,1} & 0 & & h_{2,n} \\ \dots & & & \\ h_{n,1} & h_{n,2} & \dots & 0 \end{bmatrix},$$

$$h_{k,k'} \in \mathbb{R} \quad \forall k, k' \in \{1, \dots, n\}.$$

The cross validation's principle that  $y_k$  is not involved in predicting itself is reflected by having zeros in the diagonal, therefore  $\hat{y}(\mathbf{x}_k, T_{-k}) = \mathbf{h}_k \mathbf{y}_{-k}$ , where  $\mathbf{h}_k \in \mathbb{R}^{1 \times n-1}$  is the  $k^{th}$  row of  $H_{cv}$  without the diagonal element, e.g.,

$$\mathbf{h}_1 = [h_{1,2}, \dots, h_{1,n}].$$

Of course,  $H_{cv}$  can be defined for other K-fold CV settings by adjusting the dimension of the blocks with the zeros on the diagonal (in LOO the dimension of each block is  $1 \times 1$ , however for general K-fold CV is  $n/K \times n/K$ ).

Examples for linear models are ordinary least squares (OLS), generalized least squares (GLS), ridge regression, smoothing splines, LMM, Gaussian process regression (GPR) and kernel regression.

**Definition 2.2.** Let  $\mathbf{h}_{te} \in \mathbb{R}^{1 \times n-1}$  be the hat vector of  $y_{te}$ , constructed from  $\{X_{tr}, \text{Cov}(y_{te}, \mathbf{y}_{tr})\}$  in the same way as  $\mathbf{h}_k$  is constructed from  $\{X_{-k}, \text{Cov}(y_k, \mathbf{y}_{-k})\}$ .

By definition 2.2,  $\mathbf{h}_{te}$  is the equivalent of  $\mathbf{h}_k$  for the set  $\{T_{te}, T_{tr}\}$  and they are distributed the same.

**Theorem 2.2.** Let  $\mathbf{y}_{cv}(T) = H_{cv} \mathbf{y}$  be a linear predictor of  $\mathbf{y}$ , and  $y_{te}(\mathbf{x}_{te}, T_{tr}) = \mathbf{h}_{te} \mathbf{y}_{tr}$  is its corresponding predictor of  $y_{te}$ . Then:

$$w_{cv} = \frac{2}{n} \mathbb{E}[\text{tr}(H_{cv} \text{Cov}(\mathbf{y}, \mathbf{y})) - n \mathbf{h}_{te} \text{Cov}(\mathbf{y}_{tr}, y_{te})].$$

Proof. Based on Eq. (5) and by assuming linear predictor,

$$-2\mathbb{E}\text{Cov}(\hat{y}(\mathbf{x}_{te}; T_{tr}), y_{te}) + \frac{2}{n} \mathbb{E} \text{tr}[\text{Cov}(\mathbf{y}_{cv}(T), \mathbf{y})] = w_{cv}.$$

Therefore, it is only left to show that

$$(\mathbb{E} y_{te} - \mathbb{E} \mathbf{h}_{te} \mathbf{y}_{tr})^2 = \frac{1}{n} \|\mathbb{E} \mathbf{y} - \mathbb{E} H_{cv} \mathbf{y}\|^2 \quad (10)$$

$$\mathbb{E}(\mathbf{h}_{te} \mathbf{y}_{tr} - \mathbb{E} \mathbf{h}_{te} \mathbf{y}_{tr})^2 = \frac{1}{n} \mathbb{E} \|H_{cv} \mathbf{y} - \mathbb{E} H_{cv} \mathbf{y}\|^2. \quad (11)$$

Since, by definition  $y_k$  and  $y_{te}$  were drawn from the same distribution, and  $y_{-k}$  and  $y_{tr}$  were drawn from the same distribution, then although  $P_{T_{te}, T_{tr}} \neq P_{T_k, T_{-k}}$ , still

$$\|\mathbb{E} \mathbf{y} - \mathbb{E} H_{cv} \mathbf{y}\|^2 = n(\mathbb{E} y_k - \mathbb{E} \mathbf{h}_k \mathbf{y}_{-k})^2 = n(\mathbb{E} y_{te} - \mathbb{E} \mathbf{h}_{te} \mathbf{y}_{tr})^2.$$

Therefore Eq. (10) holds. Similarly with Eq. (11).  $\square$

Using Theorem 2.2, given  $\text{Cov}(\mathbf{y}, \mathbf{y})$  and  $\text{Cov}(\mathbf{y}_{tr}, y_{te})$ , an estimator of the generalization error for a linear predictor is:

$$CV_c = \frac{1}{n} (\mathbf{y} - H_{cv} \mathbf{y})' (\mathbf{y} - H_{cv} \mathbf{y}) + \frac{2}{n} [\text{tr}(H_{cv} \text{Cov}(\mathbf{y}, \mathbf{y})) - n \mathbf{h}_{te} \text{Cov}(\mathbf{y}_{tr}, y_{te})]. \quad (12)$$

A special case is when  $\text{Cov}(\mathbf{y}_{tr}, y_{te}) = 0$ . For example, in the clustered LMM setting that was given in Eq. (8),  $\text{Cov}(\mathbf{y}_{tr}, y_{te}) = 0$  when the latent variable realizations of  $\mathbf{y}_{tr}$  and  $y_{te}$ , i.e.,  $s_0$  and  $s_{te}$ , are independent. In this case:

$$CV_c = \frac{1}{n} (\mathbf{y} - H_{cv} \mathbf{y})' (\mathbf{y} - H_{cv} \mathbf{y}) + \frac{2}{n} \text{tr}[H_{cv} \text{Cov}(\mathbf{y}, \mathbf{y})]. \quad (13)$$

In case  $\mathbf{y}_{cv}(T)$  is not a linear predictor and Theorem 2.1's condition does not hold, then there is no closed form for  $w_{cv}$ . However, a correction is required as CV is not an unbiased estimator of the prediction error in this case. See Section 2.6 for some ad-hoc solutions that may apply in specific cases.

It is important to note that:

- Theorem 2.2, as well as Eqs. (12) and (13), are valid for any K-fold setting and not only for LOO. Of course,  $H_{cv}$  is based on  $K$  and should be adjusted correspondingly.
- The correction term in  $CV_c$  depends on covariance matrices, which are commonly estimated. The effect of using estimated covariance matrices instead of the true covariance matrices is discussed and analyzed in Sections 2.7 and 3.

In case there are several alternative models, it is common to use their estimated prediction errors for selecting the best model.

**Definition 2.3.** *Given a set of models  $H$ , the best  $CV_c$  model is:*

$$\operatorname{argmin}_{h \in H} CV_c(h),$$

where  $CV_c(h)$  is  $CV_c$  for model  $h$ .

#### 2.4.1 Specifying $h_{te}$ and $\operatorname{Cov}(y_{tr}, y_{te})$

The term  $n h_{te} \operatorname{Cov}(y_{tr}, y_{te})$  in Eq. (12) is relevant when  $y_{tr}$  and  $y_{te}$  are correlated (but not in the same way as  $y_k$  and  $y_{-k}$ ) e.g., assuming  $T = \{X, y\}$  has a hierarchical clustering structure:

$$\begin{aligned} y_{i,j,r} &= \mathbf{x}_{i,j,r}^t \boldsymbol{\beta} + u_i + b_{i,j} + \dot{\alpha}_{i,j,r}, \\ u_i &\sim N(0, \sigma_u^2), b_{i,j} \sim N(0, \sigma_b^2), \dot{\alpha}_{i,j,r} \sim N(0, \sigma_{\dot{\alpha}}^2), \\ \forall i &\in \{1, \dots, I\}, j \in \{1, \dots, J\}, r \in \{1, \dots, R\}. \end{aligned}$$

Assume that the prediction goal is to estimate:

$$y_{te} = \mathbf{x}_{te}^t \boldsymbol{\beta} + u_{te} + b_{te} + \dot{\alpha}_{te},$$

where  $u_{te} \in [u_1, \dots, u_I]$ , however  $b_{te}$  does not depend on the realizations of  $\{b_{i,j}\}_{i \in \{1, \dots, I\}, j \in \{1, \dots, J\}}$ . In words, the prediction goal is to predict a new observation that

relates to one of the high-level clusters (indexed by  $l$ ), but from a new low-level cluster (indexed by  $j$ ).

In this setting, since the realization of  $u_{te}$  appears in  $y$ , then  $\text{Cov}(y_{tr}, y_{te}) \neq 0$  and  $n\mathbf{h}_{te}\text{Cov}(y_{tr}, y_{te})$  is required for calculating  $CV_c$ .

Although  $T_{te}$  is an abstract object,  $\mathbf{h}_{te}$  and  $\text{Cov}(y_{tr}, y_{te})$  can be extracted from the data. Since  $\mathbf{h}_{te}$  is distributed as  $\mathbf{h}_k$  (for random  $k$ ), then  $\mathbf{h}_{te}$  can be extracted by selecting a random  $\mathbf{h}_k$  from  $H_{cv}$ .  $\text{Cov}(y_{tr}, y_{te})$  is extracted correspondingly as  $\text{Cov}(y_{-k}, y_k | \mathbf{b}, \mathbf{\hat{o}})$ , where  $\mathbf{b}$  and  $\mathbf{\hat{o}}$  are vectors containing  $\{b_{i,j}\}_{i \in \{1, \dots, I\}, j \in \{1, \dots, J\}}$  and  $\{\hat{o}_{i,j,r}\}_{i \in \{1, \dots, I\}, j \in \{1, \dots, J\}, r \in \{1, \dots, R\}}$ , respectively. By conditioning on these vectors, the remaining covariance stems from  $\mathbf{u} = [u_1, \dots, u_I]$  only, corresponding to the covariance between  $y_{tr}$  and  $y_{te}$ .

Simpler,  $n\mathbf{h}_{te}\text{Cov}(y_{tr}, y_{te})$  can be replaced by  $\text{tr}(H_{cv}\text{Cov}(y, y | \mathbf{b}, \mathbf{\hat{o}}))$ , such that:

$$\begin{aligned} CV_c &= \frac{1}{n}(\mathbf{y} - H_{cv}\mathbf{y})^t(\mathbf{y} - H_{cv}\mathbf{y}) + \frac{2}{n}[\text{tr}(H_{cv}\text{Cov}(y, y)) - \text{tr}(H_{cv}\text{Cov}(y, y | \mathbf{b}, \mathbf{\hat{o}}))] \\ &= \frac{1}{n}(\mathbf{y} - H_{cv}\mathbf{y})^t(\mathbf{y} - H_{cv}\mathbf{y}) + \frac{2}{n}\text{tr}(H_{cv}\text{Cov}(y, y | \mathbf{u})). \end{aligned}$$

### 2.4.2 Interpretation of the Results

The correction  $2/n \times [\text{tr}(H_{cv}\text{Cov}(y, y)) - n\mathbf{h}_{te}\text{Cov}(y_{tr}, y_{te})]$  is intuitive since it expresses the difference between the correlation structure of the target prediction problem and the correlation structure in the available dataset,  $T$ .

Theorems 2.1 and 2.2 emphasize that the question whether CV is biased relates to the distributional setting and it is indifferent to the implemented algorithm. The implemented algorithm is expressed only by the values of  $H_{cv}$  in  $CV_c$ .

An interesting example that stresses this understanding is when generalized least squares (GLS) is implemented in a use case with a correlation setting of  $\text{Cov}(y_{tr}, y_{te}) = 0$ . Since GLS estimates  $E(y | X)$  and does not utilize explicitly the



random effects for achieving better prediction accuracy — as LMM does by estimating  $E(y | X, s)$  — then one may think that CV is unbiased when GLS is implemented, regardless of the correlation structure of  $\text{Cov}(y_{tr}, y_{te})$ . However, as was mentioned, this is wrong since the CV biasedness relates to the distributional setting rather than to the implemented algorithm. The bias in this case is:

$$w_{cv}(GLS) = E \sum_{k=1}^n \mathbf{x}_k^t (X_{-k}^t \text{Cov}(y_{-k}, y_{-k})^{-1} X_{-k})^{-1} X_{-k}^t \text{Cov}(y_{-k}, y_{-k})^{-1} \text{Cov}(y_{-k}, y_k).$$

If  $\text{Cov}(y, y) = \sigma_0^2 I + \rho 1_{n \times n}$ , where  $1_{n \times n}$  is a  $n$  by  $n$  matrix of ones and  $\rho \in \mathbb{R}^+$ , then:

$$w_{cv} = \frac{\rho}{\sigma_0^2 + \rho(n-1)} \sum_{k=1}^n E \mathbf{x}_k^t (X_{-k}^t \text{Cov}(y_{-k}, y_{-k})^{-1} X_{-k})^{-1} X_{-k}^t 1_{n-1},$$

where the identity  $\text{Cov}(y_{-k}, y_{-k})^{-1} \text{Cov}(y_{-k}, y_k) = 1_{n-1} \rho / (\sigma_0^2 + \rho(n-1))$  is based on the result by Miller (1981).

The intuition behind the biasedness of CV in this setting is that although GLS does not explicitly utilize estimated random effect realizations, its estimated model coefficients are still affected by the random effect realizations in  $T$ . Similarly, if GLS was replaced by OLS, CV would still be biased under this correlation setting.

It is also important to emphasize that Theorem 2.1 derives  $w_{cv}$  explicitly ( $w_{cv} = 0$ ) for any model under its assumed conditions, however  $CV_c$  uses an approximated  $w_{cv}$ , which is only relevant for linear models.

### 2.4.3 Comparison with Expected Optimism

Below, a comparison between the correction in  $CV_c$  and the expected optimism correction (Efron 1986) is presented.

Expected optimism correction was developed in a context of in-sample prediction error measure:

$$\text{in-sample error} = \mathbb{E}_{\mathbf{y}^*, \mathbf{y}} \frac{1}{n} \|\mathbf{y}^* - \mathbf{y}(X; T)\|^2,$$

where  $\mathbf{y}^* \in \mathbb{R}^n$  is identically distributed but independent copy of  $\mathbf{y}$ . The in-sample prediction error is estimated by

$$\frac{1}{n} \|\mathbf{y} - \mathbf{y}(X; T)\|^2 + w,$$

where  $w$  is the expected optimism:

$$w = \mathbb{E}_{\mathbf{y}^*, \mathbf{y}} \left( \frac{1}{n} \|\mathbf{y}^* - \mathbf{y}(X; T)\|^2 - \frac{1}{n} \|\mathbf{y} - \mathbf{y}(X; T)\|^2 \right).$$

If  $\mathbf{y}(X; T) = H\mathbf{y}$ , for some hat matrix  $H$ , then

$$w = \frac{2}{n} \text{tr}(H \text{Cov}(\mathbf{y}, \mathbf{y})).$$

The similarity between  $w$  and the correction in  $CV_c$  in case  $\text{Cov}(\mathbf{y}_{tr}, \mathbf{y}_{te}) = 0$ , i.e., in case  $w_{cv} = \mathbb{E}(2/n \times \text{tr}(H_{cv} \text{Cov}(\mathbf{y}, \mathbf{y})))$ , is interesting since it reflects the relation between generalization error and in-sample error and emphasizes the role of the linearity in this relation.

The fundamental difference between in-sample error and generalization error is that in the latter, the covariates matrices,  $X_{tr}, \mathbf{x}_{te}$  are assumed to be random variables and therefore in generalization error, unlike in in-sample prediction error:

1.  $X_{tr}$  and  $\mathbf{x}_{te}$  are not identical.
2. An expectation is taken also over  $\{X_{tr}, \mathbf{x}_{te}\}$ .

As was mentioned in the previous sections, the inner-sampling mechanism of  $\{T_k, T_{-k}\}$  in CV that emulates repeated sampling of  $\{T_{te}, T_{tr}\}$  from  $\{P_x, P_{y|x,s}\}$  expresses these properties. These properties are also reflected in the correction. Since

$$\frac{2}{n} \text{tr}(H_{cv} \text{Cov}(\mathbf{y}, \mathbf{y})) = \frac{2}{n} \sum_{k=1}^n \mathbf{h}_k \text{Cov}(\mathbf{y}_k, \mathbf{y}_{-k}),$$

then  $2/n \times \text{tr}(H_{cv} \text{Cov}(\mathbf{y}, \mathbf{y}))$  averages  $n$  identically distributed atoms,  $\mathbf{h}_k \text{Cov}(\mathbf{y}_k, \mathbf{y}_{-k})$ , where each one of them is an unbiased estimator of  $w_{cv}$ . Unlike in  $w$ , which relates to a specific covariates matrix realization,  $X$ , the atoms in  $2/n \times \text{tr}(H_{cv} \text{Cov}(\mathbf{y}, \mathbf{y}))$  relate to different covariates realizations,  $\{X_{-k}, \mathbf{x}_k\}_{k=1}^n$ .

In addition, when  $\text{Cov}(\mathbf{y}, \mathbf{y}) = \sigma_o^2 I$ , while  $w_{cv} = 0$ ,  $w = 2\sigma_o^2 / n \times \text{tr}(H)$ . This is since in CV the sample is partitioned into training and test, however in expected optimism approach the whole sample is used for both tasks — training and test.

## 2.5 Advanced Correlation Settings

### 2.5.1 Kriging

Many applications with spatial and temporal data are analyzed using *random functions* framework, rather than multivariate random variable framework. A comprehensive review about random functions data analysis can be found in [Wang et al. \(2016\)](#). Here we focus on interpreting the results from Sections 2.3 and 2.4 in the context of a specific use case in the random functions framework — Kriging with Gaussian process regression ([Rasmussen and Williams 2006](#), GPR). Numerical analysis of  $CV_c$  implementation in real spatial dataset is presented in Section 3.2.2.

In Kriging ([Goovaerts 1999](#)) the goal is to create a climate map on some surface,  $A$ , using climate predictions at a high-resolution grid of the surface. The predictions at the grid points are based on a predictive model that was fitted to a sample,  $T$ , that was drawn from this surface, but covers the surface sparsely. In

many cases, GPR is the predictive modeling method that is used for Kriging. In this method, as well as in other functional data analysis methods, the mean and the covariance of the predicted variable are formulated as functions. The mean function typically depends on fixed effects of some covariates (like elevation). The estimated mean function in GPR is a linear function of  $\mathbf{y}$ . The covariance function, which is termed the *kernel function*,  $K_A : (A \times A) \rightarrow \mathbb{R}$ , measures the covariance between each two points in  $A$ , whether the points are in the sample  $T$  or not. Unlike in the multivariate approach, where the correlation is induced by a latent random variable realization, in the functional approach the correlation structure of the surface  $A$ , as it is expressed by  $K_A$ , is induced by realization of a stochastic process instance — *latent random function*,  $s$ . Since Theorems 2.1 and 2.2 are based on the relation between  $P_{y_k | x_k, T_{-k}}$  and  $P_{y_{te} | x_{te}, T_{-k}}$ , rather than whether the source of the correlation between the observations is a latent random variable or latent random function, these theorems can also be applied here.

Let us consider three scenarios. The first scenario is the classical Kriging use case, where the observations of both samples,  $T_{te}$  and  $T$ , are randomly sampled from the same surface,  $A$ , i.e., observations of both samples are drawn independently from  $\{P_x, P_{y|x, s=s_0}\}$ , where  $s_0$  is the realization of the latent random function  $s$  in the surface  $A$ . In this case Theorem 2.1's condition is satisfied and therefore  $w_{CV} = 0$  and CV is suitable.

The second scenario is when the realization of  $s$  is not the same in  $T_{te}$  and  $T$ , and therefore while the observations of  $T$  follow  $\{P_x, P_{y|x, s=s_0}\}$ , the observation in  $T_{te}$  follows  $\{P_x, P_{y|x, s=s_{te}}\}$ . In this case Theorem 2.1's condition is not satisfied. An example for this scenario is when  $T_{te}$  is sampled from the same surface as  $T$ ,  $A$ , however at a future time-point (e.g., when the goal is to create a climate map for the next year based on this year's data). In this case,  $y_k$  and  $y_{-k}$ , which are sampled at the same time point, are more correlated than  $y_{te}$  and  $y_{-k}$ , which are sampled at different time-points. Therefore, CV is biased. As we saw in Section

2.4, if a linear model (such as GPR) is used, then  $CV_c$  is an unbiased estimator of the generalization error and therefore should be used instead of CV.

Another spatial application in this scenario is when  $T_{te}$  is sampled from the surface  $A'$ , which is different than  $A$ . Since the surfaces are different, then their latent random function realizations are different. Therefore, assuming observations in both samples,  $T$  and  $T_{te}$ , were drawn from the *same marginal distribution* —  $\{P_x, P_{y|x}\}$  —, then  $CV_c$  should be used instead of CV.

Another interesting scenario that is not covered either by Theorem 2.1 or by Theorem 2.2 is when  $\{y_k, x_k\}_{k=1}^n$  and  $\{y_{te}, x_{te}\}$  are drawn from *different marginal distributions* —  $\{P_x, P_{y|x}\}$  and  $\{P_x^{te}, P_{y|x}^{te}\}$ , respectively. An example for this scenario is when Kriging is used for predicting extrapolated spatial points with respect to the sample  $T$ . It may happen due to sampling challenges, such as sampling from mountainous and deep marine regions (Rabinowicz et al. 2020). This scenario, which violates the setting that is assumed in Theorems 2.1 and 2.2 requires further research.

### 2.5.2 Longitudinal Data

Another common setting with correlation structure is longitudinal data — where there are several subjects that are repeatedly observed over time. In this setting, due to the temporal orientation, the correlation structure is more complicated than in a simple clustered data.

Let us consider three scenarios that are equivalent to the three scenarios given in Section 2.5.1. However, unlike in Section 2.5.1, the multivariate framework would be considered (rather than the functional framework).

The first scenario is when the prediction goal is to predict a new observation,  $T_{te}$ , of one of the subjects in  $T$ , sampled at a random time-point from the same distribution that the time-points of the observations in  $T$  follow. Since  $T_{te}$  and  $T$

are sampled from the same subjects, then  $s_{te} = s_0$  and therefore  $P_{y_{te}|x_{te}, T_{-k}} = P_{y_k|x_k, T_{-k}}$ . By Theorem 2.1 this gives  $w_{cv} = 0$ , and therefore CV is suitable in this case.

The second scenario is when the prediction goal is predicting a new observation that relates to a subject that is not in  $T$  and therefore  $s_{te} \neq s_0$  and

$P_{y_{te}|x_{te}, T_{-k}} \neq P_{y_k|x_k, T_{-k}}$ . For this scenario, given that  $\{y_k, x_k\}$  and  $\{y_{te}, x_{te}\}$  were sampled from the same marginal distribution,  $\{P_x, P_{y|x}\}$ , and a linear model is implemented, then by Theorem 2.2,  $CV_c$  should be used instead of CV. Numerical analysis of  $CV_c$  implementation in this scenario is presented in Section 3.1.

Another scenario is when  $\{y_k, x_k\}$  and  $\{y_{te}, x_{te}\}$  are sampled from different marginal distributions,  $\{P_x, P_{y|x}\}$  and  $\{P_x^{te}, P_{y|x}^{te}\}$ , respectively. The marginal distributions can be different due to various reasonable prediction goals, such as forcing the data point in  $T_{te}$  to extrapolates the data points in  $T$  with respect to the time variable, which results in  $x_{te}$  and  $x_k$  being nonidentically distributed. This scenario violates the assumptions in Theorems 2.1 and 2.2 and therefore requires further research.

## 2.6 Comparison with Other Methods

Several cross-validation variants were proposed for settings involving correlated data. Some of them were proposed from a perspective that correlation between the folds causes K-fold CV to underestimate the generalization error. As was shown above, this perception is wrong in many scenarios. Other variants are relevant for very specific applications under various sampling restrictions. Below, several cross-validation variants are described and compared to  $CV_c$ .

One method is h-blocking (Burman et al. 1994), which is mainly relevant for spatial data. In h-blocking, in order to reduce the correlation between the folds, the analyzed surface is partitioned into blocks (folds) that are separated from each other by some distance,  $h$ . As was described above, many use cases do

not require any correlation reduction between the folds and K-fold CV is suitable, however the use of h-blocking is often recommended regardless of the predictive problems setup (Roberts et al. 2017). Let us focus on a scenario when

$P_{y_{te}|x_{te}, T_{-k}} \neq P_{y_k|x_k, T_{-k}}$ , and therefore the condition in Theorem 2.1 is not satisfied, causing K-fold CV to be biased. In this scenario, although the h-blocking approach may seem reasonable, in fact, it suffers from several issues that do not affect  $CV_c$ . For example, frequently, creating the separation between the folds requires omitting observations from the training sample. In addition, the folds that are generated by h-blocking have different distributions, in particular their distributions are different than  $P_{T_{te}}$ . Therefore, h-blocking may provide a biased prediction error estimator with respect to the generalization error. Moreover, since some of the blocks are at the edge of the surface, then the prediction of those blocks becomes predicting spatial extrapolation, which might be inaccurate and does not reflect the planned prediction problem. This implication can affect dramatically the prediction error estimate (Roberts et al. 2017).

Another method is leave cluster out (Rice and Silverman 1991, LCO). This method is relevant for the case when  $\text{Cov}(y_{tr}, y_{te}) = 0$  and the training set,  $T$ , has a clustered correlation structure. LCO eliminates the correlation between the folds by defining each cluster as a fold. This method suffers from several challenges that do not appear in  $CV_c$ . First, using LCO forces the number of folds to be equal to the number of clusters. Another issue is when different clusters contain a substantially different number of observations, in which case LCO prediction error estimator can be biased with respect to the generalization error. In addition, validity of LCO is challenged when some clusters have different distribution than other clusters. In this case, as opposed to the generalization error definition, the observations in  $X_k$  and  $X_{-k}$  are nonidentically distributed.

Another cross-validation variant that is relevant for a balanced longitudinal data setting is leave observation from each cluster out (Wu and Zhang 2002, LOFCO). This method is relevant for the case when the goal is to predict a new

observation for each one of the subjects that appear in the training set. The folds partitioning mechanism in LOFCO is that different folds refer to different time-points, such that each fold contains the observations that were collocated at the same time-point across all the subjects. This partitioning is feasible due to the balanced data design assumption. The challenges in this method are similar to the challenges mentioned above: it requires a balanced data design, the number of folds are forced by the data structure,  $X_k$  and  $X_{-k}$  are not identically distributed — in particular their time-points covariate is nonidentically distributed.

LCO, LOFCO and h-blocking reflect the understanding that nonstandard CV may be needed in presence of correlations, and offer solutions for very specific types of datasets and correlation settings that apply to any modeling technique. In contrast,  $CV_c$  can be applied in a wide range of types of datasets and correlation settings, however it is limited to linear models.

Several papers ([Altman 1990](#); [Francisco-Fernandez and Opsomer 2005](#)) consider non-parametric modeling with correlated errors and propose versions of CV and generalized cross-validation (GCV) which account for this correlation, for the purpose of selecting bandwidth for estimating the trend of time series and spatial datasets (respectively), which is a different goal than our objective of estimating prediction error. We omit the details for brevity, but note that in the same model settings, our results can be used to derive corrected CV estimates of prediction error that take the correlation into account, and are different than those offered in these papers. In simulated comparisons, we have confirmed that indeed the corrected versions differ in practice, and  $CV_c$  is the proper correction for our setting of interest.

## 2.7 Estimating $\text{Cov}(y, y)$

In most applications  $\text{Cov}(y, y)$  is unknown in advance and therefore in order to implement  $CV_c$ , it should be estimated. In many cases, such as when GLS is implemented, the covariance matrix is already estimated in the model fitting



stage and can also be used in the  $CV_c$  correction term. Plugging-in the estimated covariance matrix instead of the true covariance matrix can add variance to  $CV_c$  and even can make  $CV_c$  biased. This may especially occur for small sample sizes and in a non-parametric context, when the estimation of the covariance matrix can be a difficult task.

Several papers analyze the effect of plugging-in the estimated covariance matrix instead of the true one in various data-based corrected CV methods (Hart and Wehrly 1986; Altman 1990; Francisco-Fernandez and Opsomer 2005). All of them show numerically that plugging-in the estimated covariance matrix instead of the true covariance matrix has a minimal effect on the correction, and therefore recommend to plug-in the estimated covariance matrix. Using simulations and real data analysis, we also show in Section 3, that the effect of plugging-in the estimated covariance matrices in  $CV_c$  is small, especially when the sample size of the training set is not very small.

This topic is also discussed in context of conditional AIC (Vaida and Blanchard 2005; Liang et al. 2008; Greven and Kneib 2010; Rabinowicz et al. 2020) where the bias correction term also contains the covariance matrix. While Vaida and Blanchard claim that the effect of using the estimated covariance matrix instead of the true one is minimal and therefore recommend to plug it in, Greven and Kneib show that conditional AIC is biased in selecting random effects under some settings and present a new conditional AIC version that takes into account the use of the estimated covariance matrix. Greven and Kneib claim that the bias is caused since the penalty term is constructed using the same observations that determine the likelihood. Therefore, conditional AIC tends to select the larger model for the same reason that the maximum likelihood model selection criterion selects the larger model. This mechanism is of course related to penalized likelihood methods like AIC, but not relevant for CV based methods which partition the sample into a training set for fitting the model and a test set for assessing the trained model.

In summary, there is empirical evidence that the effect of using estimated covariance in data-based corrected CV methods is minor. However, a careful theoretical study of the effect of using the estimated covariance matrix instead of the true one in CV methods, and in particular  $CV_c$ , is a topic for future research.

### 3 NUMERICAL RESULTS

This section compares  $CV_c$  and CV, with respect to the approximated generalization error, using simulation and real datasets analyses. Different prediction goals and correlation structures are analyzed. Relevant datasets and code can be found in <https://github.com/AssafRab/CVc>.

#### 3.1 Simulation

The dependent variable,  $y \in \mathbb{R}^n$ , was sampled from the following model:

$$y_{i,j,k} = 0.1 \sum_{r=1}^9 x_{i,j,k,r} + u_i + \sum_{r=1}^2 z_{i,j,k,r} b_{i,j,r} + \delta_{i,j,k},$$

$$i \in \{1, \dots, I\}, j \in \{1, \dots, 5\}, k \in \{1, \dots, 10\},$$

where

- the random effects,  $u_i, \mathbf{b}_{i,j} = [b_{i,j,1}, b_{i,j,2}]$  and  $\delta_{i,j,k}$  are independent and distributed as follows:

$$u_i \stackrel{\text{ind}}{\sim} N(0, 3^2), \mathbf{b}_{i,j} = [b_{i,j,1}, b_{i,j,2}] \in \mathbb{R}^2 \stackrel{\text{ind}}{\sim} N(0, \begin{bmatrix} 3^2 & 0 \\ 0 & 1 \end{bmatrix}), \delta_{i,j,k} \stackrel{\text{ind}}{\sim} N(0, 1),$$

- the covariates,  $\{\mathbf{x}_r\}_{r=1}^9$  are:
  - $x_{i,j,k,1} = 1, x_{i,j,k,2} = k, \forall i, j, k$  are the intercept and the time covariates,
  - $x_{i,j,k,r} = \eta_i + \delta_{i,j,k}, \forall i, j, k$  and  $\forall r \in \{3, \dots, 9\}$ , where  $\eta_i \stackrel{\text{ind}}{\sim} N(0, 1), \delta_{i,j,k} \stackrel{\text{ind}}{\sim} N(0, 1)$  are independent,

- $z_{i,j,k,1} = 1, z_{i,j,k,2} = k \forall i, j, k$  are the covariates for random intercept and random slope.

This settings was simulated for three different sample sizes,  $n = 300 / 400 / 500$  (where  $/$  varies respectively to  $6 / 8 / 10$ ).

This is a hierarchical clustered structured with  $/$  clusters of 50 observations each, where within each cluster there are five subclusters of ten observations each. There is a random intercept for the high-level clusters and random intercept and slope for the subclusters. Therefore the covariance of  $y \mid x_1, \dots, x_9$  is:

$$\text{Cov}(y_{i,j,k}, y_{i',j',k'}) = 9 \times 1_{[i=i']} + 1_{[i=i', j=j']} (9 + 1 \times k \times k') + 1_{[i=i', j=j', k=k']}.$$

GLS was fitted using LOO for eight nested models. Model 1 contains the intercept and time, Model 2 also contains  $x_3$ , and so on. Model 8 contains all the covariates.

### 3.1.1 Estimating Prediction Error

$T_{tr}$  is a random subset of  $T$  of size  $n - 1$ .  $T_{te}$  is a single observation, drawn from the same marginal distribution, however with new independent realizations of all the random effects. Therefore, when implementing CV, while in  $T$ :

$$\text{Cov}(y_{-k}, y_k) \neq 0 \quad \forall k \in \{1, \dots, n\},$$

in  $\{T_{tr}, T_{te}\}$ :

$$\text{Cov}(y_{tr}, y_{te}) = 0.$$

The generalization error was approximated by averaging  $(y_{te} - \hat{y}(x_{te}; T_{tr}))^2$ , based on  $1000 \times n$  samples of  $\{T_{te}, T_{tr}\}$ . The densities of CV and  $CV_c$  were approximated based on 1000 samples of  $T$ . Since in this setting  $\text{Cov}(y_{tr}, y_{te}) = 0$ , the correction in this case is  $2 / n \times \text{tr}(H_{cv} \text{Cov}(y, y))$ .

Figure 1 shows the distribution of CV and  $CV_c$  including their means for the saturated model when  $n = 400$ , compared to the generalization error. Two versions of CV and  $CV_c$ , are presented — when the variance parameters are known, and when they are estimated.

As can be seen,  $CV_c$  is an unbiased estimator of the generalization error, while CV estimator is biased. This is true for both versions, when the variance parameters are known and when they are estimated. Also, as expected, estimating the variance parameters increases the variance of  $CV_c$ . Still, the density of the  $CV_c$  version with the estimated variance parameters is similar to the version with the known variance parameters — their averages are 60.14 and 60.09 (compared to 60.00 for the approximated generalization error), their standard deviations are 12.07 and 13.02, respectively.

In order to assess the performance of  $CV_c$  version with the estimated variance parameters, compared to the version with the known variance parameters, a two sample Anderson-Darling test ([Anderson and Darling 1952](#)) was used. The tested statistic is:

$$CV_c - \text{generalization error},$$

where one sample uses  $CV_c$  version with the estimated variance parameters, and the other sample uses  $CV_c$  version with the true variance parameters. Implementing the function `ad.test` of the package `kSamples` in R software, the p-value of the test is 0.33. The result indicates that in this setting there is no evidence for significant difference between the distribution of  $CV_c - \text{generalization error}$  when the variance parameters are known in advance or estimated.

Figure 2 shows the same analysis as in Figure 1, however for various sample sizes.

As can be seen in Figure 2, the variance of the two  $CV_c$  versions becomes similar as the sample size increases. While for  $n = 300$  the standard deviations of the version with the true variance parameters is 14.43 and for the estimated is 16.04, when  $n = 500$  they are 11.25 and 11.93, respectively. This is expected, since larger sample sizes provide more accurate variance parameters estimators. Also, as the sample size increases the CV bias,  $w_{cv}$ , decreases. This is specific to our setting, where  $\text{Cov}(\mathbf{y}, \mathbf{y})$  becomes more sparse as the sample size increases and therefore  $E[2/n \times \text{tr}(H_{cv} \text{Cov}(\mathbf{y}, \mathbf{y}))]$  decreases.

### 3.1.2 $\text{Cov}(\mathbf{y}_{tr}, \mathbf{y}_{te}) \neq 0$ and Model Selection

In this simulation  $CV_c$  is analyzed under a different prediction goal than in Section 3.1.1: estimating the generalization error of new observations from the *same* high-level clusters, but from a *different* subcluster, i.e.,

$$y_{te} = 0.1 \sum_{r=1}^9 x_{te,r} + u_i + \sum_{r=1}^2 z_{te,r} b_{te,r} + \epsilon_{te},$$

where  $u_i$  is a random effect realization that appears in the training sample, and  $\mathbf{b}_{te} = [b_{te,1}, b_{te,2}]^T$  is a new random effect realization vector. The new prediction goal derives a different correction — since in this setting  $y_{te}$  is correlated with  $\mathbf{y}_{tr}$  through the random effect  $u$ , then the correction in this case is  $2/n \times \text{tr}(H_{cv} \text{Cov}(\mathbf{y}, \mathbf{y} | u_1, u_2, \dots, u_I))$ , for more details see Section 2.4.1.

Two different predictive algorithms were implemented:

- LMM, which is the model that should be used in this setting under the normality assumption, since it utilizes explicitly the correlation between  $y_{te}$  and  $\mathbf{y}$ , using the relevant estimated random effects realizations,  $\hat{u}_1, \dots, \hat{u}_I$ .
- GLS, which is inferior to LMM under the normality assumption, however unlike LMM, GLS is relevant for use cases where distributional assumptions cannot be taken. This is since GLS can be interpreted as an

extension of the least squares algorithm, which does not rely on distributional assumptions.

Figure 3 shows the model density estimation of  $CV_c$  and CV compared to the approximated generalization error in the new setting when LMM and GLS are implemented.

As can be seen in Figure 3,  $CV_c$  estimates the generalization error unbiasedly, while CV underestimates it. The CV bias in Figure 3b is relatively small with respect to the biases in Figure 3a and Figure 2:

- The reason that the bias in Figure 3b is smaller than the bias in Figure 2 although in both settings GLS is implemented, is that the deviation of  $P_{y_{te}|x_{te},T_{tr}}$  from  $P_{y_k|x_k,T_{-k}}$  is smaller in the setting of Figure 3b, as expressed in the corrections —  $2/n \times \text{tr}(H_{cv} \text{Cov}(y, y | u_1, u_2, \dots, u_I))$  in the setting of Figure 3b and  $2/n \times \text{tr}(H_{cv} \text{Cov}(y, y))$  in the setting of Figure 2.
- The reason that the bias in Figure 3b is smaller than the bias in Figure 3a although the deviation of  $P_{y_{te}|x_{te},T_{tr}}$  from  $P_{y_k|x_k,T_{-k}}$  is the same in both settings is that in Figure 3a LMM is implemented and therefore the correlation setting is utilized more than in Figure 3b. While in GLS the realization of  $\mathbf{b}$  affects the fixed effects estimates (see Section 2.4.2), in LMM the realization of  $\mathbf{b}$  affects both, the fixed effects estimates and the estimates of  $u_1, \dots, u_I$ .

For demonstrating the model selection performance, Figure 4 presents the agreement rates of  $CV_c$  and CV with the approximated generalization error, over the repeated simulation runs. The results are presented for different sample sizes and for different predictive models (LMM and GLS).

As can be seen from Figures 4, when LMM is implemented  $\argmin_{h \in H} CV_c(h)$  performs better than  $\argmin_{h \in H} CV(h)$ , however when GLS is implemented both

criteria perform the same. It is important to note that in order to have a high agreement rate with the oracle, estimating unbiasedly the generalization error, as  $CV_c$  does, is not enough since the variance of the estimator can mix the ranks of the models relative to the oracle. Therefore, there is no guarantee that  $\argmin_{h \in H} CV_c(h)$  performs better than  $\argmin_{h \in H} CV(h)$ . This also applies to other model selection criteria that are based on comparing the models' estimated prediction errors. However, in practice, as we see here and in the real dataset examples in the next section,  $\argmin_{h \in H} CV_c(h)$  performs better than  $\argmin_{h \in H} CV(h)$  when the CV bias is large.

In addition, for LMM, both  $\argmin_{h \in H} CV_c(h)$  and  $\argmin_{h \in H} CV(h)$  with the true variance parameters perform better than the versions with the estimated variance parameters, while for GLS both perform similarly. This can be explained by the key role of the variance in predicting using LMM, compared to predicting using GLS. Also, when the sample size is small ( $n = 300$ ), it is more difficult to estimate accurately the variance parameters, and the difference between the two  $\argmin_{h \in H} CV_c(h)$  versions is substantial.

### 3.2 Real Data Analysis

This section presents analysis of two real datasets, a dataset with a clustered correlation structure, and a dataset with hierarchical spatial correlation structure.

#### 3.2.1 Black Friday Dataset — Clustered Correlation Structure

The Black Friday dataset<sup>2</sup> contains information of 737, 577 transactions made in a retail store on Black Friday by 5, 891 customers. The median number of transactions by a customer is 53, and the range is [5,1025]. The dataset is available in <https://github.com/AssafRab/CVc>.

A training sample of all transactions from 100 random customers was drawn and the goal is to fit a predictive model that predicts the purchase amount of a new

transaction of a new random customer. Therefore, while in CV setting  $\text{Cov}(\mathbf{y}_{-k}, \mathbf{y}_k) \neq 0$ , in the prediction goal setting  $\text{Cov}(\mathbf{y}_{tr}, \mathbf{y}_{te}) = 0$ . Therefore the condition of Theorem 2.1 is not satisfied ( $P_{T_{te}, T_{tr}} \neq P_{T_k, T_{-k}}$ ) and K-fold CV is unsuitable.

Three GLS models were fitted using different covariates (see Table 1) with the covariance matrix  $\text{Cov}(y_{i,j}, y_{i',j'}) = \sigma_{customer}^2 \mathbf{1}_{[i=i']} + \sigma_0^2 \mathbf{1}_{[i=i', j=j']}$ , where  $i$  is customer index and  $j$  is the index for an observation in a customer's set of observations.  $\sigma_{customer}^2$  and  $\sigma_0^2$  were estimated using restricted maximum likelihood of normal distribution.

CV and  $CV_c$  were calculated using the training sample. Test error was calculated by averaging the prediction error of observations of other random 900 customers using GLS models that were fitted by  $T$ . This procedure was repeated 50 times with different random training and test sets, and the generalization error was approximated by averaging the test errors of the 50 runs.

Figure 5 presents the means of CV,  $CV_c$ , and the test error over the 50 runs, as well as the confidence intervals of the means.

As can be seen in Figure 5,  $CV_c$  estimates the generalization error better than CV for all the three models. Also, unlike CV,  $CV_c$  selects the same model as the oracle, Model 1, and follows properly the prediction error estimation trend across the models. Analyzing the business aspect of this example, leads to the surprising conclusion that including only the product category as a fixed effect gives the best prediction model, and adding covariates like age and gender does not improve predictive performance, when the data are properly analyzed using  $CV_c$ .

### 3.2.2 California Housing Dataset — Clustered Random Field Correlation Structure



Another dataset that was analyzed is the California housing dataset, which contains house values and other housing parameters in California. The dataset has a hierarchical spatial correlation structure — each observation has spatial coordinates value, where some of the observations share the same coordinates value and therefore define a cluster. This correlation structure is similar to the correlation structure in Sections 3.1, however with spatial data rather than longitudinal data. The dataset is available in the python scikit-learn package and the code is available in <https://github.com/AssafRab/CVc>.

To emphasize the hierarchical clustering structure, all the clusters containing only a single observation were excluded from the analysis (8, 237 observations out of 20, 640), so the analyzed dataset contains 12, 403 observations from 4, 353 clusters. The median number of observations in a cluster is 2 and the range is [2,15].

A training sample,  $T$ , of 700 independent clusters was randomly drawn. The prediction goal is to predict the 'median house value in a block' of a new observation from a *different* cluster than the clusters in  $T$ . Three GPR models with different covariates were fitted (see Table 2) with the following covariance function:

$$\text{Cov}(y_i, y_{i'}) = K_{exp}(z_i - z_{i'}) + \sigma^2 \times 1_{[i=i']},$$

where  $z_i \in \mathbb{R}^2$  contains the latitude and longitude of observation  $i$  and  $K_{exp}(\cdot)$  is the exponential kernel function. The parameters of  $K_{exp}(\cdot)$  and  $\sigma^2$  were fitted using maximum likelihood of normal distribution.

Since the prediction goal is to predict a new observation from a different cluster than the clusters in  $T$ , then  $\text{Cov}(\mathbf{y}_{-k}, y_k) \neq \text{Cov}(\mathbf{y}_{-k}, y_{te})$ . Therefore the condition of Theorem 2.1 is not satisfied and standard K-fold CV is unsuitable. It is important to note that unlike the Black Friday example, here  $\text{Cov}(\mathbf{y}_{-k}, y_{te}) \neq 0$  although  $y_{te}$  is from a new cluster. This is due to the spatial correlation structure of the dataset,

i.e., the latent random function  $s$ . The  $CV_c$  correction in this case is  $2/n \times \text{tr}(H_{cv} \text{Cov}(\mathbf{y}, \mathbf{y} | s))$ , where  $\text{Cov}(\mathbf{y}, \mathbf{y} | s)$ , which expresses the extra correlation that the clusters contribute over the spatial correlation of the data (see Section 2.4.1), is estimated as follows:

$$\text{Cov}(\mathbf{y}, \mathbf{y} | s)[i, i'] = \begin{cases} K_{exp}(0) - \frac{1}{|j, j' \mid \forall \mathbf{z}_j \neq \mathbf{z}_{j'}|} \sum_{j, j' \mid \forall \mathbf{z}_j \neq \mathbf{z}_{j'}} K_{exp}(\mathbf{z}_j - \mathbf{z}_{j'}) & \mathbf{z}_i = \mathbf{z}_{i'} \\ 0 & \mathbf{z}_i \neq \mathbf{z}_{i'} \end{cases}.$$

The analysis was repeated 50 times with different random  $T$ , at each run CV and  $CV_c$  were calculated and the test error was calculated by averaging the prediction error of the remaining observations of the 3, 653 clusters that are not in  $T$  using GPR models that were fitted by  $T$ . Averaging the test error over the 50 runs approximates the generalization error.

Figure 6 presents the means of  $CV$ , and  $CV_c$ , and the test error over the 50 runs, as well as the confidence intervals of the means.

Similarly to Figure 5,  $CV_c$  estimates the generalization error better than CV for all the models, selects the same model as the oracle (Model 3), and follows properly the prediction error estimation trend across the models. Also, while in Figure 5 CV underestimates the prediction error of the saturated model more than the smaller models, here CV underestimates the saturated model less than the smaller models. Therefore, biasedness of CV prediction error can be expressed in model selection in different ways — sometimes by favoring over-parameterized models and sometimes by favoring under-parameterized models.

## 4 Conclusion and Discussion

In this paper we tackle the problem of applying CV as an estimate of generalization error in non-i.i.d. situations. While the fundamental concerns that this presents are widely acknowledged, a clear understanding of when adjustments are needed, and what adjustments are appropriate, seems lacking

in much of the literature ([Roberts et al. 2017](#); [Saeb et al. 2017](#); [Anderson et al. 2018](#)).

We first present a general formulation of the bias in using CV in presence of correlations, which leads to a clear general definition of settings where no correction to CV is needed (Theorem 2.1). It shows that non-i.i.d. situations can still facilitate correctness of regular CV, as long as the dependence structure between training and prediction points is consistent in CV and actual prediction. This simple result appears to contradict some previous claims in the literature. An example can be found in [Roberts et al. \(2017\)](#), where mechanisms of controlling folds partitioning are suggested for Kriging tasks, based on the conception that CV is always over-optimistic for non-i.i.d. data.

We then present a derivation of a bias correction for linear models under general correlation structures (Theorem 2.2), which is used in establishing a bias corrected cross-validation version,  $CV_c$ . To implement our correction, it is necessary to specify the covariance structures within the training set and between the training and prediction sets. This is typically also required to choose a modeling approach. For example, in a simple linear mixed model with normal assumptions, if one assumes that the random effect realizations are the same when predicting, then LMM prediction is appropriate, while if random effect realizations are new, then using GLS for estimation of fixed effects only for prediction is more appropriate ([Verbeke 1997](#)). However, it is important to emphasize that the validity of the correction  $CV_c$  does not depend on selection of an appropriate modeling approach. In other words, if one mistakenly uses GLS where LMM is appropriate,  $CV_c$  still gives an unbiased estimate of generalization error for the resulting model, as long as the covariance structure is correctly specified.

In practice, the covariance matrices are typically not fully known, but partially estimated from the data (for example, variance parameters in LMM can be

estimated using restricted maximum likelihood, [Verbeke \(1997\)](#)), and this is also required for applying  $CV_c$  to correct CV results. This could potentially add uncertainty to the estimates, but as discussed in Section 2.7 and demonstrated in Sections 3, it does not tend to affect their expectation.

A fundamental assumption that is taken throughout this paper is that the marginal distributions of the training and prediction sets are the same. In case the marginal distributions are different, i.e., the observations in the training set are drawn from a different marginal distribution than the prediction data, then  $CV_c$  is unsuitable. This scenario requires further research and relevant applications are given in Sections 2.5.1 and 2.5.2. Other important use cases of different marginal distributions for training and prediction sets are discussed in [Rabinowicz et al. \(2020\)](#) and solutions for these use cases are proposed in context of transductive prediction error (rather than generalization error approach taken here).

An interesting situation also covered by  $CV_c$  and not often discussed, is when the training set contains i.i.d. observations, but new data points where predictions are made are actually correlated with the training set (for example, new observations in the same set of cities). In this case, if the correlation structure is known, then  $CV_c$  can still be used to estimate the prediction error.

## Notes

<sup>1</sup>We assume the observations are randomly ordered, so the folds are exchangeable.

<sup>2</sup>Presented by [AnalyticsVidhya](#).

## References

Altman, N. S. (1990), "Kernel smoothing of data with correlated errors," *Journal of the American Statistical Association*, 85, 749–759.

Anderson, C., Hafen, R., Sofrygin, O., Ryan, L., and HBGDKi Community (2018), "Comparing predictive abilities of longitudinal child growth models," *Statistics in Medicine*, 38, 3555–3570.

Anderson, T. W. and Darling, D. A. (1952), "Asymptotic theory of certain " goodness of fit" criteria based on stochastic processes," *The Annals of Mathematical Statistics*, 23, 193–212.

Arlot, S., Celisse, A., et al. (2010), "A survey of cross-validation procedures for model selection," *Statistics Surveys*, 4, 40–79.

Breiman, L. (2001), "Random forests," *Machine Learning*, 45, 5–32.

Breslow, N. E. and Clayton, D. G. (1993), "Approximate inference in generalized linear mixed models," *Journal of the American statistical Association*, 88, 9–25.

Burman, P. (1989), "A comparative study of ordinary cross-validation, v-fold cross-validation and the repeated learning-testing methods," *Biometrika*, 76, 503–514.

Burman, P., Chow, E., and Nolan, D. (1994), "A cross-validatory method for dependent data," *Biometrika*, 81, 351–358.

Efron, B. (1986), "How biased is the apparent error rate of a prediction rule?" *Journal of the American statistical Association*, 81, 461–470.

Fang, Y. (2011), "Asymptotic equivalence between cross-validations and Akaike information criteria in mixed-effects models," *Journal of Data Science*, 9, 15–21.

Francisco-Fernandez, M. and Opsomer, J. D. (2005), "Smoothing parameter selection methods for nonparametric regression with spatially correlated errors," *Canadian Journal of Statistics*, 33, 279–295.

- Friedman, J. H. (2002), "Stochastic gradient boosting," *Computational Statistics & Data Analysis*, 38, 367–378.
- Gelman, A. (2006), "Multilevel (hierarchical) modeling: what it can and cannot do," *Technometrics*, 48, 432–435.
- Goovaerts, P. (1999), "Geostatistics in soil science: state-of-the-art and perspectives," *Geoderma*, 89, 1–45.
- Greven, S. and Kneib, T. (2010), "On the behaviour of marginal and conditional AIC in linear mixed models," *Biometrika*, 97, 773–789.
- Hand, D. J. (2017), *Practical longitudinal data analysis*, Routledge.
- Hansen, C. B. (2007), "Generalized least squares inference in panel and multilevel models with serial correlation and fixed effects," *Journal of Econometrics*, 140, 670–694.
- Hart, J. D. and Wehrly, T. E. (1986), "Kernel regression estimation using repeated measurements data," *Journal of the American Statistical Association*, 81, 1080–1088.
- Harville, D. et al. (1976), "Extension of the Gauss-Markov theorem to include the estimation of random effects," *The Annals of Statistics*, 4, 384–395.
- Hastie, T., Tibshirani, R., and Friedman, J. (2009), *The elements of statistical learning: data mining, inference, and prediction*, New York, NY: Springer.
- Hodges, J. S. and Sargent, D. J. (2001), "Counting degrees of freedom in hierarchical and other richly-parameterised models," *Biometrika*, 88, 367–379.
- Jordan, M. (2004), "Graphical models," *Statistical Science*, 19, 140–155.

Lee, Y. and Nelder, J. A. (1996), "Hierarchical generalized linear models," *Journal of the Royal Statistical Society: Series B (Methodological)*, 58, 619–656.

Liang, H., Wu, H., and Zou, G. (2008), "A note on conditional AIC for linear mixed-effects models," *Biometrika*, 95, 773–778.

Little, M. A., Varoquaux, G., Saeb, S., Lonini, L., Jayaraman, A., Mohr, D. C., and Kording, K. P. (2017), "Using and understanding cross-validation strategies. Perspectives on Saeb et al." *GigaScience*, 6, 1–6.

Maddison, W. P. (1990), "A method for testing the correlated evolution of two binary characters: are gains or losses concentrated on certain branches of a phylogenetic tree?" *Evolution*, 44, 539–557.

Miller, K. S. (1981), "On the inverse of the sum of matrices," *Mathematics Magazine*, 54, 67–72.

Rabinowicz, A., Rosset, S., et al. (2020), "Assessing prediction error at interpolation and extrapolation points," *Electronic Journal of Statistics*, 14, 272–301.

Rasmussen, C. E. and Williams, C. K. (2006), *Gaussian processes for machine learning*, MIT press Cambridge.

Raudenbush, S. W. and Bryk, A. S. (2002), *Hierarchical linear models: Applications and data analysis methods*, vol. 1, Sage.

Rice, J. A. and Silverman, B. W. (1991), "Estimating the mean and covariance structure nonparametrically when the data are curves," *Journal of the Royal Statistical Society: Series B (Methodological)*, 53, 233–243.

Roberts, D. R., Bahn, V., Ciuti, S., Boyce, M. S., Elith, J., Guillera-Aroita, G., Hauenstein, S., Lahoz-Monfort, J. J., Schröder, B., Thuiller, W., et al. (2017), "

Cross-validation strategies for data with temporal, spatial, hierarchical, or phylogenetic structure,” *Ecography*, 40, 913–929.

Saeb, S., Lonini, L., Jayaraman, A., Mohr, D. C., and Kording, K. P. (2017), “The need to approximate the use-case in clinical machine learning,” *GigaScience*, 6, 1–9.

Stone, M. (1974), “Cross-validatory choice and assessment of statistical predictions,” *Journal of the Royal Statistical Society: Series B (Methodological)*, 36, 111–133.

Vaida, F. and Blanchard, S. (2005), “Conditional Akaike information for mixed-effects models,” *Biometrika*, 92, 351–370.

Verbeke, G. (1997), “Linear mixed models for longitudinal data,” in *Linear mixed models in practice*, Springer, pp. 63–153.

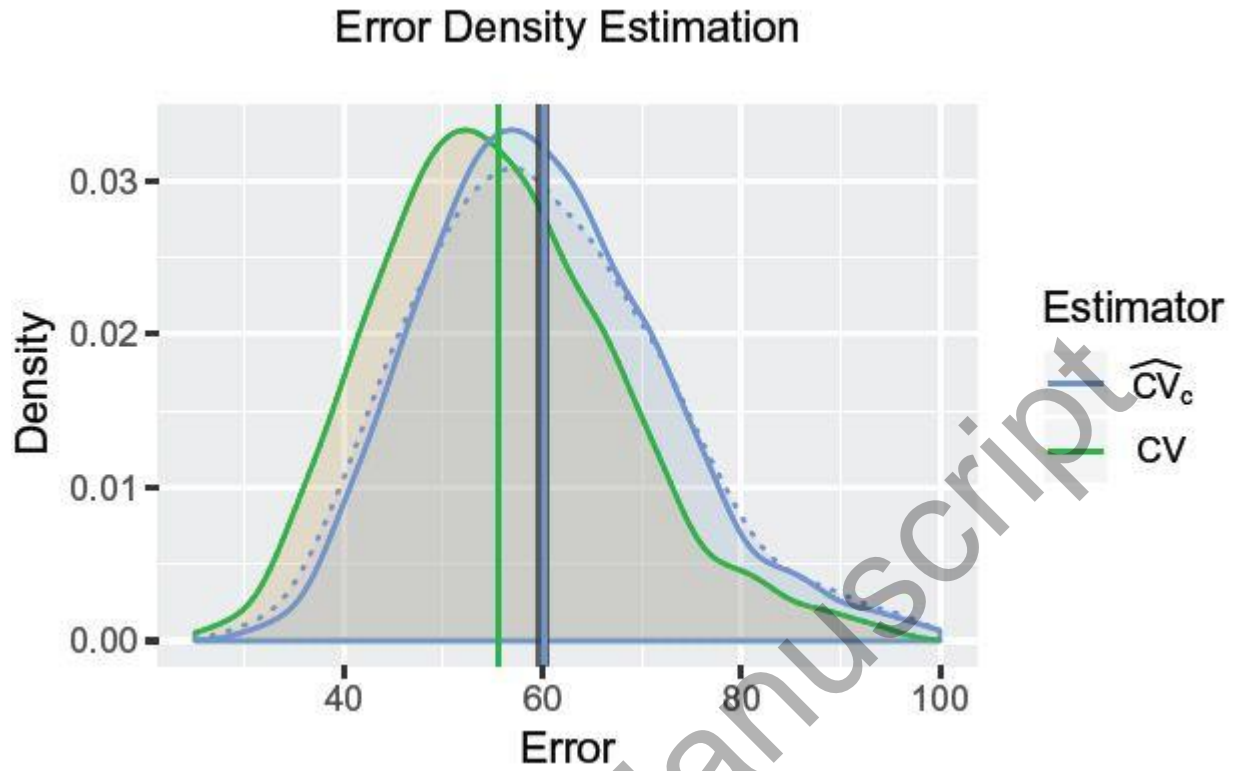
Wang, J.-L., Chiou, J.-M., and Müller, H.-G. (2016), “Functional data analysis,” *Annual Review of Statistics and Its Application*, 3, 257–295.

Ward, M. D. and Gleditsch, K. S. (2018), *Spatial regression models*, vol. 155, Sage Publications.

Wu, H. and Zhang, J.-T. (2002), “Local polynomial mixed-effects models for longitudinal data,” *Journal of the American Statistical Association*, 97, 883–897.

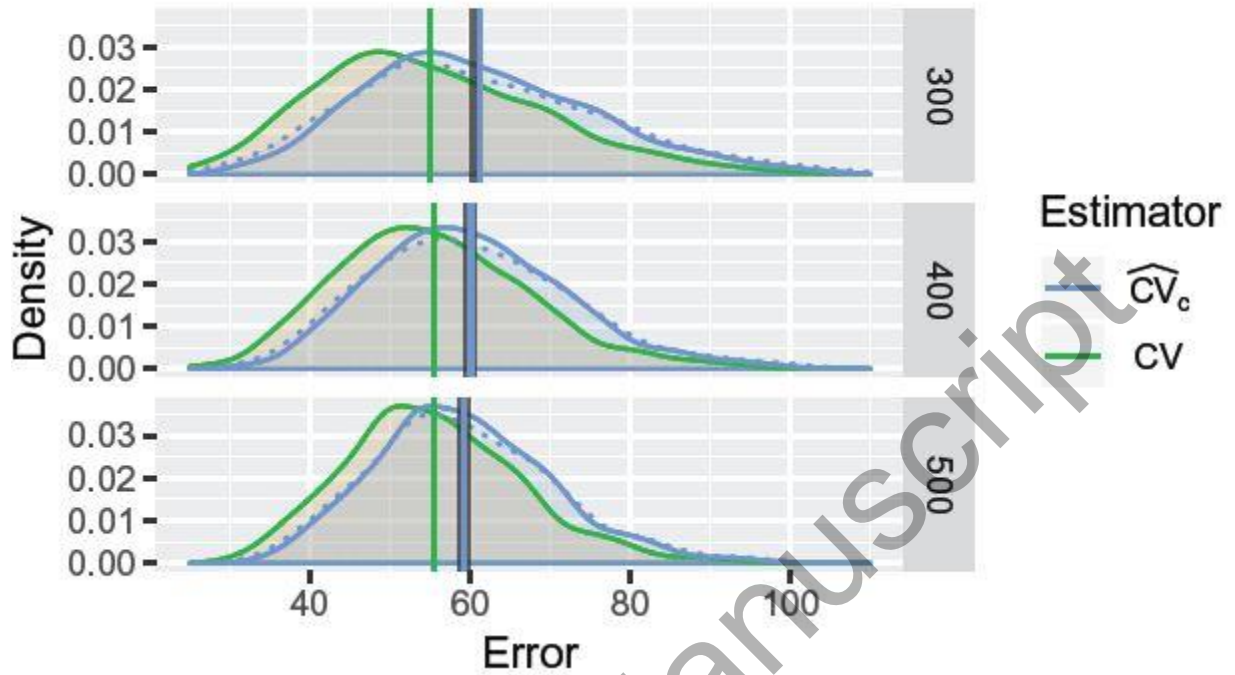
Zhang, Y. and Yang, Y. (2015), “Cross-validation for selecting a model selection procedure,” *Journal of Econometrics*, 187, 95–112.



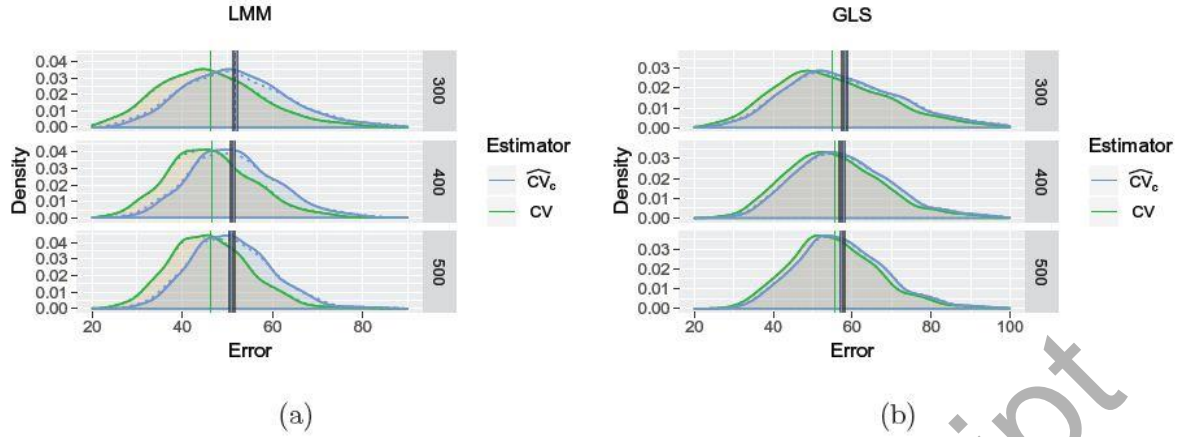


**Fig. 1** Densities of CV and  $\widehat{CV}_c$  and their means compared to the generalization error (in black). Two scenarios are presented: when the variance parameters are known (solid line) and when they are estimated (dashed line).

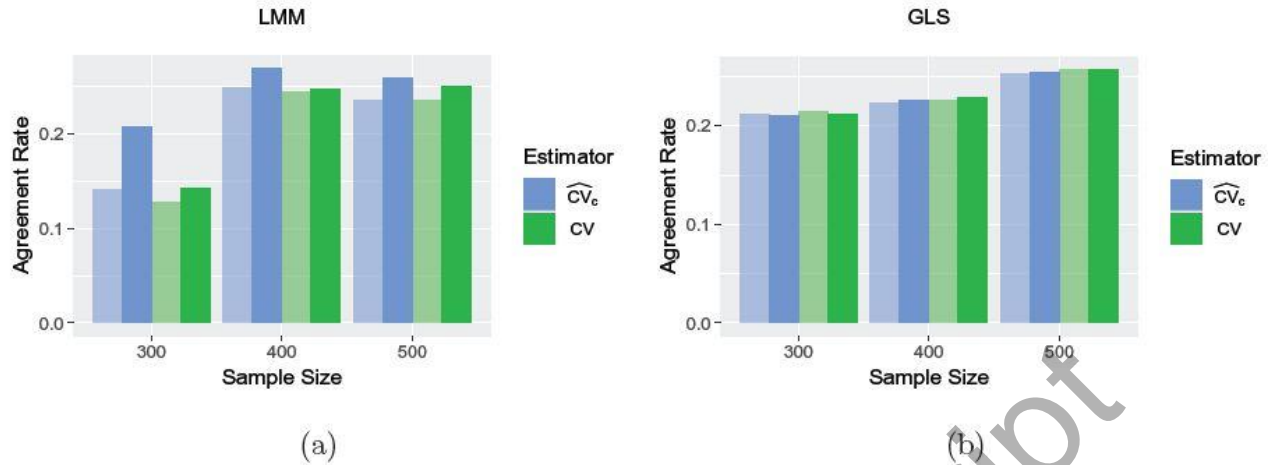
## Error Density Estimation for Different Sample Sizes



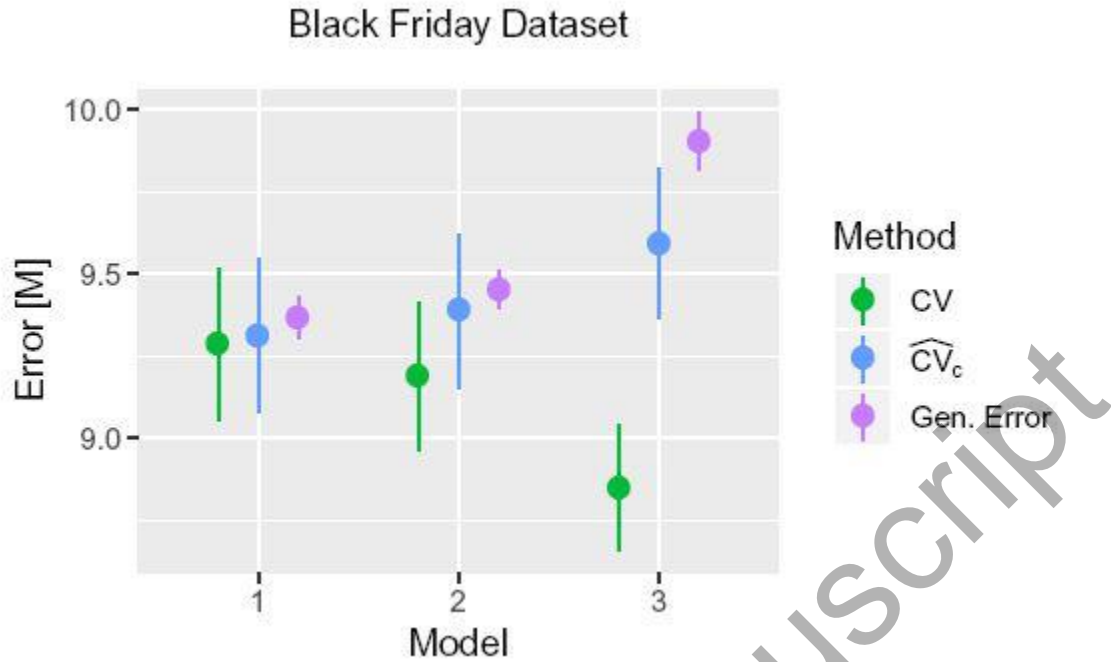
**Fig. 2** Densities of  $CV$  and  $\widehat{CV}_c$  and their means compared to the generalization error (in black) for different sample sizes. Two scenarios are presented: when the variance parameters are known (solid line) and when they are estimated (dashed line).



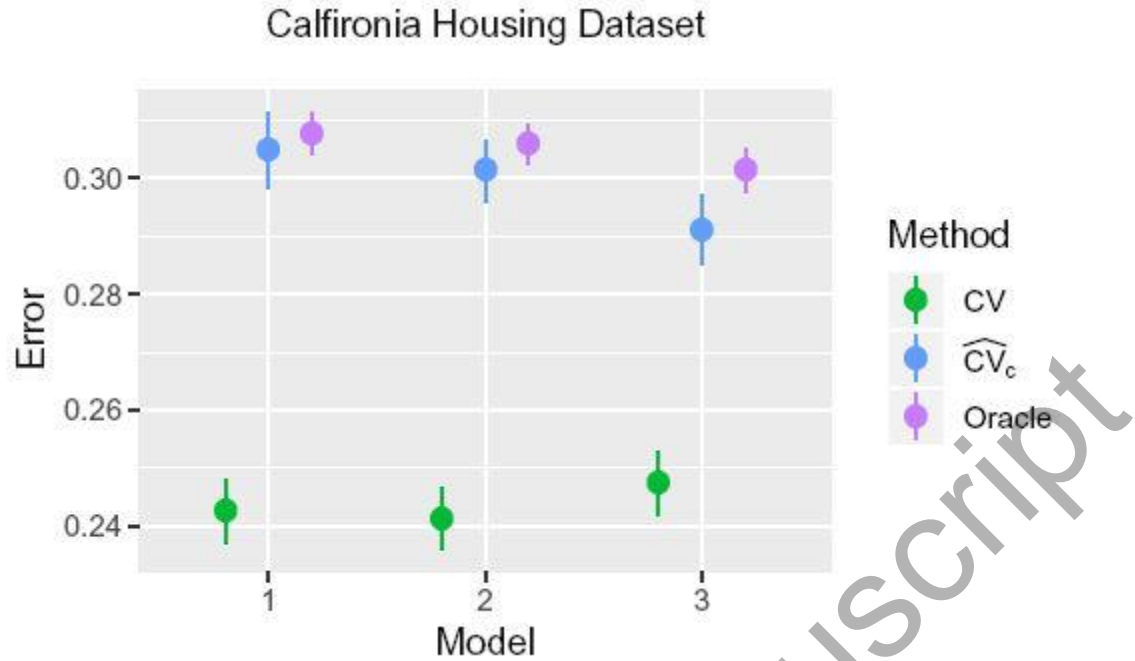
**Fig. 3** (3(a)): Using LMM, each curve refers to the density of the relevant estimator,  $CV$  and  $CV_e$ . Two scenarios are presented: when the variance parameters are known (solid line) and when they are estimated (dashed line). Their means are compared to the generalization error (in black). (3(b)): Similarly, the analysis is presented when GLS is implemented instead of LMM.



**Fig. 4** (4(a)): Each bar refers to the agreement rate of the relevant criterion,  $\argmin_{h \in H} CV_c(h)$  and  $\argmin_{h \in H} CV(h)$ , with  $\argmin_{h \in H} generalization\ error(h)$  for different sample sizes. Also, two scenarios are presented: when the variance parameters are known (in dark color) and when they are estimated (in light color). (4(b)): Similarly, the analysis is presented when GLS is implemented instead of LMM.



**Fig. 5** Prediction error estimation. For each model, the purple point is the approximated generalization error, which was calculated by averaging the test error over the 50 runs. The lines range is the two standard deviations confidence interval of the mean. Similarly, the mean estimated prediction error and the two standard deviations confidence interval of CV and  $\widehat{CV}_c$  are presented.



**Fig. 6** Prediction error estimation. For each model, the purple point is the approximated generalization error, which was calculated by averaging the test error over the 50 runs. The lines range is the two standard deviations confidence interval of the mean. Similarly, the mean estimated prediction error and the two standard deviations confidence interval of CV and  $\widehat{CV}_c$  are presented. The unit of the dependent variable is in \$100,000.

**Table 1** Model 1 contains two covariates, model 2 contains five covariates and model 3 contains eight covariates.

Covariate\Model	Model 1	Model 2	Model 3
Intercept	✓	✓	✓
Product Category 1	✓	✓	✓
Marital Status		✓	✓
Gender		✓	✓
Age		✓	✓
Occupation			✓
City Category			✓
Stay in Current City Years			✓

**Table 2** Model 1 contains three covariates, model 2 contains five covariates and model 3 contains seven covariates.

Covariate\Model	Model 1	Model 2	Model 3
Intercept	✓	✓	✓
Median income in block	✓	✓	✓
Median house age in block	✓	✓	✓
Average number of rooms		✓	✓
Average number of bedrooms		✓	✓
Block population			✓
Average house occupancy			✓