Optimization Letters (2023) 17:545–560
https://doi.org/10.1007/s11590-022-01894-6

ORIGINAL PAPER

The leave-worst-*k*-out criterion for cross validation

Lizhi Wang[1]

Received: 8 February 2021 / Accepted: 18 May 2022 / Published online: 17 June 2022

Abstract

Cross validation is widely used to assess the performance of prediction models for unseen data. Leave-*k*-out and *m*-fold are among the most popular cross validation criteria, which have complementary strengths and limitations. Leave-*k*-out (with leave-1-out being the most common special case) is exhaustive and more reliable but computationally prohibitive when $k > 2$; whereas *m*-fold is much more tractable at the cost of uncertain performance due to non-exhaustive random sampling. We propose a new cross validation criterion, leave-worst-*k*-out, which attempts to combine the strengths and avoid limitations of leave-*k*-out and *m*-fold. The leave-worst-*k*-out criterion is defined as the largest validation error out of C_{n^k} possible ways to partition *n* data points into a subset of $(n - k)$ for training a prediction model and the remaining *k* for validation. In contrast, the leave-*k*-out criterion takes the average of the C_{n^k} validation errors from the aforementioned partitions, and *m*-fold samples *m* random (but non-independent) such validation errors. We prove that, for the special case of multiple linear regression model under the $\mathcal{L}_1$ norm, the leave-worst-*k*-out criterion can be computed by solving a mixed integer linear program. We also present a random sampling algorithm for approximately computing the criterion for general prediction models under general norms. Results of two computational experiments suggested that the leave-worst-*k*-out criterion clearly outperformed leave-*k*-out and *m*-fold in assessing the generalizability of prediction models; moreover, leave-worst-*k*-out can be approximately computed using the random sampling algorithm almost as efficiently as leave-1-out and *m*-fold, and the effectiveness of the approximated criterion may be as high as, or even higher than, the exactly computed criterion.

✉ Lizhi Wang
lzwang@iastate.edu

[1] Department of Industrial and Manufacturing Systems Engineering, Iowa State University, Ames, IA 50011, USA

Keywords Cross validation · Leave-k-out · m-fold · Multiple linear regression · Bilevel optimization

1 Introduction

Cross validation is a widely used technique to estimate how the performance of a prediction model on a given in-sample dataset is generalizable to an unseen out-of-sample dataset. Take multiple linear regression for example, which predicts the response of a given explanatory data $x \in \mathbb{R}^{n\times p}$ with a linear function: $\hat{y} = x\beta$. For a given in-sample dataset of $(x^{\text{in}}, y^{\text{in}})$, the parameter β can be trained to fit the in-sample data with minimal distance between actual and predicted responses: $\beta^* \in \arg\min_{\beta} \{||y^{\text{in}} - x^{\text{in}}\beta||\}$. However, the model $\hat{y} = x\beta^*$ is not expected to fit an unseen out-of-sample dataset $(x^{\text{out}}, y^{\text{out}})$ as well as it did the in-sample dataset that it was trained to fit, or overfit [1, 2]. The essence of overfitting is to mistakenly extract noises as representative underlying model structure [3]. This phenomenon is particularly common [4] when the number of explanatory variables, p, far exceeds the number of data points, n. Bartlett et al. [5] analyzed the mysterious phenomenon of benign overfitting, which allows deep neural networks to achieve good prediction accuracy despite a perfect fit to noisy training data. They found that, for linear regression, overfitting cannot be benign unless the unimportant directions in parameter space significantly outnumber the sample size. When this characterization is not satisfied, overfitting often leads to large out-of-sample prediction errors. As a remedy to this challenge, cross validation can be used to estimate the extent of overfitting and to select prediction models that are less overfitted [1, 6, 7].

In order to estimate the performance of a prediction model in out-of-sample data, a common practice in cross validation is to partition the given in-sample data into training and validation subsets, train the model with the training data, calculate the prediction error using the validation data, and use the validation error as an estimation of the out-of-sample performance [8].

Cross validation criteria are either exhaustive or non-exhaustive. Exhaustive criteria evaluate the performance of a prediction model on all possible partitions of training and validation datasets subject to cardinality constraints. As perhaps the most popular exhaustive cross validation criterion, leave-k-out (LKO) consists of three steps. Step 1, enumerate all possible partitions to divide a given in-sample of n data points into a training subset with $n - k$ data points and a validation subset with k. Step 2, for each partition, train the prediction model and calculate its corresponding validation error. Step 3, use the average validation error as the LKO criterion [9, 10]. Due to the computational complexity, only the special case of leave-1-out is widely used [8, 11, 12]. Commonly used non-exhaustive cross validation criteria include m-fold [13, 14], holdout [15], random sampling [16], etc. It is worth

mentioning that leave-k-out is the exhaustive counterpart of (n/k)-fold when n is an integer multiple of k. As a special case, leave-1-out and n-fold are equivalent.

Exhaustive and non-exhaustive cross validation criteria have complementary strengths and limitations. Exhaustive criteria, on the one hand, take all possible partitions of training and validation subsets into consideration, at the cost of computational intractability. Non-exhaustive criteria, on the other hand, introduce the tradeoff between solution quality and computational complexity by sampling random partitions.

Recent studies in the literature have compared different cross validation methods, proposed new variants, and applied these methods in different application fields. Magnusson et al. [17] proposed a new method for model comparison by combining fast approximate leave-1-out surrogates with exact leave-1-out sub-sampling. Xu et al. [18] proposed to improve the effectiveness of m-fold by splitting the training dataset before applying m-fold to each split and pooling the prediction errors. Jung [19] suggested a variant of m-fold, which uses $m-1$ folds of data for model validation and the remaining fold for model construction. Ramezan [20] compared three cross validation methods (leave-1-out, m-fold, and Monte Carlo) using high spatial resolution remotely sensed datasets and found "minimal differences in accuracy for the different cross-validation tuning methods." Duarte [21] found cross validation to be more effective than internal metrics for tuning SVM hyperparameters. Recent applications of cross validation included detecting Alzheimer disease [22], predicting basketball game outcomes [23], cryo-EM map reconstruction [24], and wind energy predition [25].

The proposed leave-worst-k-out (LWKO) criterion can be defined as a variant of LKO. With the first two steps of the aforementioned LKO definition being the same, LWKO uses the largest one (as opposed to the average) of the C_n^k validation errors in the third step. As such, LWKO has two advantages over the LKO or m-fold criteria. First, LWKO is less intractable than LKO. To compute LKO, all the C_n^k validation errors need to be calculated, whereas LWKO only requires the identification of the largest error without having to calculate the exact values of the others. In Sect. 2.2, we present a random sampling algorithm for calculating LWKO approximately, which has a lower yet comparable computational efficiency with leave-1-out and m-fold. In Sect. 2.3, for the special case of using multiple linear regression as the prediction model under $\mathcal{L}_1$ norm, we present a mixed integer linear programming (MILP) formulation for computing the exact value of the LWKO criterion. Second, LWKO has been found to be more effective than LKO and m-fold in identifying outliers to the prediction model, which could be caused by underfitted or overfitted models. The LKO and m-fold criteria favor models that fit the validation data well on average, allowing a large number of good fits to compensate for the bad performance of a few outliers. In contrast, the LWKO criterion favors models whose worst outliers are not too outlying, even though the average fit may not be the best possible.

2 Methods

2.1 Definition of LWKO as a bilevel optimization model

The LWKO criterion is defined as the largest possible validation error when the model is trained using $n-k$ data points and validated using the remaining k data points. Since there are C_{n^k} possible such partitions of training and validation subsets, a brute force algorithm for calculating LWKO or LWO would require training the prediction model and calculating its validation error C_{n^k} times. In this section, we present the following bilevel optimization model (1)–(6), whose optimal solution yields the LWKO criterion:

$$\max_{w,\beta,\mathcal{T},\mathcal{V}} ||y_{\mathcal{V}} - h(x_{\mathcal{V}}|\beta)||_{\mathcal{L}} \tag{1}$$

$$\text{s.t.} \sum_j w_j = n-k \tag{2}$$

$$w \in \mathbb{B}^n \tag{3}$$

$$\mathcal{T} = \{j : w_j = 1\} \tag{4}$$

$$\mathcal{V} = \{j : w_j = 0\} \tag{5}$$

$$\beta \in \arg\min_b ||y_{\mathcal{T}} - h(x_{\mathcal{T}}|b)||_{\mathcal{L}}. \tag{6}$$

Here, the upper level objective function (1) is to maximize the validation error under norm $\mathcal{L}$. Function $h(x_{\mathcal{V}}|\beta)$ predicts the response from validation data $x_{\mathcal{V}}$ using parameter β. Variable w indicates the partition decision: $w_i = 1$ or $w_i = 0$ means that data point i is used in the training set, $\mathcal{T}$, or the validation set, $\mathcal{V}$, respectively. Constraints (2) and (3) require w to be a binary vector with exactly $n-k$ elements being 1 and k being 0. Constraints (4) and (5) define the training set, $\mathcal{T}$, and validation sets, $\mathcal{V}$, respectively. The lower level (6) finds the optimal parameter β to minimize the training error under norm $\mathcal{L}$.

This bilevel optimization model (1)–(6) suggests three underlying approaches to computing the LWKO criterion, exactly or approximately. First, train the prediction model for all C_n^k partitions in a brute force manner, compute their corresponding validation errors, and then use the largest one as LWKO. Second, train the prediction model with a random sample of partitions and use the largest validation error to approximate LWKO. Third, take advantage of special (such as linear) properties of function $h(x|\beta)$ to identify one training-validation dataset partition that results in the largest validation error, without having to compute the exact values of all other validation errors. The latter two approaches will be discussed in more details in the following two subsections.

2.2 Random sampling algorithm for computing LWKO approximately

We present a random sampling algorithm for calculating LWKO approximately, which is applicable to a general prediction function $h(x|\beta)$ under a general norm $\mathcal{L}$. This algorithm trains the prediction model for a randomly sampled subset of partitions and uses the largest validation error among the samples as an underestimated approximate value of LWKO. The premise of this algorithm is that approximate values of LWKO may also be informative for model selection when all prediction models are cross validated by the same approximate algorithm.

Algorithm 1 Random sampling algorithm for LWKO

Input: (1) explanatory data $x \in \mathbb{R}^{n\times p}$, (2) response data $y \in \mathbb{R}^n$, (3) prediction function $h(x|\beta)$, (4) norm $\mathcal{L}$, (5) integer parameter $1 < k < n$, and (6) integer parameter $s \geq 1$.

Output: approximate LWKO value r^*.

for $i \in \{1, 2, ..., s\}$ **do**
  Randomly partition indices $\{1, ..., n\}$ into $\mathcal{T}$ and $\mathcal{V}$ with $|\mathcal{T}| = n - k$ and $|\mathcal{V}| = k$.
  Compute $\beta_i \in \arg\min_b ||y_\mathcal{T} - h(x_\mathcal{T}|b)||_\mathcal{L}$ and $r_i = ||y_\mathcal{V} - h(x_\mathcal{V}|\beta_i)||_\mathcal{L}$.
end

Return: $r^* = \max_{i\in\{1,2,...,s\}} r_i$.

2.3 LWKO for multiple linear regression under the $\mathcal{L}_1$ norm

In this section, we consider a special case using multiple linear regression for the prediction function $h(x|\beta) = x\beta$ under the $\mathcal{L}_1$ norm and present an approach for calculating the LWKO criterion exactly. In such special case, model (1)–(6) reduces to the following bilevel optimization model:

$$\max_{w,\beta} \frac{1}{k}(1 - w)^\top |y - x\beta| \tag{7}$$

$$\text{s.t. } 1^\top w = n - k \tag{8}$$

$$w \in \mathbb{B}^n \tag{9}$$

$$\beta \in \arg\min_b \left\{ \frac{1}{n-k} w^\top |y - xb| \right\}. \tag{10}$$

Then, we reformulate model (7)–(10) as the following MILP, which can be solved efficiently by numerous algorithms [26–28] and commercial solvers such as CPLEX [29] and GUROBI [30]:

$$\max_{w,l,u,\lambda,\mu,r,\beta} \frac{1}{k}\left[1^\top r - (n-k)y^\top(\mu-\lambda)\right] \tag{11}$$

$$\text{s.t. } r \le Ml + x\beta - y \tag{12}$$

$$r \le Mu + y - x\beta \tag{13}$$

$$l + u \le 1 \tag{14}$$

$$1^\top w = n - k \tag{15}$$

$$r \ge x\beta - y \tag{16}$$

$$r \ge y - x\beta \tag{17}$$

$$x^\top(\mu - \lambda) = 0 \tag{18}$$

$$\lambda + \mu = \frac{w}{n-k} \tag{19}$$

$$w, l, u \in \mathbb{B}^n; \lambda, \mu \ge 0; r, \beta \text{ free.} \tag{20}$$

Theorem 1 *Bilevel model* (7)–(10) *and MILP* (11)–(20) *are equivalent in the sense that* (w, β) *is an optimal solution to* (7)–(10) *if and only if there exists* (l, u, λ, μ, r) *such that* $(w, l, u, \lambda, \mu, r, \beta)$ *is an optimal solution to* (11)–(20). *As such, the LWKO criterion can be calculated by solving the MILP* (11)–(20).

Proof This proof consists of three steps of equivalent reformulations. First, we reformulate (7)–(10) as follows to linearize the absolute value functions in both upper and lower levels:

$$\max_{w,l,u,r,\beta} \frac{1}{k}(1-w)^\top r \tag{21}$$

$$\text{s.t. } r \le Ml + x\beta - y \tag{22}$$

$$r \le Mu + y - x\beta \tag{23}$$

$$l + u \le 1 \tag{24}$$

$$1^\top w = n - k \tag{25}$$

$$w, l, u \in \mathbb{B}^n; r \text{ free} \tag{26}$$

$$\beta \in \arg\min_{s,b} \left\{ \frac{1}{n-k} w^\top s \,:\, x\beta - y \le s; y - x\beta \le s \right\} \tag{27}$$

Linearization of the absolute value under minimization at the lower level is straightforward. At the upper level, which is a maximization problem, it requires additional binary variables l and u, free variable r, a big-M parameter M, and new constraints to establish $r = |y - x\beta|$. Constraints (22)–(24) ensure that $r \le |y - x\beta|$, and the maximization objective will close any gap between r and $|y - x\beta|$.

Second, we replace constraint (27) with the optimality condition of the underlying linear program:

$$r \ge x\beta - y \tag{28}$$

$$r \ge y - x\beta \tag{29}$$

$$x^\top(\mu - \lambda) = 0 \tag{30}$$

$$\lambda + \mu = \frac{w}{n-k} \tag{31}$$

$$\lambda, \mu \ge 0, \tag{32}$$

where β and r correspond to, respectively, decision variables b and s at optimality.

Finally, according to strong duality, we have $\frac{1}{n-k} w^\top r = y^\top(\mu - \lambda)$, which helps convert the bilinear objective function $\frac{1}{k}(1 - w)^\top r$ into an equivalent linear one $\frac{1}{k}\left[1^\top r - (n-k) y^\top(\mu - \lambda)\right]$. □

3 Computational experiments

3.1 Data

We collected two data sets from publicly available sources for our computational experiments. The first dataset was from regression problem 3 of the CoEPrA (Comparative Evaluation of Prediction Algorithms) 2006 competition [31]. The explanatory data x has a dimension of 133 rows and $p = 5,787$ columns, representing 133 objects of nonapeptides, each described by 5,787 descriptors; the response data y represents the property of the 133 nonapeptides.

The second dataset was from the DRYAD repository: DOI:10.5061/dryad.fc55k [32]. After the removal of incomplete data points, the explanatory data x has a dimension of 2,034 rows and $p = 353$ columns, representing 2,034 mice specimens, each

genotyped at 353 single nucleotide polymorphisms on the 19 autosomal chromosomes; total body weight was used as the response data y. The datasets generated and/or analysed during the current study are available from the corresponding author on reasonable request.

3.2 Design of experiments

3.2.1 Experiment 1

Experiment 1 used the first dataset to represent the $n << p$ scenario. Eight cross validation criteria were selected to test the performance of the LWKO criterion in comparison with its counterpart LKO or (n/k)-fold for a range of k values under the $\mathcal{L}_1$ norm. For $k = 1$ and $k = 2$, LKO was calculated with brute force enumeration; for $k = 4$ and $k = 8$, enumeration became computationally expensive, thus (n/k)-fold was used instead as the counterpart. This experiment consisted of the following 6 steps.

Step 1 Randomly partition the data into an in-sample subset that consists of $n = 24$ data points and an out-of-sample subset that consists of the remaining 109 data points.
Step 2 Build ten models by solving the best subset problem with the parameter $t \in \{2, 4, 6, 8, 10, 12, 14, 16, 18, 20\}$:

$$\min_{\beta} \left\{ \frac{1}{n} ||y - x\beta||_{\mathcal{L}_1} \ : \ ||\beta||_0 \leq t \right\}.$$

Step 3 Cross validate the ten models from Step 2 with the following eight criteria using the in-sample data:

❶ Leave-1-out (L-1-O)	❷ Leave-worst-1-out (LW-1-O)
❸ Leave-2-out (L-2-O)	❹ Leave-worst-2-out (LW-2-O)
❺ 6-fold	❻ Leave-worst-4-out (LW-4-O)
❼ 3-fold	❽ Leave-worst-8-out (LW-8-O)

The LW-1-O and LW-2-O criteria were calculated with brute force enumeration; LW-4-O and LW-8-O were calculated using both the MILP model (11)–(20) (with a time limit for the solver) and the random sampling algorithm 1 to compare the performances of exact and approximate algorithms.
Step 4 For each cross validation criterion, select one model with the smallest cross validation error.
Step 5 Calculate the mean absolute errors of the eight selected models using out-of-sample data.
Step 6 Repeat Steps 1–5 three hundred times with different random partitions of in-sample and out-of-sample subsets.

3.2.2 Experiment 2

Experiment 2 was designed to use the second dataset to test the performance of LWKO for a general prediction model, random forest, under $\mathcal{L}_2$ norm using realistic explanatory and response data. This experiment had the following 6 steps.

Step 1 Randomly partition the data into an in-sample subset that consists of $n = 1000$ data points and an out-of-sample subset that consists of the remaining 1034 data points.
Step 2 Build ten random forest models with the number of variables to select at random for each decision split, q, being $q \in \{20, 40, 60, 80, 100, 120, 140, 160, 180, 200\}$. All other hyper-parameters are the same (100 trees in the forest, each created using 200 random data points).
Step 3 Cross validate the ten models from Step 2 with the following eight criteria using the in-sample data:

❶ Leave-1-out (L-1-O)	❷ Leave-worst-1-out (LW-1-O)
❸ 200-fold	❹ Leave-worst-5-out (LW-5-O)
❺ 100-foldww	❻ Leave-worst-10-out (LW-10-O)
❼ 50-fold	❽ Leave-worst-20-out (LW-20-O)

The LW-1-O criterion was calculated with brute force enumeration; LW-5-O; LW-10-O, and LW-20-O were calculated using the random sampling algorithm 1, in which parameter s was set to be $\frac{10n}{k}$ so that each data point is used ten times on average to build binary decision trees.
Step 4 For each cross validation criterion, select one model with the smallest cross validation error.
Step 5 Calculate the root mean square errors of the eight selected models using out-of-sample data.
Step 6 Repeat Steps 0–5 three hundred times with different random partitions of in-sample and out-of-sample subsets.

Experiment 2 had three major differences with experiment 1. First, linear regression models in experiment 1 were created by solving the best subset problem, whereas random forest was used as the prediction model in experiment 2. Second, experiment 1 represented a scenario with a small number of data points ($n = 24$) and a large number of variables ($p = 5,787$). In contrast, there was a larger number of data points ($n = 1,000$) and a smaller number of variables ($p = 353$) in experiment 2. Third, in experiment 1, both the MILP formulation (11)–(20) and the random sampling algorithm 1 were used to compute LWKO, whereas only the random sampling algorithm 1 was used under the $\mathcal{L}_2$ norm in experiment 2.

3.3 Simulation results

The two experiments were carried out in Matlab on a laptop with 16 GB of ram and 2.8 GHz CPU. Gurobi with default settings was used as the MILP solver for model (11)–(20); a time limit of 10 minutes was imposed for the solver, although it was able to terminate ahead of time in majority of the instances. The Matlab code for solving the MILP (11)–(20) was uploaded to https://github.com/lzwang2017/LWKO. Matlab function `fitrtree.m` was used to fit binary decision tree for regression.

3.3.1 Results from experiment 1

Table 1 gives an example of simulation result from experiment 1. Each column corresponds to a multiple linear regression model with t non-zero variable effects. The in-sample error, error_{in}, decreases as t increases from 2 to 20. In contrast, the out-of-sample errors, $\text{error}_{\text{out}}$, of the ten models do not have such monotonic property; the errors may be high for very small or large values of t due to underfitting or overfitting. An ideal cross validation criterion would be able to estimate the out-of-sample performance of the models by using in-sample data. The ten models were validated against the eight cross validation criteria using the in-sample data, and the validation errors are given in the table in four groups of rows, for comparison between LKO or (n/k)-fold and the proposed LWKO criteria, and also for comparison between exact and approximate algorithms for computing LWKO. For each of the eight criteria (in ten rows), the smallest validation error is bolded, and its corresponding t^* that achieved such smallest error is recorded. As a benchmark, the

Table 1 Results from a sample simulation of experiment 1

t	2	4	6	8	10	12	14	16	18	20
error_{in}	0.29	0.21	0.20	0.15	0.09	0.07	0.06	0.04	0.01	**0.01**
$\text{error}_{\text{out}}$	1.11	1.01	0.94	0.78	**0.75**	0.82	0.95	2.50	1.39	1.45
L-1-O	0.32	0.38	0.45	0.26	0.17	0.13	0.20	0.33	**0.08**	0.14
LW-1-O	1.31	3.02	4.54	0.97	0.63	0.75	0.48	1.73	**0.20**	1.04
L-2-O	0.31	0.36	0.34	0.26	0.18	0.13	0.19	0.36	**0.08**	0.30
LW-2-O	1.26	3.11	5.18	0.85	**0.49**	0.59	0.89	2.06	0.84	9.68
6-fold	0.35	0.38	0.32	0.33	0.19	**0.17**	0.32	0.38	0.18	1.33
LW-4-O (MILP)	**1.03**	2.64	4.20	0.93	1.06	1.32	2.90	376.68	999.89	999.84
LW-4-O (RS)	1.49	**1.09**	1.27	1.38	1.49	2.57	3.50	7.06	115.66	510.61
3-fold	0.36	0.40	0.56	**0.31**	0.37	0.49	0.72	2.93	7.57	17.23
LW-8-O (MILP)	**1.06**	5.62	6.08	4.03	3.66	25.83	717.38	890.01	976.59	999.93
LW-8-O (RS)	**1.06**	1.13	1.27	2.75	3.26	2.68	103.81	47.84	358.81	38.43

Bolded errors are the smallest ones in each row. LW-4-O and LW-8-O were computed using both MILP and random sampling (RS) algorithms

Table 2 Average computation time (in seconds) for ten different models under LW-4-O (MILP) and LW-8-O (MILP) criteria in experiment 1

t	2	4	6	8	10	12	14	16	18	20
LW-4-O (MILP)	3	28	139	> 301	> 442	> 485	> 478	> 440	275	42
LW-8-O (MILP)	6	42	218	> 509	> 583	> 597	> 599	> 598	> 588	> 542

The > signs are used when the 10-minute time limit had been exceeded and 600 seconds recorded as the computation time

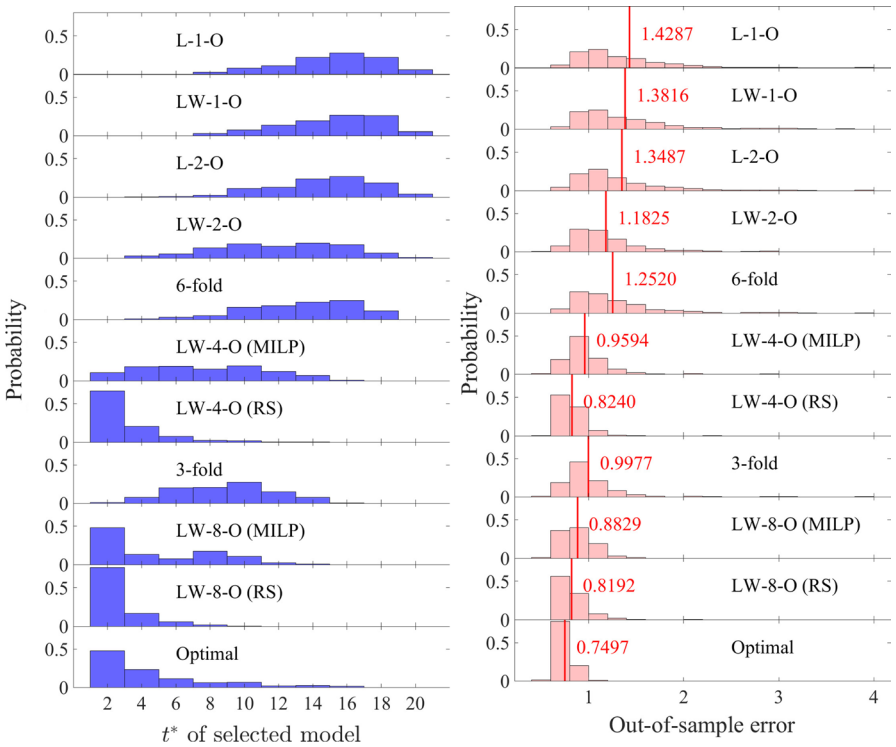

Fig. 1 *Left* Histograms of t^* for the selected models of eight cross validation criteria using the in-sample data over 300 random simulations in experiment 1. Results for the "optimal" criterion were obtained after observing the out-of-sample data, which represent the best possible performance of any cross validation criterion. *Right* Histograms and means of $\text{error}_{\text{out}}$ for the selected models of eight cross validation criteria using the in-sample data over 300 random simulations in experiment 1

smallest $\text{error}_{\text{out}}$ and its corresponding t^* are also recorded, which represent the best possible performance of any cross validation criterion in this experiment.

The sample simulation of Table 1 was repeated for 300 times randomly and independently, each time with a different partition of the in-sample and out-of-sample data sets. Average computation time for ten different models under LW-4-O (MILP) and LW-8-O (MILP) criteria are shown in Table 2. The histograms of t^* for the selected models and the corresponding $\text{error}_{\text{out}}$ are plotted in the two

Table 3 Relative performance of LWKO and the counterpart LKO or (n/k)-fold in experiment 1, in terms of the percentages of times that LWKO led to a lower (better), higher (worse), or the same $\text{error}_{\text{out}}$ in 300 random simulations

	LWKO better (%)	LWKO worse (%)	Same (%)
LW-1-O versus L-1-O	19	14	67
LW-2-O versus L-2-O	40	13	46
LW-4-O (MILP) versus 6-fold	65	15	21
LW-4-O (RS) versus 6-fold	89	11	0
LW-8-O (MILP) versus 3-fold	58	27	15
LW-8-O (RS) versus 3-fold	76	20	4

The sum of each row may not add up to 100% due to rounding errors

subfigures in Fig. 1. We make three observations. First, when used for a very small k value (e.g., $k = 1$ or $k = 2$), LKO may not be an effective indicator of the model's out-of-sample performance. Second, compared with LKO for the same k value, the LWKO criterion is more likely to lead to a smaller out-of-sample error. Third, approximate LWKO calculated using the random sampling algorithm could be an even more effective indicator of the model's out-of-sample performance than the exact LWKO.

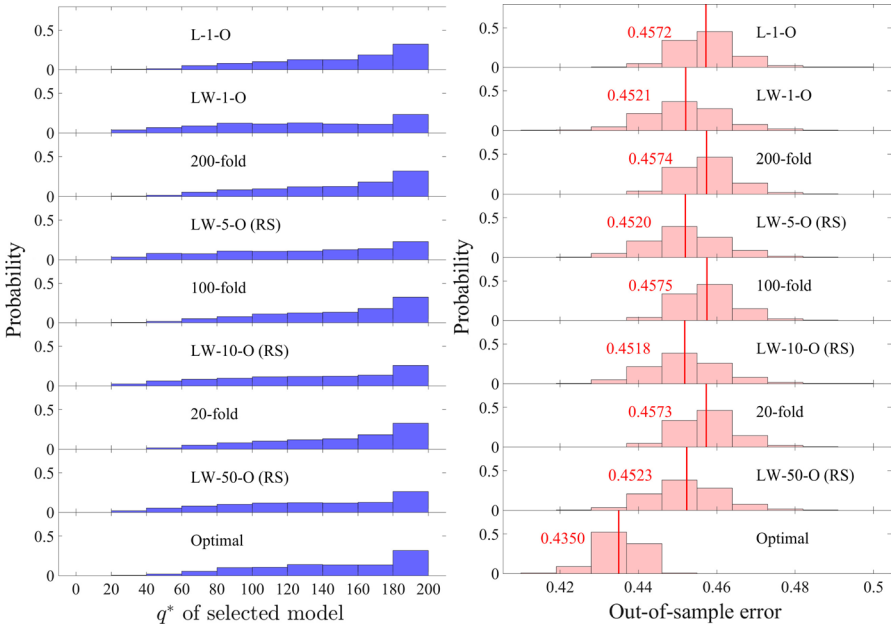

Fig. 2 *Left* Histograms of q^* for the selected models of eight cross validation criteria using the in-sample data over 300 random simulations in experiment 2. Results for the "optimal" criterion were obtained after observing the out-of-sample data, which represent the best possible performance of any cross validation criterion. *Right* Histograms and means of $\text{error}_{\text{out}}$ for the selected models of eight cross validation criteria using the in-sample data over 300 random simulations in experiment 2

Table 4 Relative performance of LWKO and the counterpart LKO or (n/k)-fold in experiment 2, in terms of the percentages of times that LWKO led to a lower (better), higher (worse), or the same $\text{error}_{\text{out}}$ in 300 random simulations

	LWKO better (%)	LWKO worse (%)	same (%)
LW-1-O versus L-1-O	58	19	23
LW-5-O (RS) versus 200-fold	58	19	23
LW-10-O (RS) versus 100-fold	60	17	22
LW-50-O (RS) versus 20-fold	56	18	26

The sum of each row may not add up to 100% due to rounding errors

Direct comparisons between LKO and LWKO are presented in Table 3, which summarizes the percentages of times that the models selected by the LWKO cross validation criteria have led to a lower, higher, or the same out-of-sample errors in experiment 1, compared with the counterpart LKO or (n/k)-fold criteria. Results suggested that (both exact and approximate) LWKO outperformed the counterparts LKO and (n/k)-fold for all four compared cases of $k \in \{1, 2, 4, 8\}$, with $k = 4$ showing the most significant advantage.

3.3.2 Results from experiment 2

The histograms of q^* for the selected models and the corresponding $\text{error}_{\text{out}}$ are plotted in the two subfigures in Fig. 2. Direct comparisons between LKO and LWKO are presented in Table 4. The observations from these results are similar with those from experiment 1, suggesting a consistent performance of the LWKO criterion for different sizes of data sets, different prediction models, and different norms. Results from experiment 2 also suggest that, even when the optimality of the worst validation dataset cannot be guaranteed, the LWKO criteria calculated using the random sampling algorithm 1 could still be more effective than their counterparts LKO or (n/k)-fold in estimating the performance of general prediction models for out-of-sample data.

Table 5 compares the computation time for ten different models under eleven cross validation criteria for a random sample simulation in experiment 2. Results suggested that, using algorithm 1, leave-worst-k-out has a lower yet comparable

Table 5 Computation time (in seconds) for ten different models under eight cross validation criteria for a random sample simulation in experiment 2

q	20	40	60	80	100	120	140	160	180	200
L-1-O	16	16	16	16	16	16	17	17	17	17
LW-1-O (enumeration)	16	16	16	16	16	16	17	17	17	17
200-fold	5	6	6	6	6	6	6	7	7	7
LW-5-O (RS)	36	37	37	38	39	39	40	41	41	41
100-fold	7	7	8	8	9	9	10	11	11	11
LW-10-O (RS)	24	28	26	27	27	28	30	31	32	32
50-fold	18	24	23	25	27	28	31	32	34	36
LW-20-O (RS)	56	66	66	71	75	80	85	90	92	100

computational speed with leave-1-out and (n/k)-fold for a wide range of k. Moreover, parameter s in algorithm 1 can also be adjusted to balance the tradeoff between speed and solution quality.

4 Conclusion

We proposed the leave-worst-k-out as a new criterion for cross validation. This work makes three major contributions. First, we proved that, for the special case of using multiple linear regression as the prediction model under $\mathcal{L}_1$ norm, the LWKO criterion is an optimal solution to a mixed integer linear program, which can be solved by numerous algorithms and commercially available solvers more efficiently than brute force enumeration. Second, we proposed a random sampling algorithm for computing LWKO approximately for general prediction models and norms. Third, we demonstrated with two computational experiments that LWKO not only is more effective than its counterparts LKO and (n/k)-fold but also can be computed much more efficiently than LKO and almost as fast as (n/k)-fold.

As a new cross validation criterion, LWKO faces similar challenges as LKO does. First, using the random sampling algorithm 1, LWKO has a higher yet comparable complexity with leave-1-out and (n/k)-fold, which may be considered computationally intractable under certain circumstances. When effectiveness is more important than speed, our simulation results suggested that using the worst validation error instead of the average could be more informative, even if the search for the k worst validation data points is approximate and non-exhaustive. Second, there is no guaranteed correlation between LWKO and the out-of-sample performance, although such correlation appeared to be higher than that of LKO according to our computational experiment results. Third, the selection of value k is more of an art than science, in the sense that the extent to which LWKO outperformed LKO did not demonstrate a clear pattern with respect to k. However, it was clear from both experiments that the popular choice of leave-1-out was not the most effective criterion for cross validation. Since algorithm 1 applies to nonlinear models and more general norms, testing the LWKO criterion on more prediction models (such as neural networks) appears to be an interesting follow up research direction.

Acknowledgements This work was partially supported by the National Science Foundation under the LEAP HI and GOALI programs (Grant Number 1830478) and under the EAGER program (Grant Number 1842097) and the Plant Sciences Institute at Iowa State University. This manuscript was greatly improved thanks to constructive and insightful feedback from the Associate Editor and an anonymous reviewer. The author is grateful to Dr. Qing Li and Lijie Liu for the suggestion of the CoEPrA data source and to Dr. Guiping Hu and Dr. Dan Nettleton for inspiring conversations about the proposed LWKO criterion.

References

1. Hawkins, D.M.: The problem of overfitting. J. Chem. Inf. Comput. Sci. **44**(1), 1–12 (2004)

2. Trippa, L., Waldron, L., Huttenhower, C., Parmigiani, G., et al.: Bayesian nonparametric cross-study validation of prediction methods. Ann. Appl. Stat. **9**(1), 402–428 (2015)
3. Burnham, K.P., Anderson, D.R., Huyvaert, K.P.: Aic model selection and multimodel inference in behavioral ecology: some background, observations, and comparisons. Behav. Ecol. Sociobiol. **65**(1), 23–35 (2011)
4. Candes, E., Tao, T., et al.: The dantzig selector: statistical estimation when p is much larger than n. Ann. Stat. **35**(6), 2313–2351 (2007)
5. Bartlett, P.L., Long, P.M., Lugosi, G., Tsigler, A.: Benign overfitting in linear regression. In: Proceedings of the National Academy of Sciences. (2020)
6. Falkner, B., Schröder, G.F.: Cross-validation in cryo-EM-based structural modeling. Proc. Natl. Acad. Sci. **110**(22), 8930–8935 (2013)
7. Scheres, S.H., Chen, S.: Prevention of overfitting in cryo-EM structure determination. Nat. Methods **9**(9), 853 (2012)
8. Vehtari, A., Gelman, A., Gabry, J.: Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC. Stat. Comput. **27**(5), 1413–1432 (2017)
9. Celisse, A., et al.: Optimal cross-validation in density estimation with the L{2} -loss. Ann. Stat. **42**(5), 1879–1910 (2014)
10. Airola, A., Pahikkala, T., Waegeman, W., De Baets, B., Salakoski, T.: An experimental comparison of cross-validation techniques for estimating the area under the roc curve. Comput. Stat. Data Anal. **55**(4), 1828–1844 (2011)
11. Cawley, G.C., Talbot, N.L.: Efficient leave-one-out cross-validation of kernel fisher discriminant classifiers. Pattern Recogn. **36**(11), 2585–2592 (2003)
12. Homrighausen, D., McDonald, D.J.: Leave-one-out cross-validation is risk consistent for lasso. Mach. Learn. **97**(1–2), 65–78 (2014)
13. Rodriguez, J.D., Perez, A., Lozano, J.A.: Sensitivity analysis of k-fold cross validation in prediction error estimation. IEEE Trans. Pattern Anal. Mach. Intell. **32**(3), 569–575 (2009)
14. Fushiki, T.: Estimation of prediction error by using k-fold cross-validation. Stat. Comput. **21**(2), 137–146 (2011)
15. Blum, A., Kalai, A., and Langford, J. Beating the hold-out: bounds for k-fold and progressive cross-validation. In: Proceedings of the Twelfth Annual Conference on Computational Learning Theory, pp. 203–208. (1999)
16. Kohavi, R., et al.: A study of cross-validation and bootstrap for accuracy estimation and model selection. In: IJCAI, vol. 14, pp. 1137–1145. (1995)
17. Magnusson, M., Vehtari, A., Jonasson, J., Andersen, M.: Leave-one-out cross-validation for Bayesian model comparison in large data. In: International Conference on Artificial Intelligence and Statistics, pp. 341–351. PMLR (2020)
18. Xu, L., Hu, O., Guo, Y., Zhang, M., Lu, D., Cai, C.-B., Xie, S., Goodarzi, M., Fu, H.-Y., She, Y.-B.: Representative splitting cross validation. Chemom. Intell. Lab. Syst. **183**, 29–35 (2018)
19. Jung, Y.: Multiple predicting k-fold cross-validation for model selection. J. Nonparametric Stat. **30**(1), 197–215 (2018)
20. Ramezan, A., Warner, A., Maxwell, A.: Evaluation of sampling and cross-validation tuning strategies for regional-scale machine learning classification. Remote Sens. **11**(2), 185 (2019)
21. Duarte, E., Wainer, J.: Empirical comparison of cross-validation and internal metrics for tuning svm hyperparameters. Pattern Recogn. Lett. **88**, 6–11 (2017)
22. Sampath, R., Indumathi, J.: Earlier detection of Alzheimer disease using n-fold cross validation approach. J. Med. Syst. **42**(11), 1–11 (2018)
23. Horvat, T., Havaš, L., Srpak, D.: The impact of selecting a validation method in machine learning on predicting basketball game outcomes. Symmetry **12**(3), 431 (2020)
24. Cossio, P.: Need for cross-validation of single particle cryo-EM. J. Chem. Inf. Model. **60**(5), 2413–2418 (2020)
25. Adnan, R.M., Liang, Z., Yuan, X., Kisi, O., Akhlaq, M., Li, B.: Comparison of lssvr, m5rt, nf-gp, and nf-sc models for predictions of hourly wind speed and wind power based on cross-validation. Energies **12**(2), 329 (2019)
26. Bénichou, M., Gauthier, J.-M., Girodet, P., Hentges, G., Ribière, G., Vincent, O.: Experiments in mixed-integer linear programming. Math. Program. **1**(1), 76–94 (1971)
27. Codato, G., Fischetti, M.: Combinatorial benders' cuts for mixed-integer linear programming. Oper. Res. **54**(4), 756–766 (2006)

28. Testa, A., Rucco, A., Notarstefano, G.: Distributed mixed-integer linear programming via cut generation and constraint exchange. IEEE Trans. Autom. Control **65**, 1456–1467 (2019)
29. Cplex, I.I.: V12. 1: User's manual for cplex. Int. Bus. Mach. Corp. **46**(53), 157 (2009)
30. Gurobi Optimization, I. Gurobi Optimizer Reference Manual. *URL http://www. gurobi. com*. (2018)
31. Comparative Evaluation of Prediction Algorithms, C. http://www.coepra.org/CoEPrA_regr.html. (2006)
32. Mitteroecker, P., Cheverud, J., Pavlicev, M.: Multivariate analysis of genotype-phenotype association. Genetics **202**(4), 1345–1363 (2016)