Determining the K in K-fold cross-validation

Kenichiro McAlinn[*] & Kōsaku Takanashi[†]

Abstract

Cross-validation is a standard technique used across science to test how well a model predicts new data. Data are split into K "folds," where one fold (i.e., hold-out set) is used to evaluate a model's predictive ability. Researchers typically rely on conventions when choosing K, commonly $K = 5$, or $80{:}20$ split, even though the choice of K can affect inference and model evaluation. Principally, this K should be determined by balancing the predictive accuracy (bias) and the uncertainty of this accuracy (variance), which forms a tradeoff based on the size of the hold-out set. More training data means more accurate models, but fewer testing data lead to uncertain evaluation, and vice versa. The challenge is that this evaluation uncertainty cannot be estimated directly from data. We propose a procedure to determine the optimal K by deriving a finite-sample upper bound on the evaluation uncertainty and adopting a utility-based approach to make this tradeoff explicit. Analyses of real-world datasets using linear regression and random forest demonstrate this procedure in practice, providing insight into implicit assumptions, robustness, and model performance. Critically, the results show that the optimal K depends on both the data and the model, and that conventional choices implicitly make assumptions about the fundamental characteristics of the data. Our framework makes these assumptions explicit and provides a principled, transparent way to select K based on the data and model rather than convention. By replacing a one-size-fits-all choice with context-specific reasoning, it enables more reliable comparisons of predictive performance across scientific domains.

Keywords: Cross-validation; Model evaluation; Prediction uncertainty; Bias–variance tradeoff

---

[*] Department of Statistical Science, Fox School of Business, Temple University, Philadelphia, PA. Email: kenichiro.mcalinn@temple.edu

[†] RIKEN Center for Advanced Intelligence Project Email: kosaku.takanashi@riken.jp

Introduction

In typical data analysis, the researcher has a plethora of models or hyperparameters to choose from. For example, a researcher often has access to a set of models that generate drastically different inferences or predictions, and the researcher must choose one of these models to make informed decisions. Which model to use– i.e., model uncertainty– is a central issue in statistics and machine learning. To assess model performance, a common technique used is cross-validation (CV), where a certain portion of the data is randomly allocated to train the model, and the remaining portion is used to evaluate the model through its predictions. Cross-validation, developed simultaneously and independently by Seymour Geisser and Mervyn Stone (Christensen and Johnson, 2007), is popular, and used across a broad breadth of fields (e.g., machine learning, econometrics, genomics, and epidemiology, to name just a few), because the concept and implementation are simple, while not being constrained to a specific data type or a domain, and theoretically known to produce unbiased estimates of the prediction loss (see, e.g., Efron, 1983; Shao, 1993; Bates et al., 2024). In simple terms, cross-validation tests how well a model trained on part of the data predicts the rest. Repeating this process multiple times provides an estimate of how the model will generalize to unseen data. The most common version is K-fold cross-validation, where the data are split into K equal-sized folds, and the test set is constructed by removing one of them. In practice, K is determined by convention (e.g., 5-fold CV, or 80:20 split), and does not typically differ across domain, dataset, or model. Even though this choice of K directly affects selection and inference, there is no principled way to determine this K with theoretical justification. Consider a concrete example: a researcher analyzing a dataset might find that Model A outperforms Model B using 5-fold CV, but Model B beats Model A using 10-fold CV. This could occur because Model B is more complex than Model A, requiring more data to produce good forecasts, but it could also occur because, particularly if the total sample size is small, the test data happened to favor Model B. In either case, the researcher could select the wrong model and reach incorrect scientific conclusions if the number of folds is poorly chosen. More importantly, a researcher faced with different results based on the number of folds does not have a principled way to discern which to use.

The difficulty in determining the hold-out size is due to the inherent tradeoff that exists in cross-validation. The more training data one has, the better the model estimation will be. Conversely, the more test data one has, the better the performance assessment of the model will be. Given a fixed total sample size, a tradeoff exists between the accuracy of the model and the uncertainty of the model evaluation. Consider an extreme case where all of the data are allocated to the training set. The model is the most accurate, using all the data to train, but there is no model evaluation (due to not having any test samples), and the uncertainty regarding the assessment is the largest. At the other extreme, if all data are allocated to the test set, the model cannot learn from the data and can only predict based on prior assumptions. The evaluation uncertainty, however, is the smallest because all of the data are used to evaluate. This tradeoff suggests that an optimal balance exists between having sufficient training data for learning and enough test data for reliable evaluation. This simple thought experiment illustrates why a universal choice of K cannot exist, since the optimal split depends on how much uncertainty one can tolerate in evaluating model accuracy.

Our goal is to formalize this tradeoff to enable a systematic choice of K. More precisely, this tradeoff is between the prediction loss of the model and its uncertainty regarding the prediction loss; similar to the bias-variance tradeoff regarding model complexity. While this tradeoff has been intuited, it has never been rigorously explicated due to the impossibility of a universal unbiased estimator for the latter uncertainty (Bengio and Grandvalet, 2004), except for specific problem settings (e.g., Arlot and Lerasle, 2016), and only asymptotic results have been derived for the purpose of testing model performance (Austern and Zhou, 2020; Bayle et al., 2020; Li, 2023). While these results provide valuable asymptotic insights, they do not yield a usable criterion for choosing K in a given dataset with finite samples. For our theoretical contribution, we derive a finite sample upper bound (the theoretical maximum) to this uncertainty, rigorously linking it to the prediction loss. Given this uncertainty bound, the optimal test set size can be derived by balancing this tradeoff. We express this tradeoff using a mean-variance utility (a combined score of accuracy and reliability) of the von Neumann-Morgenstern utility function. The optimal test set size maximizes this utility function, given the researcher's belief in the irreducible error in the data-generating process, which acts as a tuning parameter.

A key insight of our approach is recognizing that any choice of K implicitly assumes the level of irreducible noise in the data-generating process. Thus, choosing $K = 5$ assumes one noise level and $K = 10$ assumes another. Current practice makes these choices by convention without examining what they imply about the fundamental characteristics of the data. We make these implicit assumptions explicit, transforming convention into a transparent, data-informed decision.

In the following, we formalize this tradeoff, derive theoretical results for its components, and apply it to real data.

Definitions and main theoretical result

To derive our central theoretical results, we first define the notation, data-generating assumptions, and explicate cross-validation, including its key theoretical properties. For the data generating assumptions, we consider cases where the irreducible error is symmetric (e.g., Gaussian) and homoskedastic (equal variability across observations), and where it is asymmetric (e.g., Gamma) or heteroskedastic (unequal variability across observations). Given the definitions, we derive the theoretical upper bound of the predictive accuracy uncertainty for both assumptions.

Denote the training set sample size as n and the test set sample size as m. The total sample size is $n + m = N$. Given a fixed total sample size, an increase in n means a decrease in m, and vice versa. The central question is determining m that best balances model accuracy and evaluation certainty.

For K-fold cross-validation, the test set size is $m = N/K$, where N is divided into $\{C_1, ..., C_k, ..., C_K\}$ sets. The test set is C_k and the training set is denoted as $C_{-k} = \{C_1, ..., C_{k-1}, C_{k+1}, ..., C_K\}$ to represent the divided dataset minus the test set, C_k. The training set consists of $\{(Y_i, X_i)\}_{i=1}^{n}$ and the test set consists of $\{(Y_j, X_j)\}_{j=n+1}^{n+m}$. Throughout, we assume that the response follows the data-generating process

$$Y_i = f(X_i) + \varepsilon_i,$$

where the outcome is generated from some function of covariates, $f(X_i)$, and consider cases where ε_i is symmetric and homoskedastic, and where it is asymmetric or heteroskedastic. Together, this covers most problems encountered in data analysis.

For notational simplicity, we denote data and predictions on the test set as C_k and the training set as C_{-k}. We consider the squared error loss,

$$\begin{aligned}\hat{L}_k\left(\hat{\boldsymbol{\mu}}_k\right) &= \frac{1}{m}\left\|\hat{\boldsymbol{\mu}}_k - \boldsymbol{y}_k\right\|_{\mathbb{R}^m}^2 - \frac{1}{m}\left\|\boldsymbol{\varepsilon}\right\|_{\mathbb{R}^m}^2 \\ &= \frac{1}{m}\sum_{j\in C_k}\left\{\left(\hat{f}_{-k}\left(x_j\right) - y_j\right)^2 - \varepsilon_j^2\right\},\end{aligned}$$

where $\hat{\boldsymbol{\mu}}_k = \left[\hat{f}_{-k}\left(x_{n+1}\right), ..., \hat{f}_{-k}\left(x_{n+m}\right)\right]^\top$ is the prediction generated from the model trained on the training set, C_{-k}, to predict the test set, C_k, and $\boldsymbol{y}_k = \left[y_{n+1}, ..., y_{n+m}\right]^\top, \boldsymbol{\varepsilon}_k = \left[\varepsilon_{n+1}, \cdots, \varepsilon_{n+m}\right]^\top$. The K-fold cross-validation loss is

$$\hat{L}_k\left(\hat{\boldsymbol{\mu}}_k\right) = \frac{1}{K}\sum_{k=1}^{K}\left\{\frac{1}{m}\left\|\hat{\boldsymbol{\mu}}_k - \boldsymbol{y}_k\right\|_{\mathbb{R}^m}^2 - \frac{1}{m}\left\|\boldsymbol{\varepsilon}_k\right\|_{\mathbb{R}^m}^2\right\}.$$

In other words, these expressions quantify how close the model's predictions are to the true data-generating process, averaged over all test sets. The fundamental tradeoff in cross-validation is that as n increases, $\hat{f}_{-k}$ becomes more accurate (more learning), but the uncertainty of $\hat{L}_k\left(\hat{\boldsymbol{\mu}}_k\right)$ increases, due to a decrease in m.

Cross-validation is popular due to its ease of use (only requiring data splitting, training, and predicting), but also due to its theoretical properties, namely that it is an unbiased estimate of the prediction loss (and a conditional unbiased estimate of pure loss):

$$\mathbb{E}_Y\left[\frac{1}{m}\left\|\hat{\boldsymbol{\mu}}_k - \boldsymbol{y}_k\right\|_{\mathbb{R}^m}^2 - \frac{1}{m}\left\|\boldsymbol{\varepsilon}\right\|_{\mathbb{R}^m}^2 \middle| \boldsymbol{x}\right] = \frac{1}{m}\left\|\hat{\boldsymbol{\mu}}_k - \boldsymbol{\mu}_k\right\|_{\mathbb{R}^m}^2, \tag{1}$$

$$\mathbb{E}_Y\left[\frac{1}{K}\sum_{k=1}^{K}\left\{\frac{1}{m}\left\|\hat{\boldsymbol{\mu}}_k - \boldsymbol{y}_k\right\|_{\mathbb{R}^m}^2 - \frac{1}{m}\left\|\boldsymbol{\varepsilon}_k\right\|_{\mathbb{R}^m}^2\right\} \middle| \boldsymbol{X}\right] \approx \frac{1}{mK}\left\|\hat{\boldsymbol{\mu}}_k - \boldsymbol{\mu}_k\right\|_{\mathbb{R}^{mK}}^2, \tag{2}$$

for the basic cross-validation and K-fold cross-validation, respectively, where $\boldsymbol{\mu}_k = [f(x_{n+1}), ..., f(x_{n+m})]$ denotes the vector of true response functions, and $\sigma^2 = \mathbb{E}[\varepsilon^2]$ is the irreducible error. While this loss can be empirically estimated, except for the irreducible error, the variance of this loss,

$$\begin{aligned}\mathbb{V}\text{ar}\left[\hat{L}_m\left(\hat{\mu}_k\right)\right] = \mathbb{E}_Y\Bigg[\Bigg(&\frac{1}{m}\left\|\hat{\boldsymbol{\mu}}_k-\boldsymbol{y}_k\right\|^2_{\mathbb{R}^m}-\frac{1}{m}\left\|\boldsymbol{\varepsilon}\right\|^2_{\mathbb{R}^m}\\ &-\mathbb{E}_Y\left[\frac{1}{m}\left\|\hat{\boldsymbol{\mu}}_k-\boldsymbol{y}_k\right\|^2_{\mathbb{R}^m}-\frac{1}{m}\left\|\boldsymbol{\varepsilon}\right\|^2_{\mathbb{R}^m}\middle|\,\boldsymbol{x}\right]\Bigg)^2\,\Bigg|\,\boldsymbol{x}\Bigg]\end{aligned}$$

has been previously unknown, and the empirical variance of the loss is known to severely underestimate the variance due to overlapping samples. This makes the explicit expression of the variance, and thus the tradeoff, unknown.

Instead of determining the exact variance, we derive its upper bound. The upper bound serves as the upper limit on what the variance of the predictive accuracy can be, and is considered the worst-case scenario. For the symmetric, homoskedastic case, this upper bound is "tight," meaning that the bound is close to the unbiased variance.

We first provide the upper bound for the case in which the errors, ε_i, are symmetric and homoskedastic.

Theorem 1. *Assume that ε_i is symmetrically distributed and homoskedastic. Then, the variance of the loss is,*

$$\begin{aligned}&\mathbb{E}_Y\Bigg[\Bigg(\frac{1}{m}\left\|\hat{\boldsymbol{\mu}}_k-\boldsymbol{y}_k\right\|^2_{\mathbb{R}^m}-\frac{1}{m}\left\|\boldsymbol{\varepsilon}\right\|^2_{\mathbb{R}^m}\\ &\quad-\mathbb{E}_Y\left[\frac{1}{m}\left\|\hat{\boldsymbol{\mu}}_k-\boldsymbol{y}_k\right\|^2_{\mathbb{R}^m}-\frac{1}{m}\left\|\boldsymbol{\varepsilon}\right\|^2_{\mathbb{R}^m}\middle|\,\boldsymbol{x}\right]\Bigg)^2\,\Bigg|\,\boldsymbol{x}\Bigg]\\ &=4\frac{1}{m^2}C\left(2,m\right)\sigma^2\left(\sum_{j\in C_k}\left(\hat{f}_{-k}(x_j)-f(x_j)\right)^2\right)\\ &\leqq 4\frac{1}{m^2}\sigma^2\left(\sum_{j\in C_k}\left(\hat{f}_{-k}(x_j)-f(x_j)\right)^2\right)\end{aligned}\tag{3}$$

where

$$C\left(s,t\right)=\frac{1}{2^t}\frac{1}{t^{s/2}}\sum_{r=0}^{t}\left(\begin{array}{c}t\\r\end{array}\right)\left|t-2r\right|^s\leq\frac{2^{s/2}}{\sqrt{\pi}}\varGamma\left(\frac{s+1}{2}\right).$$

The right-hand side of (3) is what bounds the variance from above. Intuitively, Theorem 1 states that as the test-set size increases, the uncertainty in model evaluation shrinks, up to a calculable upper limit. Since the bound is on the constant $C\left(s,t\right)$, the bound is tight, and close to the unbiased estimator. The proof of this theorem is given in the Appendix.

Note that the upper bound depends on three variables: the model bias, $\frac{1}{m}\left(\sum_{j\in C_k}\left(\hat{f}_{-k}(x_j)-f(x_j)\right)^2\right)$, irreducible error, σ^2, and the hold-out size, m. Intuitively, this theorem tells us that when test samples are more numerous (larger m), the uncertainty in our estimate of predictive accuracy shrinks due to the extra $\frac{1}{m}$, though at the cost of model accuracy from fewer training samples.

The model bias is the difference between the prediction and the data minus irreducible error, and can be estimated from the empirical cross-validation loss, given the specification of the irreducible error.

In many real datasets, errors may not be symmetrically distributed (e.g., binary outcomes), or evenly distributed (e.g., income data vary more among high earners). Theorem 2 relaxes the earlier assumptions to cover such cases.

When ε is not symmetric, or when ε_j are independently distributed but heteroskedastic, we have the following upper bound:

Theorem 2. *Let* $\sigma_j^2 = \mathbb{E}\left[\varepsilon_j^2\right]$, *we have*

$$\begin{aligned}
&\mathbb{E}_Y\Bigg[\Bigg(\frac{1}{m}\left\|\hat{\boldsymbol{\mu}}_k - \boldsymbol{y}_k\right\|_{\mathbb{R}^m}^2 - \frac{1}{m}\left\|\boldsymbol{\varepsilon}\right\|_{\mathbb{R}^m}^2 \\
&\quad - \mathbb{E}_Y\left[\frac{1}{m}\left\|\hat{\boldsymbol{\mu}}_k - \boldsymbol{y}_k\right\|_{\mathbb{R}^m}^2 - \frac{1}{m}\left\|\boldsymbol{\varepsilon}\right\|_{\mathbb{R}^m}^2 \,\middle|\, \boldsymbol{x}\right]\Bigg)^2 \,\Bigg|\, \boldsymbol{x}\Bigg] \\
\leqq& 16\frac{1}{m^2}C\left(2,m\right)\left(\sum_{j\in C_k}\sigma_j^2\left(\hat{f}_{-k}(x_j) - f(x_j)\right)^2\right) \\
\leqq& 16\frac{1}{m^2}\left(\sum_{j\in C_k}\sigma_j^2\left(\hat{f}_{-k}(x_j) - f(x_j)\right)^2\right).
\end{aligned}$$

If the error is not symmetric, the bound worsens as the constant term goes from 4 to 16. If the error is heteroskedastic, σ changes to σ_j. Apart from those changes, the intuition is identical to Theorem 1. The proof of this theorem is given in the Appendix.

Together, the two theorems cover a broad set of problems, while the former provides an improved bound for a more specific case.

Determining K

With the loss and its upper bound variance, we have an explicit tradeoff. To determine the optimal point of tradeoff, we adopt the mean-variance utility within the von Neumann-Morgenstern expected utility framework:

$$\text{Utility}\left(n,m\right) = -\left(\underbrace{\mathbb{E}\left[\hat{L}_k\left(\hat{\mu}_k\right)\right]}_{\text{Loss}} + \underbrace{\mathbb{V}\text{ar}\left[\hat{L}_k\left(\hat{\mu}_k\right)\right]}_{\text{Risk}}\right). \tag{4}$$

Here, this ‘utility’ simply measures the overall quality of a split. A higher utility means a better balance between accuracy and uncertainty. While there are other forms of utility, we select this utility due to its simplicity and ease of interpretation, particularly how its analogous to the bias-variance tradeoff.

Based on the above, our proposed strategy for determining K involves four main steps:

Step 1: Estimate the model's empirical prediction loss for several training/test splits;

Step 2: Estimate how uncertain the losses in Step 1 are from the upper bound in Theorem 1 or 2;

Step 3: Combine both into a single 'utility' measure;

Step 4: Choose the K that maximizes utility (the best tradeoff).

To set up the above strategy, we first prepare the expected negative utility under mean-variance:

$$\begin{aligned} &\mathbb{E}_{X,Y}\left[\frac{1}{m}\left\|\hat{\boldsymbol{\mu}}_k - \boldsymbol{y}_k\right\|^2_{\mathbb{R}^m} - \frac{1}{m}\left\|\boldsymbol{\varepsilon}\right\|^2_{\mathbb{R}^m}\right] \\ &\quad + \mathbb{E}_Y\left[\left(\frac{1}{m}\left\|\hat{\boldsymbol{\mu}}_k - \boldsymbol{y}_k\right\|^2_{\mathbb{R}^m} - \frac{1}{m}\left\|\boldsymbol{\varepsilon}\right\|^2_{\mathbb{R}^m}\right.\right. \\ &\qquad\left.\left. - \mathbb{E}_Y\left[\frac{1}{m}\left\|\hat{\boldsymbol{\mu}}_k - \boldsymbol{y}_k\right\|^2_{\mathbb{R}^m} - \frac{1}{m}\left\|\boldsymbol{\varepsilon}\right\|^2_{\mathbb{R}^m}\middle|\,\boldsymbol{x}\right]\right)^2\middle|\,\boldsymbol{x}\right], \end{aligned}$$

where the first term is the loss and the second term is the uncertainty. To calculate the above, we first need to estimate the two terms as a function of m.

(Step 1) The first term (prediction loss) can be estimated empirically using the empirical cross-validation loss:

$$\mathbb{E}\left[\hat{L}_k\left(\hat{\mu}_k\right)\middle|\,\boldsymbol{x}\right] \sim \frac{1}{m}\left\|\hat{\boldsymbol{\mu}}_k - \boldsymbol{y}_k\right\|^2_{\mathbb{R}^m} - \sigma^2, \tag{5}$$

where σ^2 is the irreducible error. To determine this loss as a function of m, or find the "loss curve," we need the empirical loss at, at least, three points: $m = 0$, $m = N$, and $0 < m < N$. These reference points anchor the curve that relates test-set size to model performance, allowing us to interpolate how loss changes with m. While estimating the loss is possible when $m = 0$, this estimate requires model class specific calculations can be too involved (discussed later). Instead, we estimate the loss at $m = 1$, or leave-one-out cross-validation (LOOCV). Since we can assume that the drop-off in loss from $m = 0$ to $m = 1$ is minimal, the LOOCV works as a good stand-in. Estimating the loss when $m = N$ is also possible, but it is unrealistic that the optimal test size lies near $n = 0$, so instead we use some minimal number for the model to produce reasonable predictions (in our example, we use $K = 2$), which we call leave-most-out cross-validation (LMOCV). Finally, we calculate some $0 < m < N$, e.g., 5-fold CV. Each loss is denoted as $\hat{L}_{LOO}$, $\hat{L}_{LMO}$, and $\hat{L}_{5-fold}$, and sample size as m_{LOO}, m_{LMO}, and m_{5-fold}, respectively. Since the loss is monotone decreasing, the curve can be estimated as $E_{\hat{L}}(m) = m^{\frac{\log(\beta)}{\log(\alpha)}}(\hat{L}_{LMO} - \hat{L}_{LOO}) + \hat{L}_{LOO} - \sigma^2$, where $\beta = \frac{\hat{L}_{5-fold}-\hat{L}_{LOO}}{\hat{L}_{LMO}-\hat{L}_{LOO}}$ and $\alpha = \frac{m_{5-fold}-m_{LOO}}{m_{LMO}-m_{LOO}}$. While there are other ways to estimate this curve, the above empirically fits well and satisfies all constraints. From this, we have the loss curve. Note that the estimation of the loss curve can be avoided entirely if the researcher is only interested in finding the optimal K within a limited set of K (e.g., $K = 4, 5, 10, 20$).

(Step 2) Given the loss curve, $E_{\hat{L}}(m)$, the upper bound of the variance curve, $V_{\hat{L}}(m)$, is calculated as:

$$V_{\hat{L}}(m) = 4\frac{1}{m}\sigma^2 E_{\hat{L}}(m)$$

Again, σ^2 is the irreducible noise of the data. This formula shows how uncertainty decreases as test-set size increases, providing a way to quantify the tradeoff illustrated earlier.

(Step 3) Finally, the utility can be calculated as,

$$\text{Utility(m)} = -\left(E_{\hat{L}}(m) + V_{\hat{L}}(m)\right).$$

(Step 4) Given σ^2, the optimal test set size is what maximizes this utility (or minimizes $E_{\hat{L}}(m) + V_{\hat{L}}(m)$).

Making implicit assumptions explicit: the role of σ^2

A central element of our approach– and of any cross-validation procedure– is the irreducible noise of the data-generating process, σ^2. Every choice of K implicitly assumes some value of σ^2, and conventional approaches leave this assumption unexamined. For example, choosing $K = 5$ for a particular dataset and model implies a specific belief about this noise that differs from $K = 10$. Our framework makes this implicit assumption explicit and operational.

In practical terms, σ^2 reflects how noisy or unpredictable the data are. In physical sciences, σ^2 may be small (e.g., controlled experiments), whereas it can be large in social sciences. For example, in a typical biological experiment, where measurement variability is roughly 10% of total signal, a good estimate could be $\sigma^2 \approx 0.1$, while in a social survey, responses vary widely; $\sigma^2 \approx 1 - 2$. Since the estimation of σ^2 requires the knowledge of the true model, and thus makes cross-validation unnecessary, the optimal hold-out size should be viewed as conditional on the researcher's beliefs about the level of noise (e.g., high in social sciences, low in deterministic systems). While the ability to choose σ^2 may seem too subjective, it is important to note that the researcher is implicitly assuming some σ^2 whenever they choose K. Rather than viewing this as a limitation, this framing allows for the researcher to conduct a useful sensitivity analysis, or "reverse engineer" implicit noise levels under conventional hold-out splits. In other words, σ^2 is a "knob" that the researcher can "tune," allowing for the researcher to take an implicit assumption and relax it by exploring other assumptions. For example, by varying σ^2 over a plausible range, researchers can examine how the optimal test set changes, and identify robustness ranges where splits remain near-optimal (e.g., "for all $\sigma^2 \in [0.5, 2.0]$, 5-fold CV remains near-optimal"). Reverse engineering σ^2 allows the researcher to interpret the choice of K in terms of underlying assumptions about data quality. Both uses provide ways to understand σ^2 from a predictive perspective. Since the researcher is already assuming some σ^2 when they conduct K-fold CV, treating σ^2 as a tuning parameter allows the researcher to reveal their implicit assumptions and explore other assumptions.

A more holistic view is to examine how the optimal test size changes given a wide range of σ^2. This relation defines the Pareto frontier, which summarizes all optimal tradeoffs between accuracy and uncertainty for different assumed noise levels. The Pareto frontier is analogous to modern portfolio theory, where one specifies the risk aversion parameter, λ, to calculate the efficient frontier of the optimal portfolio (i.e., what is the optimal portfolio proportion given risk aversion). Thus, choosing σ^2 is similar to choosing between high-risk, high-return and low-risk, low-return investments. In this sense, choosing σ^2 can be viewed as the researcher's risk tolerance regarding the tradeoff. If the researcher prefers lower uncertainty around the prediction loss, they should choose a lower σ^2, and vice versa. Note that, when calculating the Pareto frontier, there is a point where σ^2 is too large and the frontier cuts off. This is an– albeit loose– upper bound on σ^2, calculated from the model and data. This provides some idea regarding what the most risk-averse σ^2 is, from which we can calculate the "safest" K for cross-validation.

For the heteroskedastic errors, σ^2 should be specified as $\sigma^2 = \max(\sigma_j^2)$. This is conservative, though necessary due to heteroskedasticity inducing further uncertainty.

Real-wold example: Abalone age prediction

To demonstrate our approach in practice, we apply it to a regression problem using the Abalone dataset (Nash et al., 1994). The Abalone dataset has $N = 4,177$ observations with eight covariates, and the goal is to predict the age of the abalone based on these features. We consider two models, a linear regression model and a random forest model, for comparison to estimate the optimal test set size, due to the latter being a more complex (nonlinear) model compared to the former.

Step 1: Estimate loss curve

First, we calculate the LOOCV squared error loss, $\hat{L}_{LOO}$, by removing one sample for testing ($m = 1$) and using the rest ($n = 4,176$) to train the models. This produces a loss value of 4.9394 for the linear model and 4.6379 for the random forest.

Second, we calculate the LMOCV squared error loss, $\hat{L}_{LMO}$, by removing half of the samples ($m = 2,088$) for testing and using the rest ($n = 2,089$) to train the models. This produces a loss value of 4.9594 for the linear model and 5.0571 for the random forest.

Third, we calculate the LMOCV squared error loss, $\hat{L}_{5-fold}$, by removing 1/5th of samples ($m = 835$) for testing and using the rest ($n = 3,342$) to train the models. This produces a loss value of 4.9426 for the linear model and 4.6692 for the random forest.

Plugging this into the loss curve function, we obtain $E_{\hat{L}}(m) = m^{2.0010}0.0200 + 4.9394 - \sigma^2$ for the linear model and $E_{\hat{L}}(m) = m^{2.7898}0.4192 + 4.6379 - \sigma^2$ for the random forest.

The estimated loss curve, with $\sigma^2 = 1$, is visualized in Figure 1, which shows how prediction loss changes with the size of the test set. The flatness of the curve for the linear model means its performance

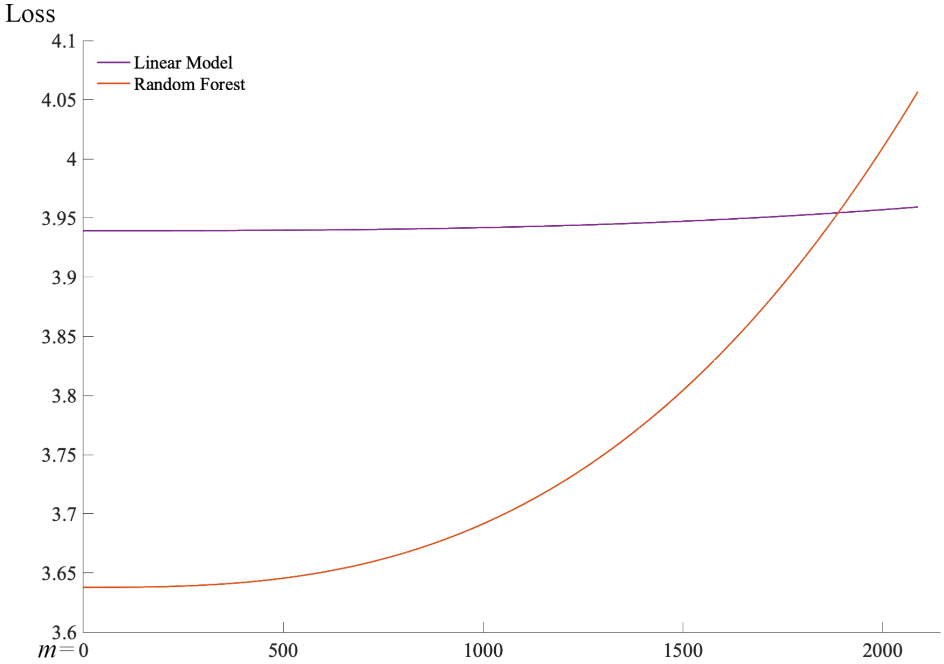

Figure 1: Abalone data: Estimated loss curve for the linear model and random forest. The y-axis is the estimated mean squared error under different hold-out data sizes (m: x-axis). σ^2 is the irreducible error, which is set to $\sigma^2 = 1$.

stabilizes quickly, while the steeper curve for the random forest shows that complex models need more training data.

Step 2: Calculate variance curve

After assessing how model accuracy changes, we next examine how the uncertainty of this evaluation behaves. Given the estimated loss curve in Step 1, we can directly calculate the upper bound of the variance curve as

$$V_{\hat{L}}(m) = 4\frac{1}{m}\sigma^2 E_{\hat{L}}(m).$$

The calculated variance curve, with $\sigma^2 = 1$, is visualized in Figure 2. The variance is extremely high when $m = 1$, because the evaluation is based on a single data point, and there is a large amount of uncertainty regarding whether this one sample is representative of the population. As m increases, the uncertainty decreases, as there are more data to assess the predictive ability of a model. Compared to the loss curve, the variance curve is less contrastive, though it follows a similar pattern to the loss curve in terms of the difference between the two models.

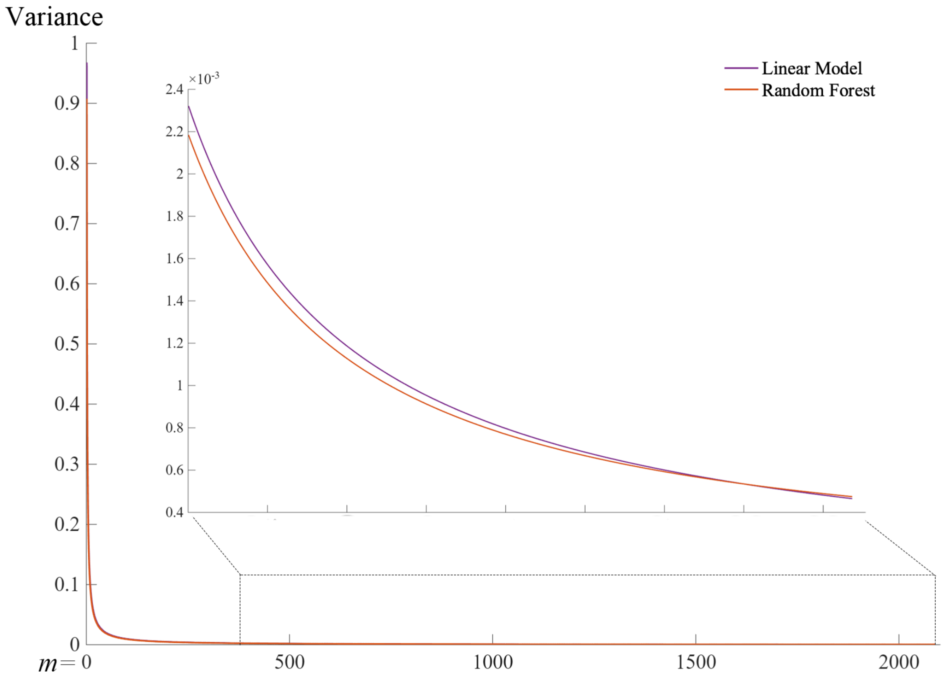

Figure 2: Abalone data: Estimated variance curve for the linear model and random forest. The y-axis is the calculated upper bound of the variance of the loss under different hold-out data sizes (m: x-axis). σ^2 is the irreducible error, which is set to $\sigma^2 = 1$.

Step 3: Calculate the mean-variance utility

Given both the loss curve and the variance curve, we can calculate the negative utility,

$$E_{\hat{L}}(m) + V_{\hat{L}}(m),$$

and whatever minimizes this value, given some value of σ^2, is the optimal test set size.

Figure 3 summarizes how the overall utility changes with test-set size under different noise assumptions ($\sigma^2 = \{0.01, 0.1, 1\}$). The lowest point on each curve marks the best choice of K, as it minimizes the negative utility (and maximizes the utility). For the linear model, the optimal test set ranges between roughly 200 and 950 samples; for the random forest, between 150 and 450. The optimal test set size is different between the two models, reflecting the difference in model complexity. In particular, the linear model is a faster learner, requiring fewer data for training and allocating more for testing. The optimal test size of the linear model also varies more than the random forest given the specification of σ^2. Notably, except for the linear model with $\sigma^2 = 1$, neither model has the optimal test size around $K = 5$. Given these results, a researcher using the random forest with this data should not be using $K = 5$ because the model has not learned enough for accurate predictions.

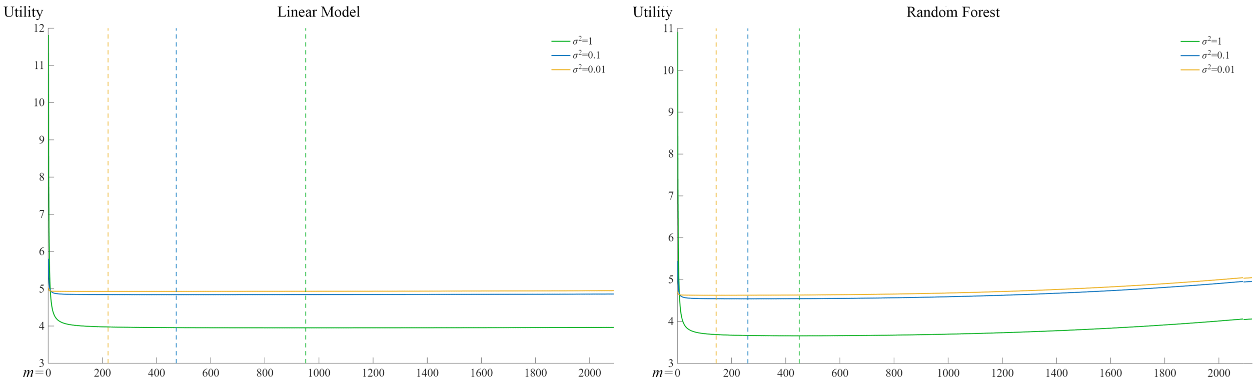

Figure 3: Abalone data: Negative utility of the linear model (left) and random forest (right), for $\sigma^2 = \{0.01, 0.1, 1\}$. Dashed lines are the minimal point under each σ^2.

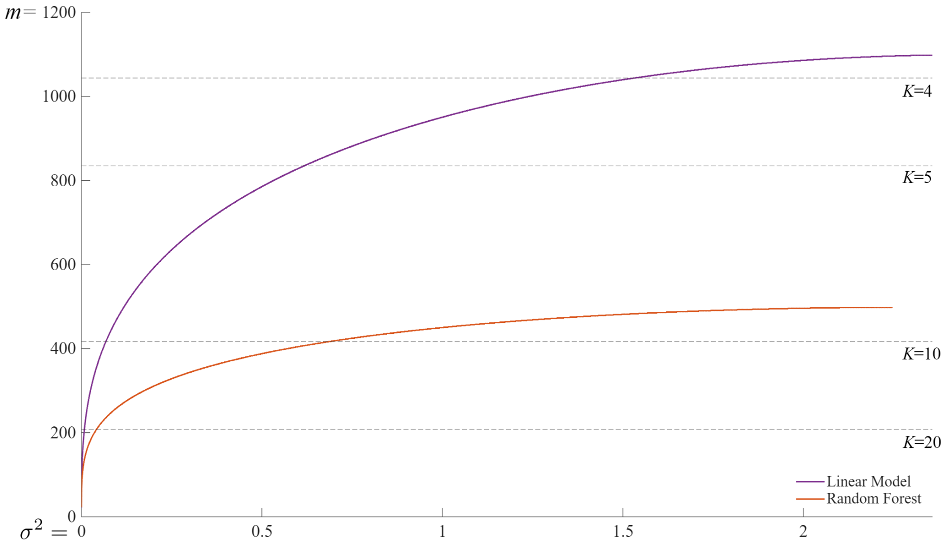

Figure 4: Abalone data: Pareto frontier for the linear model and random forest. Dashed lines denote sample size that equals $K = 4, 5, 10, 20$.

Step 4: Calculate the Pareto frontier

Once the utility is calculated, we can obtain the Pareto frontier by varying σ^2. This is done by calculating the minimal test set size under σ^2, while varying σ^2 until the minimal test set size peaks. The Pareto frontier for this data set is given in Figure 4.

The frontier maxes out at around $\sigma^2 = 2.25 \sim 2.5$, which is the loose upper bound of the irreducible noise. Comparing the two models, we see that the random forest has a much gradual curve than the linear model. This shows that, no matter what the belief of σ^2 is, the random forest requires a certain (large) amount of data for training. The linear model, with its steep curve, shows that, because the model is simple enough to require fewer data for training, the optimal sample size is sensitive to the specification of σ^2.

Table 1: Description of datasets used to compare optimal hold-out size. N is the total sample size and p is the number of covariates

	Description	Source	Domain	Target	N	p
`Abalone`:	Age of abalone	Nash et al. (1994)	Biology	Rings	4,177	8
`Concrete`:	Concrete compressive strength	Yeh (1998)	Materials Engineering	Concrete strength	1,030	8
`Servo`:	Simulation of a servo system	Ulrich (1986)	Engineering	Rise time	167	4
`Diabetes`:	Diabetes study	Efron et al. (2004)	Medicine	Disease progression score	442	10
`Income`:	Census income	Becker and Kohavi (1996)	Social Science	log(Income)	32,561	5

As a result, $K = 4, 5$ is never optimal for the random forest, but is for the linear model if we assume that $\sigma^2 = 0.7 \sim 1.6$. The linear model is a faster learner and requires fewer data to be "certain" compared to the more complex random forest. We can then calculate the implicit σ^2 under different K, or analyze how different models learn the data.

The key takeaway is that, even for a common dataset, the optimal K varies between models. Using a conventional split for both models, thus, can mislead model evaluation if the implicit assumption of σ^2 is drastically different.

Comparison with different datasets

To further illustrate how the optimal test size differs by model and dataset, we expand the analysis to five benchmark datasets (including the Abalone data in the previous section) that span multiple domains. The descriptions of datasets are given in Table 1. The datasets span diverse domains, from engineering, biology, to social science, with total sample sizes from 167 to over 32,561. This range enables us to compare how optimal test-set sizes differ when data noise levels vary across fields.

Table 2 summarizes the optimal test-set sizes across five datasets, two models, and three noise levels. Overall, the observations in the Abalone application are consistent for the other datasets: the optimal test set size increases as σ^2 increases, and the linear model requires fewer training data than the random forest (because the linear model is a faster learner). The exception to this observation is with the `Diabetes` dataset, where the random forest requires more test data than the linear model (for $\sigma^2 = 0.01$).

Note that there were instances where the empirical loss was less than $\sigma^2 = 1$, denoted by "-" (for `Diabetes`, RF, and `Income`, LM and RF). These entries provide valuable diagnostic information that the data are telling us that $\sigma^2 = 1$ is inconsistent with the observed model performance. If a researcher truly believed $\sigma^2 \geq 1$ for these datasets, they should question whether their models are overfitting or whether their noise assumptions are incorrect. For valid analysis, one should only consider $\sigma^2 < 1$. This diagnostic capability is precisely what conventional K selection obscures.

An important result is that, for all datasets and models considered, the optimal test-set size is neither leave-one-out (LOOCV), nor 2-fold. In most cases, the optimal K is in the range of $4 - 20$, depending on the model and assumption on σ^2. This is somewhat in line with conventional wisdom, though the variability given the data, model, and σ^2 is quite stark. The results show that the best K varies widely depending on the dataset and model, ranging roughly from 4 to 20 folds. This variability highlights why a fixed choice (e.g.,

Table 2: The optimal hold-out size for five datasets, for both linear model (LM) and random forest (RF), under varying noise levels, $\sigma^2 = \{0.01, 0.1, 1\}$. Entries with "-" indicate that the estimated loss was smaller than the irreducible noise.

	N	Model	σ^2= 0.01	0.1	1
`Abalone`	4,177	LM	221	473	951
		RF	143	260	450
`Concrete`	1,030	LM	44	114	299
		RF	17	34	69
`Servo`	167	LM	6	16	21
		RF	5	12	-
`Diabetes`	442	LM	16	46	133
		RF	21	38	71
`Income`	32,561	LM	1,675	4,165	-
		RF	1,412	2,053	-

5-fold CV) can be misleading. In practical terms, this means researchers should test several K values rather than assume one default, especially when the dataset size or expected noise level differs across studies.

Finally, we reverse engineer to estimate the implicit σ^2, given a specific K fold. The results, with $K = \{4, 5, 10, 20\}$, is reported in 3. Overall, the results are in line with earlier results: the implied σ^2 varies amongst models and data, and for each K. Within K, the implicit σ^2 for both models is somewhat consistent, particularly for large K. However, the implicit σ^2 drastically differs with differing K. Furthermore, for the random forest, the implicit σ^2 does not exist for $K = 4, 5$ (except for the `Diabetes` dataset), and does not exist for $K = 10$ for the `Servo` and `Income` datasets. This means that, if a researcher chooses those K for those datasets, there does not exist an implied σ^2, and thus the optimal hold-out size does not exist. From a researcher's perspective, this narrows down the potential K in a data-driven manner. Reverse engineering σ^2, thus, greatly aids in determining K by removing some K from the candidates. Given a smaller set of K, the researcher can select a K that best fits their assumption regarding the noise. These findings reinforce that optimal K is context-specific and that reverse-engineering σ^2 can guide researchers toward data-consistent choices.

Computational considerations

While conceptually straightforward, the proposed method can be computationally demanding because it requires multiple rounds of model fitting. At a minimum, cross-validation has to be performed three times, which can be too prohibitive for more complex machine learning methods, such as neural networks. To decrease computation cost, one can avoid full cross-validation to approximate the prediction loss. For example, a 5-fold CV can be assessed with only one test set, rather than the full five. Although there are some accuracy concerns with this approximation, it is fairly robust as long as the test set is not wildly inconsistent with the training set.

Table 3: The implicit σ^2 for five datasets, for both linear model (LM) and random forest (RF), under varying K, $K = \{4, 5, 10, 20\}$. Entries with "-" indicate that the implied σ^2, under that K, is greater than the upper bound of σ^2, i.e., these K values are too small for the model to learn effectively from these data.

	Model	$K = 4$	5	10	20
`Abalone`	LM	1.5284	0.6160	0.0683	0.0084
	RF	-	-	0.6847	0.0418
`Concrete`	LM	0.6960	0.4054	0.0770	0.0149
	RF	-	-	4.1242	0.3783
`Servo`	LM	-	-	0.0908	0.0114
	RF	-	-	-	0.0267
`Diabetes`	LM	0.6732	0.4061	0.0898	0.0196
	RF	5.3339	2.1891	0.1602	0.0115
`Income`	LM	-	-	0.0493	0.0094
	RF	-	-	-	0.0225

Modern parallel computing alleviates many of these concerns, but some large-scale machine learning methods might still find this computationally prohibitive. In some cases, one can use the theoretical convergence rate of the model as a stand-in. The difficulty, however, is in deriving these rates for all of the models considered. Another way is to use an approximate model that is less computationally intensive but retains the core characteristics.

The most computationally intensive cross-validation is the LOOCV. As mentioned above, we can calculate the Stein's unbiased risk estimator (SURE) when $m = 0$. Since $\hat{\boldsymbol{\mu}}$ is generally a function of $\boldsymbol{y}$, $\hat{\boldsymbol{\mu}}(\boldsymbol{y})$, from Stein's equality, we have

$$\mathbb{E}\left[\boldsymbol{\varepsilon}^\top \hat{\boldsymbol{\mu}}(\boldsymbol{x}) \middle| \boldsymbol{x}\right] = \mathbb{E}\left[\sigma^2 \sum_{i=1}^{n+m} \frac{\partial \hat{\mu}}{\partial y_i}(\boldsymbol{x}_i) \middle| \boldsymbol{X}\right].$$

Then, the SURE is,

$$\frac{1}{n}\widehat{\text{SURE}} = \frac{1}{n}\|\hat{\boldsymbol{\mu}} - \boldsymbol{y}\|^2_{\mathbb{R}^{n+m}} + 2\sigma^2 \frac{1}{n+m} \sum_{i=1}^{n+m} \frac{\partial \hat{\mu}}{\partial y_i}(\boldsymbol{x}_i). \tag{6}$$

This formula provides an analytic shortcut to estimate prediction risk without retraining the model for every possible test split. While div can be calculated for simpler models (e.g., linear models), it is quite prohibitive for more complex models, such as neural networks. Whether one uses LOOCV or SURE depends on the computational cost and mathematical availability.

Discussion

The K-fold cross-validation procedure can be viewed as a balance between two competing goals: model accuracy and evaluation certainty. Based on this tradeoff, we devise a strategy based on multi-objective optimization to determine the optimal K in K-fold cross-validation. Our proposed strategy is easily implemented through basic cross-validation estimates and is optimized based on the researcher's belief in the irreducible noise. Through regression exercises, we demonstrate how the optimal test set size can be estimated, and how that informs each model's predictive ability, sensitivity to assumptions regarding the irreducible error, and how conventional splits can mislead evaluation. Our proposed method can be applied to many statistical problems, but is particularly useful (due to a tighter bound) for regression problems where the errors are symmetrically distributed and homoskedastic.

For applied scientists, the main takeaway is conceptual: there is no universal best K. Instead, K should be chosen based on data quality and model complexity, similar to how one adjusts sample size or significance levels in experimental design.

The fundamental contribution of this work is not eliminating the need for assumptions in cross-validation, which is impossible, but making existing hidden assumptions explicit and actionable. Every choice of K in conventional practice implicitly embeds beliefs about the irreducible noise, model learning rates, and acceptable uncertainty levels. Rather than relying on these implicit assumptions, our framework provides tools to align cross-validation design with domain knowledge and data characteristics. This not only allows researchers to explicate and test their assumptions, but also improves transparency in scientific research.

Our approach has several limitations. The most important thing is the computational cost of estimating the cross-validation loss. While we have discussed reducing this cost by reducing the number of test sets in cross-validation, another approach is to use the convergence rate of each model, though this requires knowledge of this rate, which can be difficult to obtain. As discussed above, whether one reduces the number of test sets, uses the convergence rate, or SURE to avoid LOOCV, depends on the availability of computational resources and theoretical understanding.

Finally, as seen in the results of our analysis, it is not always necessary to obtain exact optimal test set sizes. Our results show that, for the Abalone data, there is a reasonable range for which little utility is lost by deviating from the optimal size. The key result is that the optimal size will rarely be LOOCV, or something like a 2-fold CV, but somewhere in between. Instead of relying on a single conventional K across models and datasets, we recommend identifying a reasonable range of K values tailored to specific problem types or data domains. This is similar to having different significance levels for statistical testing for different domains. The main takeaway is that there is no one-size-fits-all for cross-validation, and it should be considered on a domain-by-domain, dataset-by-dataset, and model-by-model basis.

As a general heuristic, our results indicate that K should be large if the researcher believes that the irreducible noise is small (e.g., physics compared to economics), or the model is complex and requires more data for training. Conversely, K can be large, which has the benefit of reducing computation, if the irreducible noise is large or the model is simple. The difficulty lies in the middle ground, e.g., when the noise is large but the model is complex, or when both the noise and model are moderate. For these cases, the optimal K should be calculated, or, at the very least, a reasonable range of K should be explored (e.g., $K = \{5, 10\}$).

Future work could explore automatic or adaptive methods that estimate the optimal K directly from data, making cross-validation more user-friendly for non-specialist researchers.

More broadly, this framework invites researchers across disciplines to think of cross-validation not as a fixed recipe, but as a design decision tuned to their data and scientific goals.

References

Arlot, S., Lerasle, M., 2016. Choice of v for v-fold cross-validation in least-squares density estimation. Journal of Machine Learning Research 17, 1–50.

Austern, M., Zhou, W., 2020. Asymptotics of cross-validation. arXiv preprint arXiv:2001.11111 .

Bates, S., Hastie, T., Tibshirani, R., 2024. Cross-validation: what does it estimate and how well does it do it? Journal of the American Statistical Association 119, 1434–1445.

Bayle, P., Bayle, A., Janson, L., Mackey, L., 2020. Cross-validation confidence intervals for test error. Advances in Neural Information Processing Systems 33, 16339–16350.

Becker, B., Kohavi, R., 1996. Adult. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5XW20.

Bengio, Y., Grandvalet, Y., 2004. No unbiased estimator of the variance of k-fold cross-validation. Journal of machine learning research 5, 1089–1105.

Christensen, R., Johnson, W., 2007. A conversation with seymour geisser .

Efron, B., 1983. Estimating the error rate of a prediction rule: improvement on cross-validation. Journal of the American statistical association 78, 316–331.

Efron, B., Hastie, T., Johnstone, I., Tibshirani, R., 2004. Least angle regression. The Annals of Statistics 32, 407–499.

Li, J., 2023. Asymptotics of k-fold cross validation. Journal of Artificial Intelligence Research 78, 491–526.

Nash, W., Sellers, T., Talbot, S., Cawthorn, A., Ford, W., 1994. Abalone. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C55C7W.

Shao, J., 1993. Linear model selection by cross-validation. Journal of the American statistical Association 88, 486–494.

Ulrich, K., 1986. Servo. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5Q30F.

Whittle, P., 1960. Bounds for the moments of linear and quadratic forms in independent variables. Theory of Probability & Its Applications 5, 302–305.

Yeh, I.C., 1998. Concrete Compressive Strength. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5PK67.

Appendix

Proof of Theorem 1

Proof. First, we introduce the following theorem:

Theorem 3. *(Theorem 1. of Whittle (1960)) Let the random variables* $z_1, z_2, \ldots, z_n$ *be statistically independent, and each assume the values* ± 1 *with respective probabilities* $\frac{1}{2}$ *. Then, for any set of real coefficients* b_j, $s \geq 2$,

$$\mathbb{E}\left[\left|\sum_{j}^{n} b_j z_j\right|^{s}\right] \leq C\left(s, n\right)\left(\sum_{j} b_j^2\right)^{s/2} \tag{7}$$

where $C\left(s.n\right) = \frac{1}{2^n}\frac{1}{n^{s/2}}\sum_{k=0}^{n}\begin{pmatrix} n \\ k \end{pmatrix}\left|n-2k\right|^{s}$. *The equality sign holds in (7) if* $s = 2$ *or if all the* b_j *have equal modulus, but not otherwise.*

From direct calculation, we have

$$\begin{aligned} &\left(\frac{1}{m}\left\|\hat{\boldsymbol{\mu}}_k - \boldsymbol{y}_k\right\|_{\mathbb{R}^m}^2 - \frac{1}{m}\left\|\boldsymbol{\varepsilon}\right\|_{\mathbb{R}^m}^2\right. \\ &\left.- \mathbb{E}_Y\left[\frac{1}{m}\left\|\hat{\boldsymbol{\mu}}_k - \boldsymbol{y}_k\right\|_{\mathbb{R}^m}^2 - \frac{1}{m}\left\|\boldsymbol{\varepsilon}\right\|_{\mathbb{R}^m}^2 \middle| \boldsymbol{x}\right]\right)^2 \\ = &\left(\frac{1}{m}2\boldsymbol{\varepsilon}^\top\left(\hat{\boldsymbol{\mu}}_k - \boldsymbol{\mu}_k\right)\right)^2 \end{aligned}$$

We now aim to obtain, $\mathbb{E}\left[\left(\frac{1}{m}2\boldsymbol{\varepsilon}_k^\top\left(\hat{\boldsymbol{\mu}}_k - \boldsymbol{\mu}_k\right)\right)^2\middle|\boldsymbol{x}\right]$. Within the $\boldsymbol{x}$-given expectation operator, the $\left(\hat{\boldsymbol{\mu}}_k - \boldsymbol{\mu}_k\right)$ term can be treated as a constant. Let $\mathbb{E}_{\pm}\left[\cdot\right]$ denote an expectation with respect to the sign of the ε. Since ε is symmetric, we have

$$\begin{aligned} &\mathbb{E}\left[\left|\sum_{j=1}^{m}\varepsilon_j\left(\hat{\mu}_j - \mu_j\right)\right|^2\middle|\boldsymbol{x}\right] \\ &= \mathbb{E}\left[\mathbb{E}_{\pm}\left[\left|\sum_{j=1}^{m}\text{sgn}\left(\varepsilon_j\right)\left|\varepsilon_j\right|\left(\hat{\mu}_j - \mu_j\right)\right|^2\middle|\boldsymbol{x}\right]\middle|\boldsymbol{x}\right]. \end{aligned}$$

and from (7), we have

$$\begin{aligned}
&\mathbb{E}\left[\mathbb{E}_{\pm}\left[\left|\sum_{j=1}^{m}\text{sgn}\left(\varepsilon_j\right)\left|\varepsilon_j\right|\left(\hat{\mu}_j-\mu_j\right)\right|^2\middle|\boldsymbol{x}\right]\middle|\boldsymbol{x}\right]\\
=&\mathbb{E}\left[C\left(2,m\right)\left|\sum_{j=1}^{m}\left|\varepsilon_j\right|^2\left(\hat{\mu}_j-\mu_j\right)^2\right|\middle|\boldsymbol{x}\right]\\
=&C\left(2,m\right)\sigma^2\left(\sum_{j\in C_k}\left(\hat{f}_{-k}(x_j)-f(x_j)\right)^2\right).
\end{aligned}$$

Further, since $C\left(2,m\right)$ is monotone increasing regarding m, we have

$$C\left(2,m\right)\leq\lim_{m\to\infty}C\left(2,m\right)=\frac{2}{\sqrt{\pi}}e^{-x^2}x^2=\frac{2}{\sqrt{\pi}}\varGamma\left(\frac{3}{2}\right)=1.$$

□

Proof of Theorem 2

Proof. When ε is not symmetric, let $\tilde{\varepsilon}_j$ be the copy of ε_j (sampled from the same distribution, independently from ε_j). Then from the symmetry of $\varepsilon_j-\tilde{\varepsilon}_j$, we have

$$\mathbb{E}\left[\left|\sum_{j=1}^{m}\varepsilon_j\left(\hat{\mu}_j-\mu_j\right)\right|^2\middle|\boldsymbol{x}\right]\leq\mathbb{E}\left[\left|\sum_{j=1}^{m}\left(\varepsilon_j-\tilde{\varepsilon}_j\right)\left(\hat{\mu}_j-\mu_j\right)\right|^2\middle|\boldsymbol{x}\right].$$

As with the case when ε_j is symmetric, from (7), we have

$$\begin{aligned}
&\mathbb{E}\left[\mathbb{E}_{\pm}\left[\left|\sum_{j=1}^{m}\text{sgn}\left(\varepsilon_j-\tilde{\varepsilon}_j\right)\left|\varepsilon_j-\tilde{\varepsilon}_j\right|\left(\hat{\mu}_j-\mu_j\right)\right|^2\middle|\boldsymbol{x}\right]\middle|\boldsymbol{x}\right]\\
=&\mathbb{E}\left[C\left(2,m\right)\left|\sum_{j=1}^{m}\left|\varepsilon_j-\tilde{\varepsilon}_j\right|^2\left(\hat{\mu}_j-\mu_j\right)^2\right|\middle|\boldsymbol{x}\right].
\end{aligned}$$

Further, from the Minkowski inequality,

$$\mathbb{E}\left[\left|\varepsilon_j-\tilde{\varepsilon}_j\right|^2\right]\underset{\text{Minkowski}}{\leq}\left(\mathbb{E}\left[\left|\varepsilon_j\right|^2\right]^{1/2}+\mathbb{E}\left[\left|\tilde{\varepsilon}_j\right|^2\right]^{1/2}\right)^2=\left(2\sigma_j\right)^2.$$

Therefore,

$$\begin{aligned}&\mathbb{E}\left[C\left(2,m\right)\left|\sum_{j=1}^{m}\left|\varepsilon_j-\tilde{\varepsilon}_j\right|^2\left(\hat{\mu}_j-\mu_j\right)^2\right|\middle|\boldsymbol{x}\right]\\ \leq &4C\left(2,m\right)\mathbb{E}\left[\left|\sum_{j=1}^{m}\sigma_j^2\left(\hat{\mu}_j-\mu_j\right)^2\right|\middle|\boldsymbol{x}\right].\end{aligned}$$

□