RESEARCH ARTICLE

Optimally adjusted last cluster for prediction based on balancing the bias and variance by bootstrapping

Jeongwoo Kim^{1,2*}

1 Korea Maritime Institute, Busan, Republic of Korea, **2** Biomedical Research Center, Asan Institute for Life Sciences, Seoul, Republic of Korea

* jwkim@kmi.re.kr

OPEN ACCESS

Citation: Kim J (2019) Optimally adjusted last cluster for prediction based on balancing the bias and variance by bootstrapping. PLoS ONE 14(11): e0223529. <https://doi.org/10.1371/journal.pone.0223529>

Editor: Peng Li, Rutgers University, UNITED STATES

Received: March 13, 2019

Accepted: September 23, 2019

Published: November 4, 2019

Copyright: © 2019 Jeongwoo Kim. This is an open access article distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Data Availability Statement: The minimal data set necessary to replicate the study is available from the following DOI: [10.17605/OSF.IO/TQUYX](https://doi.org/10.17605/OSF.IO/TQUYX). (The simulation data sets were randomly generated for each simulation experiment and are not included.)

Funding: The authors received no specific funding for this work.

Competing interests: The authors have declared that no competing interests exist.

Abstract

Estimating a predictive model from a dataset is best initiated with an unbiased estimator. However, since the unbiased estimator is unknown in general, the problem of the bias-variance tradeoff is raised. Aside from searching for an unbiased estimator, the convenient approach to the problem of the bias-variance tradeoff may be to use the clustering method. Within a cluster whose size is smaller than the whole sample, we would expect the simple form of the estimator for prediction to avoid the overfitting problem. In this paper, we propose a new method to find the optimal cluster for prediction. Based on the previous literature, this cluster is considered to exist somewhere between the whole dataset and the typical cluster determined by partitioning data. To obtain a reliable cluster size, we use the bootstrap method in this paper. Additionally, through experiments with simulated and real-world data, we show that the prediction error can be reduced by applying this new method. We believe that our proposed method will be useful in many applications using a clustering algorithm for a stable prediction performance.

Introduction

When analyzing relationships among variables, including random variables with an equation of $y = f(x) + e$, where e is a random vector of a certain distribution with zero mean and preset covariance, two main problems are generally encountered: one is related to $f(x)$ and the other to e . The latter mainly includes heteroscedasticity and serial correlation problems, which induce efficiency of the estimator to decrease, thereby hindering rigorous hypothesis testing. The former, which is related to the misspecification problem leading to biased estimator, frequently presents a critical problem. In numerous applications, unless an underlying theoretical foundation for an approximation model has been rigorously established or is widely acknowledged in a related area, one is prone to obtaining a biased estimator.

If the purpose of a study is to verify an exact relationship among the given variables of a certain dataset, the first priority is to set the approximation model with the best fit—i.e., an unbiased model specific to the given dataset. However, for prediction purposes, one may not

struggle to obtain an unbiased estimator to yield the smallest error metric by considering only the given dataset. In fact, focusing only on the given dataset may result in the overfitting problem, which hinders accurate prediction; therefore, a biased approximation model can sometimes be useful for the purpose of prediction.

Using a biased approximation model, the more instances we use, the greater the predictor bias we obtain. Therefore, with a biased approximation model, a subsampling approach could yield a more accurate performance up to the point where the gain from the reduced bias exceeds the loss from the increased variance [1]. Conversely, the subsampling approach could have serious drawbacks when the increased variance considerably exceeds the reduced bias because the convergence rate of variance is generally $O(N^{-1})$, where N is the sample size. Moreover, searching for every neighbor reference of a target point, such as the k -nearest neighbor and moving average methods, is computationally expensive [2]. Considering these factors, a more efficient method would be to partition the data into clusters. Clustering methods are easy to implement and extensively used as one of the unsupervised learning methods. However, in general, the number of clusters is not analytically obtained, but numerically searched in a set of candidate numbers. Therefore, it is prone to lead to the overfitting problem [3, 4]. With the effect of the overfitting problem, a poor cluster would cause high variance [5] because the small size of the subsample usually displays a high variance [6, 7]. Thus, determining how to adequately adjust the size of the cluster would be a key approach to overcome the high variance and the overfitting problem. In particular, to adjust the size of the cluster for prediction, we may consider the cluster close to a prediction target, unlike the usual clustering method fitting a whole dataset. In the current study, the term *last cluster* indicates the cluster located closest to a prediction target after partitioning the data into clusters using a method such as k -means clustering because considering only the last cluster in a prediction problem can be an efficient way to achieve accurate prediction. For example, Oh and Kim [8] obtained a good prediction performance using stock market data by referring to the last homogeneous clusters obtained from the data.

Meanwhile, the high variance problem using the last cluster can still present a problem during prediction because it is difficult to determine the size of the last cluster that ensures an accurate prediction. Hence, balancing the bias and the variance would be useful for the prediction problem using the last cluster. As mentioned above, adjusting the size of the last cluster would reduce the variance of a cluster obtained from partitioning data, which helps balancing the bias and the variance, so that the prediction accuracy would improve. Thus, in the current study, adjusting the size of the last cluster for the accurate prediction will be focused on.

Consider a function $f(\mathbf{x})$ and its arbitrary estimator $\hat{f}(\mathbf{x})$. Then, the mean squared error (MSE) of $\hat{f}(\mathbf{x})$ is decomposed into $E[(f(\mathbf{x}) - E[\hat{f}(\mathbf{x})])^2] + E[(E[\hat{f}(\mathbf{x})] - \hat{f}(\mathbf{x}))^2]$, which is a squared bias plus a variance. Hence, we can reduce the MSE by controlling the squared bias and the variance. However, unless $f(\mathbf{x})$ is known, it is not easy to directly control the bias by specifying an estimator. Therefore, the method of reducing the variance can be more efficient. In addition, model regularization using a shrinkage method, such as least absolute shrinkage and selection operators (LASSO), is also popular for balancing bias and variance. In model estimation, these methods avoid the overfitting problem by adopting a new hyperparameter, termed a tuning parameter, that leads to increasing bias but decreasing variance through a smoothing effect. In general, this tuning parameter is usually obtained by cross-validation [9] and is helpful in alleviating the overfitting problem.

Aside from the model regularization methods, other unique approaches to the bias-variance balance in the econometrics community exist that address the trade-off between bias and variance with structural breaks in a dataset because structural breaks can be useful in determining

where to partition a given dataset. Elliott [10] attempted to improve prediction accuracy by using weighted-averaging on the regression results for multiple structural breaks. Pesaran and Timmermann [11] presented some approaches based on structural breaks to determine a window (or cluster) size to achieve accurate prediction. Furthermore, also considering the structural breaks, Clark and McCracken [12] proposed a novel approach to improve the prediction accuracy by combining two estimated regression coefficients using a new hyperparameter, similar to LASSO.

The bootstrap method is well-known for reducing the bias [13] and variance [14] of an estimator. This method allows us to approximate an underlying functional form of a given dataset by averaging the noise of different bootstrapped samples out, thereby reducing bias. Furthermore, it is useful in obtaining an estimator with low variance through training a model multiple times. In particular, the relative advantage of variance reduction is often achieved at the expense of a slight increase in bias [15], which is one of the reasons why the bagging algorithm can improve model accuracy. Bauer and Kohavi [16] reported numerical results on the relatively greater variance reduction compared to bias reduction by using the bagging algorithm in a series of experiments. Fumera et al. [17] also showed that the bagging algorithm of N bootstrap replicates induces the same bias as that of a single bootstrap replicate but decreases the variance by a factor of N .

Therefore, combining the idea from a new model parametrization (such as the shrinkage method or the combination of two estimated regression coefficients) with the bootstrap method would be useful in balancing bias and variance, thus improving model prediction accuracy. Additionally, because a data-fitting-oriented model often leads to overfitting in many applications, it would be more efficient to adjust the size of the last cluster rather than to fit the entire dataset. In this study, the size-adjusted last cluster is termed as an adjusted last cluster (aCL), which has a size that falls between the size of the entire dataset and that of a last cluster by partitioning data. We expect this approach would improve the prediction performance of models based on the balance between bias and variance.

Materials and methods

Notations and assumptions

Here, we follow some conventional notations. Hat ($\hat{\cdot}$) notations such as $\hat{a}$ indicate an estimate, and an overbar ($\bar{\cdot}$) such as $\bar{a}$ denotes an average value.

Consider a given dataset $Z_N = \{Z_n : Z_n = (\mathbf{x}_n, y_n) \in \mathbb{R}^{m+1}, n = 1, \dots, N\}$, where $\mathbf{x}_n = (x_{1n}, x_{2n}, \dots, x_{mn})^T \in \mathbb{R}^m, y_n \in \mathbb{R}$, in which the Z_n 's are arranged by the norm of $\mathbf{x}_n, \|\mathbf{x}_n\|$, for prediction purposes. We want to determine the aCL to accurately predict y_{N+h} at $\mathbf{x}_{N+h}$ beyond $\mathbf{x}_N$, where h is the prediction horizon. To accomplish this, assume that $\mathbf{x}_n$ and y_n follow a data generating process (DGP) as follows:

$$y_n = f(\mathbf{x}_n) + e_n, \quad (1)$$

where e_n is an independent and identically distributed random variable with zero mean, a variance of σ^2 and is independent of $\mathbf{x}_n$.

Considering an estimator $\hat{f}(\mathbf{x}_n)$ based on Z_N for $f(\mathbf{x}_n)$, our goal would then be to estimate $\hat{f}(\mathbf{x}_n)$ to accurately predict y_{N+h} at $\mathbf{x}_{N+h}$. The function $\hat{f}(\mathbf{x}_n)$ is usually obtained by minimizing the risk function as follows:

$$R(\hat{f}(\mathbf{x}_n)) = E[(y_n - \hat{f}(\mathbf{x}_n))^2], \quad (2)$$

where $E[\cdot]$ is the expected value.

Using $\hat{f}(\mathbf{x}_n)$, the prediction error (PE) is given by

$$\text{PE}(\hat{f}(\mathbf{x}_{N+h})) = E[(y_{N+h} - \hat{f}(\mathbf{x}_{N+h}))^2]. \quad (3)$$

In the Results and Discussion section, in order to verify the method demonstrated in this paper, we will compare the prediction performances of the methods presented in this paper using the PE and test the statistical significance of the prediction performances.

Bias and variance of a predictor: Whole dataset and subsample

In practice, the risk function is estimated as follows:

$$R(\hat{f}(\mathbf{x}_n)) = \frac{1}{N} \sum_{n=1}^N (y_n - \hat{f}(\mathbf{x}_n))^2. \quad (4)$$

Considering a linear estimator $\hat{f}(\mathbf{x}_n) = \sum_{k=1}^m b_k x_{kn} = \beta^T \mathbf{x}_n$ for $\beta \in \mathbb{R}^m$; the estimate is obtained by minimizing the risk function with respect to β :

$$\begin{aligned} \hat{\beta} &= \arg \min_{\beta \in \mathbb{R}^m} R(\hat{f}(\mathbf{x}_n)) \\ &= (\mathbf{X}_N^T \mathbf{X}_N)^{-1} \mathbf{X}_N^T \mathbf{y}_N, \end{aligned} \quad (5)$$

where $\mathbf{X}_N = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N)^T$ and $\mathbf{y}_N = (y_1, y_2, \dots, y_N)^T$. Then, the predictor is $\hat{f}(\mathbf{x}_{N+h}) = \hat{\beta}^T \mathbf{x}_{N+h}$.

However, this predictor is based on the assumption that a linear estimator is applicable to the whole dataset; hence, this predictor produces a vulnerable prediction performance depending on the shape of the true function. For example, consider a DGP, $y_n = -0.5x_{1n} + x_{1n}^2 + e_n$, $e_n \sim N(0,1)$ and a linear estimator $\hat{f}(x_{1n}) = a_0 + a_1 x_{1n}$ over the whole dataset. Then, the value predicted by this estimator over the whole dataset would be quite distant from the prediction target as shown in Fig 1. However, the predicted value from a subsample of the given dataset achieves a more accurate prediction.

Furthermore, as more samples are used, the bias of the predictor increases but its variance decreases, as shown in Fig 2. In general, an estimator from a subsample suffers from high variance, whereas the estimator from a whole dataset suffers from high bias [18]. Therefore, a size-adjusted subsample optimal for prediction should exist based on the balance between bias and variance [19]. For example, the size of such a subsample would be approximately 26, as seen in Fig 2.

To obtain the optimal subsample for prediction, first it is necessary to partition a given dataset using a method such as k-means clustering. However, determining the size of the last cluster for prediction may not be easy. To reliably determine the size of the last cluster, knowing the change points, structural breaks and local extrema of the given DGP can be useful. In practice, such values are not known *a priori*; one must guess the size of the last cluster by simply observing a graph of the DGP in an exogenous manner. However, partitioning the dataset in such a manner might be unreliable because the dataset observed is only a single realized sample among all the possible samples generated by the DGP. Thus, to obtain the size of the last cluster reliably, we would need to consider as many realized samples as possible, which would be unfeasible in practice. As an alternative, the bootstrap method used in [20] can be useful for virtually mimicking these samples to obtain the reliable last cluster.

Fig 1. Prediction when using the whole dataset (whole, solid line) and a subsample (subsample, dashed line).

<https://doi.org/10.1371/journal.pone.0223529.g001>

Adjusted last cluster for prediction

When simply partitioning one realized dataset, the size of the last cluster may be estimated to be smaller than the last cluster of the true function because the partitioning process of one realized dataset tends to be affected by noise, e_n . To avoid this problem, adequately expanding the size of the last cluster results in reducing a predictor's variance and would therefore be an alternative for approximating the last cluster of the true function. Hence, the aCL between the last cluster achieved by partitioning one realized dataset and the whole dataset would yield a more accurate prediction performance as long as the decreased variance by expanding the size of the last cluster exceeds the increased bias, similar to a regularization method such as LASSO [21].

In the following example, Fig 3A depicts the prediction by one realized sample (dotted line). A true function (solid line) is $f(x)$, and a linear estimator is used. The size of the last cluster (cluster, dashed line) from which a predictive value is obtained appears to be relatively small and is much influenced by noise, hence rendering a nonnegligible gap between the predictive value (circle) and the prediction target (asterisk). In contrast, Fig 3B shows that the new sample (BTSmean, dashed line with square) averaged over bootstrapped samples (BTS, dotted lines) well approximates the true function $f(x)$. The last cluster (aCL, dashed line) based on the new sample is larger than the last cluster of the sample in Fig 3A. Thus, the predictive value of this last cluster more accurately indicates the prediction target.

Fig 2. Change of MSE, squared bias (Bias²) and variance (Var) of the predictor according to the number (n) of samples used in estimation. Here, the DGP and the estimator are the same as those in Fig 1.

<https://doi.org/10.1371/journal.pone.0223529.g002>

Fig 3. Prediction based on the last cluster of one realized sample (A) and the last cluster of the new sample from bootstrapped samples (B). The predictive value (circle) in 3B indicates the prediction target (asterisk) more accurately than does the predictive value (circle) in 3A. DGP: $y_n = -2x_{1n} + x_{1n}^2 + e_n$, $e_n \sim N(0,1)$.

<https://doi.org/10.1371/journal.pone.0223529.g003>

Analogous to the idea of aCL, Clark and McCracken [12] proposed a novel approach to obtain a new regression coefficient for more accurate prediction by linearly combining two estimated regression coefficients under structural breaks: one estimated from a last rolling window (or a last cluster) and the other estimated from a recursive window (or a whole dataset).

Consider a sample $\mathbf{Z}_N$ realized from (1) and the linear estimator in (4)–(5), and denote the estimated regression coefficient obtained from the recursive window of size N as β_R and the estimated regression coefficient obtained from the last rolling window of size L as β_L . Then, the linear combination of these two estimated regression coefficients β_C is

$$\beta_C = \alpha\beta_L + (1 - \alpha)\beta_R, \quad 0 \leq \alpha \leq 1. \quad (6)$$

Hence, β_C is a compromise solution between β_L and β_R . Clark and McCracken [12] showed that β_C allows more accurate prediction by inducing lower bias than β_R and lower variance than β_L .

However, in [12], only one realized dataset was considered, and the regression coefficients already estimated from the last rolling window and the recursive window were used for β_C rather than being based on a last cluster optimal for prediction.

Therefore, if a sufficient number of realized datasets from a certain DGP are given and β_C is estimated from using these datasets, we can obtain a more generalized estimate for β_C , which leads to more accurate prediction.

However, in practice, we cannot acquire many realized datasets from the DGP. Hence, in this paper, we adopt the bootstrap method to reproduce such datasets and achieve more accurate prediction as shown in Fig 3. We summarize the aCL method based on the bootstrap method as below.

Algorithm Prediction (one-step ahead) using the aCL

- 1: Given a set $\mathbf{Z}_N = \{\mathbf{Z}_n : \mathbf{Z}_n = (\mathbf{x}_n, y_n) \in \mathbb{R}^{m+1}, n = 1, \dots, N\}$
- 2: **for** $i \in \{1, \dots, m\}$ **do**
- 3: Reproduce the i -th bootstrap dataset,
 $\mathbf{Z}_N^i = \{\mathbf{Z}_n^i : \mathbf{Z}_n^i = (\mathbf{x}_n, y_n) \in \mathbb{R}^{m+1}, n = 1, \dots, N\}$
- 4: Obtain a last cluster by partitioning the dataset (such as k-means clustering), and denote the size of the last cluster as L^i
- 5: Let the size of aCL be $C^i = \alpha_i L^i + (1 - \alpha_i)N$, $0 \leq \alpha_i \leq 1$
- 6: Estimate $\hat{C}^i$ by minimizing the value of the risk function,
 $\hat{C}^i = \arg \min_{L^i \leq C^i \leq N} \frac{1}{C^i} \sum_{n=N-C^i+1}^N (y_n - \hat{f}(\mathbf{x}_n))^2$
- 7: **end for**
- 8: Calculate the average of the $\hat{C}^i$'s, $\bar{C} = \frac{1}{m} \sum_{i=1}^m \hat{C}^i$
- 10: Estimate the regression coefficient using
 $\bar{C}, \hat{\beta}_C = \arg \min_{\beta_C \in \mathbb{R}^m} \frac{1}{\bar{C}} \sum_{n=N-\bar{C}+1}^N (y_n - \hat{f}(\mathbf{x}_n))^2$
- 11: Using $\hat{\beta}_C$, construct the predictor $\hat{f}(\mathbf{x}_{N+1})$ for y_{N+1}

Hence, the aCL acquired via the bootstrapped datasets is used for the predictor $\hat{f}(\mathbf{x}_{N+1})$ rather than the last cluster obtained by simply partitioning the dataset, and it is expected to achieve a more accurate prediction performance based on the balance between bias and variance. The prediction performance of the presented method will be compared with other relevant methods in the Results and Discussion section.

Methods for evaluating prediction error

In this section, we present several methods for comparing the prediction performance of the aCL method. First, we briefly introduce three existing optimization methods to check whether the aCL method improves the prediction performance of the optimization methods. Next, as

competitors to the aCL method, we introduce three methods relevant to the aCL method. The numerical results of the prediction errors comparison are then presented in the Results and Discussion section.

Optimization methods. Because the aCL method is a way of searching for a last cluster optimal for prediction, we can apply it to general optimization methods. Therefore, we need to assess whether combining the aCL method with an optimization method reduces the prediction error of the optimization methods.

First, we combine the aCL method with ordinary least square (OLS), which is the most widely used optimization method due to its ease of use and easy interpretation. Under certain conditions, OLS is known to yield the estimator with the lowest variance among linear unbiased estimators [22]; however, in practice, it is not easy to create a linear unbiased estimator covering an entire dataset. However, a linear estimator can fit the data well when considering only a subsample of the entire dataset.

Least squares adjusted with pseudo data. The aCL method was next applied to a new optimization method for prediction—least squares adjusted with pseudo data (LSPD) [23]. The main principle of LSPD is to convert test error (or prediction error) into training error (or residual) by adopting a pseudo data point for a prediction target.

In general, to calculate the prediction target, the parameters of a predictive model are estimated by minimizing the training error using a given dataset, which can often lead to overfitting. Focusing only on a given dataset during optimization is one of the biggest reasons for the occurrence of the overfitting problem because it results in estimated parameters that fail to accurately predict the prediction target beyond the given dataset. Some methods using pseudo data beyond the given dataset have been suggested in the nonparametric community. For example, Schuster [24] suggested the reflection data method. When estimating kernel density with a dataset $\{X_1, X_2, \dots, X_N\}$, this method claimed that adding a pseudo dataset $\{-X_1, -X_2, \dots, -X_N\}$ to the former could yield a consistent kernel density estimator based on the symmetry property of the kernel density function. Similarly, Cowling and Hall [25] proposed a general version of the reflection data method that can be used when discontinuities in the density derivatives exist.

However, unlike in kernel density function estimation, we are unable to use the symmetry property of a kernel density function in the prediction problem. There is little research on using pseudo data in the prediction problem, but Breiman [26] proposed a pseudo data method that reproduces the pseudo data from a certain convex combination of a given dataset to achieve more accurate prediction. Similarly, LSPD supposes that we can attain some pseudo data points close to the prediction target and add those to the given dataset to generate a new dataset. The estimated parameters based on the new dataset may then be useful for a more accurate prediction result by avoiding focusing only on the given dataset and thus alleviating the overfitting problem.

Put a pseudo data point y_{N+1}^p close to y_{N+1} and denote the set including it by $\mathbf{Z}_{N+1}^p = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_N, y_N), (\mathbf{x}_{N+1}, y_{N+1}^p)\}$. Then, the following holds [23]:

$$E[(y_{N+1} - \hat{f}(\mathbf{x}_{N+1}))^2 | \mathbf{Z}_{N+1}^p] \leq E[(y_{N+1} - \hat{f}(\mathbf{x}_{N+1}))^2 | \mathbf{Z}_N], \quad (7)$$

where $E[\cdot | \mathbf{A}]$ is the conditional expected value given $\mathbf{A}$.

In addition, among the various types of y_{N+1}^p , Kim et al. [23] reported that a sample mean of a given dataset usually results in the lowest prediction error. We use the sample mean for y_{N+1}^p in this study.

Least absolute shrinkage and selection operator. Finally, we apply the aCL method to LASSO to investigate whether it still holds in the regularization method. LASSO generally

provides a biased estimator due to penalization of the estimated coefficients [27]. Taking the previous example of a linear estimator used in (4)–(5), the LASSO method minimizes

$$\sum_{n=1}^N (y_n - \sum_{k=1}^m b_k x_{kn})^2 + \lambda \sum_{k=1}^m |b_k|, \quad (8)$$

where λ is the penalty (or tuning) parameter. 10-fold cross-validation is used to estimate λ in this study. The first term of (8) is the OLS method, which can possibly cause overfitting in a whole dataset, and the second term is the penalization term, which allows a predictive model to avoid the overfitting problem but leads to a biased estimator. Meanwhile, the aCL method uses a subsample of the given data; hence, it would not cause serious overfitting. In addition, based on this advantage, the penalization term would not shrink the estimated coefficients more than necessary, which alleviates the biased-estimator problem. Therefore, the aCL method is expected to reduce the prediction error when combined with LASSO.

Competing methods.

K-means clustering. The k-means clustering algorithm is a well-known and widely used method of partitioning data. Hence, we evaluated whether the aCL method can reduce the prediction error compared to k-means clustering.

In general, k-means clustering is used for performing classifications; however, because it is based on unsupervised learning, k-means clustering is also useful for making predictions on real-world data problems [27, 28]. Meanwhile, the problem of selecting the appropriate k value in an application persists. Suppose that Z_N is partitioned into k subsets, $S_1, S_2, \dots, S_k$. The k value is estimated as follows:

$$\hat{k} = \operatorname{argmin}_{k \in \mathbb{Z}^+} \sum_{i=1}^k \sum_{Z_n \in S_i} \|Z_n - \mu_i\|^2, \quad (9)$$

where μ_i is the mean of S_i .

Because the value of k is numerically determined in many cases [28], it is dependent on a type of data or on researcher knowledge. Moreover, the noise in a given dataset can easily lead to a small-sized cluster. Very small clusters result in high predictor variance, such that the prediction error increases beyond what is desirable [7]. Therefore, the aCL with a size between a last cluster and the whole dataset can result in lower prediction error than can k-means clustering.

The new sample averaged over bootstrap samples. As shown in Fig 3, the true function can be well approximated by bootstrap samples. Namely, a new sample averaged over the bootstrap samples can achieve a more accurate prediction performance than when simply using only one realized sample. Therefore, we need to evaluate the prediction performance based on the new sample averaged over the bootstrap samples compared to the aCL method.

Linear combination of two coefficients. As explained in the previous section, Clark and McCracken [12] suggested that the linear combination of two estimated regression coefficients improves the prediction performance, similar to the idea of the aCL method. However, in [12], only one realized sample was used, and a small number of change points were assumed. The aCL method is derived from using bootstrap samples and involves the last cluster among the multiple clusters in a given dataset; thus, the aCL method is expected to achieve a more accurate prediction result than the linear combination of two estimated regression coefficients.

Results and discussion

Simulation study

Simulation models. To examine whether the aCL method applies well to various functional forms, we conducted Monte Carlo simulations over four types of DGPs (100 Monte

Carlo simulations for each DGP). The aCL method helps to find the change point of $f(\mathbf{x})$, at which a whole sample is adequately partitioned so that a linear estimator fits well to the last cluster. Therefore, the DGPs having the change point would be good examples. DGP 1 is a univariate quadratic model, in which a change point exists at the local maximum. DGP 2 consists of two planes where the change points occur at the midpoint of a sample; thus the optimal location for partitioning data would be around the midpoint. DGP 3 is a hyperbolic paraboloid that resembles a saddle, in which a change point can be made from two directions. Thus, finding where to partition a given dataset is challenging. Finally, as a general example of DGP, we set DGP 4 to a multivariate polynomial function form. All the DGPs are estimated by a linear estimator as in (4)–(5):

$$\text{DGP 1 } y_n = -3(\theta_1 + \theta_2) - 2\frac{\theta_1}{\theta_2}x_{1n} + 2\frac{\theta_2}{\theta_1}x_{1n}^2 + e_n$$

$$\text{DGP 2 } y_n = \begin{cases} \frac{1}{\theta_1} - \frac{\theta_1}{\theta_2}x_{1n} + \frac{\theta_1}{\theta_2}x_{2n} + e_n & \text{for } n \leq \frac{N}{2} \\ \frac{1}{\theta_2} - \frac{\theta_2}{\theta_1}x_{1n} + \frac{\theta_2}{\theta_1}x_{2n} + e_n & \text{for } n > \frac{N}{2} \end{cases}$$

$$\text{DGP 3 } y_n = \frac{1}{\theta_1}x_{1n}^2 - \frac{1}{\theta_2}x_{2n}^2 + e_n$$

$$\text{DGP 4 } y_n = \frac{\theta_1}{\theta_2}x_{1n} + \frac{\theta_2}{\theta_1}x_{2n} + \theta_1x_{1n}x_{2n} + \theta_2x_{2n}^2 + e_n.$$

To consider the dynamic structure in each DGP, we let x_{mn} follow the AR(1) structure for $m = 1$ and 2; $x_{1n} = 0.5x_{1n-1} + e_{1n}$, $e_{1n} \sim N(0,1)$ and $x_{2n} = 0.7x_{2n-1} + e_{2n}$, $e_{2n} \sim N(0,1)$, respectively. Moreover, to observe the variations in the numerical results according to the changes in coefficients, we generated the coefficients according to [29],

$$\theta_k = \sqrt{\frac{p}{1-p}}\gamma_k, \quad (10)$$

$$\gamma_k = \{(k/3)^A(3-k/3)^B\} / \sum_{k=1}^2 \{(k/3)^A(3-k/3)^B\}^2,$$

where p takes the values $\{0.1, 0.2, \dots, 0.9\}$, and both A and B take the values $\{0.1, 0.2, \dots, 1\}$. The noise term e_n in each DGP follows an independent and identical normal distribution of zero mean and variances (denoted by Sig) of 1, 2, and 3. Additionally, we consider sample sizes (denoted by N) of 100, 200, and 300 and prediction horizons (denoted by H_0) of 1, 2, and 3. In brief, to test the prediction performance of the aCL method in a variety of situations, all DGPs were generated by Monte Carlo simulation with the independent variable of the AR(1) process, varying coefficients of DGP and variances of the noise term. In addition, we varied the sample size of DGP and prediction horizon to test the prediction performance of the proposed method.

Despite various simulation settings, these four DGP experiments may be insufficient to assert that the proposed method is always more accurate. For example, in the DGP consisting of only noise, a simple predictor of the sample mean would be a better choice; in contrast, in a DGP with no noise, 1-nearest-neighbor algorithm would achieve a good prediction result.

However, for easily visible DGPs such as these extremal cases, one can directly specify an estimator for a given dataset.

As a statistical test on the prediction results among the presented methods we applied the Diebold-Mariano test [30], which is used to test the null hypothesis that two methods have the same prediction accuracy. This test is useful in many applications because it can be used for non-Gaussian or serially correlated data.

Results on the simulated data. First, it would be helpful to see the existence of the last cluster for more accurate prediction using each DGP. From Fig 4, we can observe that prediction based on some last clusters results in lower prediction error than does prediction using the entire dataset. Nevertheless, there exist other last clusters that yield higher prediction error than using the whole dataset. Therefore, selecting a last cluster of adequate size is important and so using the aCL method can be useful for obtaining the last cluster.

Furthermore, Fig 5 depicts how the aCL is chosen and the predictions are made based on the aCL. For all the DGPs, the estimator based on the aCL more accurately indicates the prediction target than does the estimator based on the entire dataset.

Tables 1–4 show the prediction errors of the methods used in this study. The first nine rows in the table present the results when the aCL method is applied to OLS, LSPD, and LASSO. The last six rows show the prediction errors of the competing methods (k-means clustering, the new sample obtained from bootstrap samples, and the linear combination of two estimated regression coefficients). Additionally, the tables present the changes of prediction errors according to the different sample sizes (N), horizons (H_0), and variances (Sig). The abbreviations in the tables are as follows. OLS = ordinary least squares, OLSaCL = OLS combined with aCL, LSPD = least squares adjusted with pseudo data, LSPDaCL = LSPD combined with aCL, LASSO = least absolute shrinkage and selection operator, LASaCL = LASSO combined with aCL, Kmeans = k-means clustering, Bootstr = the new sample obtained from bootstrap samples, and Comb = the linear combination of two estimated regression coefficients.

The results in Tables 1–4 indicate that the three aCL methods outperform their preceding optimization methods under different conditions. Considering the type of DGP, the prediction errors are relatively lower in the curved functional forms (DGP 1, 3, and 4) compared to DGP 2, which comprises two planes. Additionally, when Sig is large in some cases, the aCL method shows relatively low prediction performance. Regarding the prediction errors of the three competing methods, we also observe that the aCL method mainly shows lower prediction errors than those of the three competitors. It is quite evident that the aCL method yields lower prediction errors than do Bootstr and Comb. Meanwhile, k-means clustering yields good prediction results when $N = 300$ in DGP 2, where the effect of noise is relatively low, and so k-means clustering can work well [31].

In addition, because the size of the aCL is between the whole dataset and a last cluster, the variance of prediction error when using the aCL method is lower than that when using k-means clustering, as presented in Table 5, and so the aCL method can help to obtain a stable prediction performance.

To demonstrate the robustness of the aCL method under various conditions, boxplots of the prediction errors are presented according to N , H_0 , Sig , and coefficients. First, changes of the sample size N have little influence on the performance of the aCL method compared to its preceding optimization method and competing methods, as seen in Fig 6. As N increases, the aCL method consistently shows lower prediction errors than the other methods. K-means clustering achieves the second-lowest prediction errors in many cases but sometimes causes very high prediction errors.

As in the previous example, the aCL method outperforms the other methods for all the prediction horizons, as shown in Fig 7. A longer prediction horizon should affect the prediction

Fig 4. Change of the prediction error based on the size of the last cluster. Prediction based on the last cluster (cluster, dashed line) can yield lower prediction error than does linear regression over the entire dataset (LR, solid line). Because the full dataset is used for LR, the PE does not change in LR. For each DGP, $N = 100$.

<https://doi.org/10.1371/journal.pone.0223529.g004>

ability of a prediction model more than a shorter prediction horizon, and so we can observe that prediction errors increase as H_0 increases. In particular, k-means clustering shows high prediction errors when H_0 is high.

A high variance of noise generally degrades prediction ability. Because the size of a cluster is smaller than the size of the full dataset, methods using the cluster would be considerably influenced by the size of the variance. In Fig 8, we can see that the prediction performance of

Fig 5. Selection of aCL and its prediction. DGP 1 shows how aCL (square) is selected among the data (white circle) and that the fitted line of aCL (aCL-fit, dashed line) more accurately indicates the prediction target (target, black circle) than does the linear regression fitted line of a whole dataset (LR-fit, solid line). DGP 2–4 also show that aCL is well selected in three-dimensional space, and the fitted plane of aCL (aCL-fit, dark green plane) more accurately indicates the prediction target than does the linear regression fitted plane of the whole dataset (LR-fit, light gray plane). For each DGP, $N = 100$, $H_0 = 1$, and $\text{Sig} = 1$.

<https://doi.org/10.1371/journal.pone.0223529.g005>

the aCL method decreases as the variance increases. In particular, k-means clustering produces very poor prediction performances in cases with increasing variance.

In Fig 9, we present the prediction errors resulting from changes of coefficients. Similar to previous experiments, the aCL methods achieve lower prediction errors than do the other methods. The prediction errors of k-means clustering also seem low in some cases; however, its prediction performance appears to fluctuate depending on the change of coefficients.

Estimation of change point. This section discusses whether a change point of a DGP is well approximated by the aCL method when compared to k-means clustering. Because a change point is distinctly set as half the sample size in DGP 2, we use the data generated by

Table 1. Comparison of prediction error in DGP 1.

Sig	N = 100						N = 200						N = 300					
	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3
OLS	245.019	348.978	342.525	387.463	392.13	375.065	274.54	421.606	31.435	27.26	13.484	26.915	44.693	29.793	27.662	460.07	482.527	419.929
(p-value)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)
LSPD	246.171	346.137	340.285	381.899	385.778	370.38	271.444	412.312	255.431	322.563	446.152	347.975	502.971	364.028	461.974	484.368	420.827	480.497
(p-value)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)
LASSO	243.205	341.971	339.174	384.461	389.413	372.398	272.355	418.626	253.913	318.275	441.989	344.245	500.569	361.098	457.155	479.373	407.125	476.788
(p-value)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)
Kmeans	31.493	57.695	56.587	57.589	73.917	65.173	42.737	75.498	51.922	49.992	61.279	61.051	82.013	83.619	60.22	84.505	74.83	73.919
(p-value)	(0.002)	(<0.001)	(<0.001)	(0.001)	(<0.001)	(0.002)	(0.008)	(0.003)	(0.014)	(0.002)	(0.008)	(0.001)	(0.033)	(<0.001)	(0.003)	(0.004)	(<0.001)	(0.094)
Boostr	242.343	346.982	341.665	388.084	390.902	374.487	276.42	421.122	259.228	321.503	445.978	348.669	506.693	363.502	463.408	484.174	422.316	480.895
(p-value)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)
Comb	176.012	274.632	282.323	324.494	304.418	292.052	205.806	363.356	216.887	200.121	323.458	241.346	409.515	260.936	341.701	331.195	325.951	347.176
(p-value)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)

NOTE: The three optimization methods (OLS, LSPD, and LASSO) are compared with their aCL methods. Other competing methods (Kmeans, Boostr, and Comb) are compared with OLSaCL. The p-value of the Diebold-Mariano test is shown in parentheses.

<https://doi.org/10.1371/journal.pone.0223529.t001>

Table 2. Comparison of prediction error in DGP 2.

Ho	N = 100						N = 200						N = 300					
	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3
Sig.																		
OLS	1.312	1.262	3.423	2.445	2.055	5.018	1.567	2.245	6.831	4.172	7.628	11.988	4.761	8.751	21.282	5.275	10.754	16.625
OLS-CL	0.313	1.162	3.469	0.633	2.153	3.01	0.675	2.324	4.422	1.074	1.965	5.088	1.418	2.741	7.993	1.889	6.768	10.178
(p-value)	(<0.001)	(0.4)	(0.599)	(0.016)	(0.548)	(0.084)	(0.002)	(0.55)	(0.122)	(0.002)	(0.004)	(0.014)	(<0.001)	(<0.001)	(0.004)	(0.004)	(0.029)	(0.001)
LSPD	2.421	2.595	4.819	3.518	3.578	7.047	2.751	3.752	8.743	4.336	7.79	12.044	4.956	8.903	21.75	5.579	10.369	16.046
(p-value)	0.571	1.661	4.098	1.028	2.754	3.522	0.933	3.275	5.662	0.784	2.409	5.454	1.385	2.423	8.857	1.846	5.911	10.426
LSPD-CL	0.571	1.661	4.098	1.028	2.754	3.522	0.933	3.275	5.662	0.784	2.409	5.454	1.385	2.423	8.857	1.846	5.911	10.426
(p-value)	(<0.001)	(0.039)	(0.302)	(0.004)	(0.2)	(0.021)	(<0.001)	(0.262)	(0.114)	(<0.001)	(0.008)	(0.015)	(0.001)	(0.001)	(0.006)	(0.003)	(0.007)	(0.003)
LASSO	1.287	1.175	3.27	2.073	2.277	5.035	1.477	1.958	6.175	3.96	6.821	10.83	4.518	8.244	20.203	5.162	10.429	15.718
(p-value)	0.424	1.354	4.099	0.803	2.369	2.036	0.531	3.048	3.2	0.989	1.714	4.673	1.526	2.721	8.551	1.922	7.845	13.449
LASSO-CL	0.424	1.354	4.099	0.803	2.369	2.036	0.531	3.048	3.2	0.989	1.714	4.673	1.526	2.721	8.551	1.922	7.845	13.449
(p-value)	(<0.001)	(0.406)	(0.563)	(0.034)	(0.356)	(0.018)	(0.001)	(0.462)	(0.08)	(0.001)	(0.003)	(0.016)	(<0.001)	(<0.001)	(0.006)	(0.005)	(0.043)	(0.001)
Kmeans	1.108	2.673	7.525	1.201	16.642	11.836	1.265	8.266	19.091	1.682	4.908	17.438	1.936	5.385	25.424	2.862	11.506	19.876
(p-value)	(0.001)	(0.051)	(0.014)	(0.105)	(0.078)	(<0.001)	(0.004)	(0.009)	(0.001)	(0.096)	(0.012)	(0.017)	(0.171)	(0.047)	(0.002)	(0.157)	(0.116)	(0.039)
Boostr	1.308	1.314	3.334	2.562	1.9	4.72	1.673	1.971	6.622	4.16	7.817	11.992	4.927	8.852	21.006	5.52	10.931	16.743
(p-value)	(<0.001)	(0.349)	(0.638)	(0.016)	(0.625)	(0.11)	(0.001)	(0.699)	(0.157)	(0.001)	(0.005)	(0.014)	(<0.001)	(0.001)	(0.005)	(0.003)	(0.024)	(0.001)
Comb	0.895	1.201	3.818	1.104	2.565	5.613	0.948	2.906	9.575	2.702	6.005	13.179	3.284	6.805	20.8	4.098	8.631	16.924
(p-value)	(0.002)	(0.461)	(0.457)	(0.128)	(0.335)	(0.023)	(0.272)	(0.164)	(0.013)	(0.009)	(0.005)	(0.022)	(0.019)	(0.003)	(0.006)	(0.017)	(0.177)	(0.015)

<https://doi.org/10.1371/journal.pone.0223529.t002>

Table 3. Comparison of prediction error in DGP 3.

Sig	N = 100						N = 200						N = 300					
	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3
OLS	11.451	14.406	20.64	12.451	17.555	24.14	11.901	17.448	22.579	11.881	11.586	19.696	12.124	16.382	27.168	13.35	13.871	17.187
OLS+CL	1.364	2.792	6.841	2.442	4.825	6.806	2.26	5.097	9.574	2.224	4.368	6.184	2.595	7.312	13.042	1.735	7.756	2.574
(p-value)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)
LSPD	11.537	14.523	20.668	12.408	17.518	23.999	11.818	17.327	22.339	11.916	11.621	19.651	12.141	16.457	27.106	13.382	13.879	17.185
LSPD+CL	1.933	3.41	7.432	3.106	5.221	7.171	2.75	5.396	9.798	2.611	4.604	6.442	2.95	7.395	13.389	2.101	4.917	2.948
(p-value)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)
LASSO	10.674	13.425	20.546	11.735	19.472	25.508	10.179	15.338	20.469	11.643	12.053	19.365	12.089	16.668	25.719	13.541	11.842	17.867
LASSO+CL	1.463	2.781	5.935	2.982	4.056	8.66	2.494	5.447	7.367	2.246	4.227	5.356	2.811	8.88	13.62	2.023	4.381	8.616
(p-value)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)
Kmeans	2.563	7.893	10.698	3.343	8.103	16.749	4.24	20.094	18.19	4.522	8.317	18.469	4.543	15.137	26.893	3.158	7.748	11.102
(p-value)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)
Boostr	11.041	14.159	20.696	12.14	16.87	22.999	11.782	17.205	22.1	11.971	11.623	19.586	12.064	16.029	27.127	13.919	14.261	18.269
(p-value)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)
Comb	10.029	12.552	15.673	10.116	11.488	18.54	10.705	16.353	20.803	10.821	11.194	18.972	11.045	15.652	26.992	10.366	12.985	16.602
(p-value)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)

<https://doi.org/10.1371/journal.pone.0223529.t003>

Table 4. Comparison of prediction error in DGP 4.

Sig	DGP = 4						N = 100						N = 200						N = 300					
	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3
OLS	7.727	7.565	7.016	8.059	7.505	6.873	9.74	12.159	8.514	17.703	21.741	27.373	17.958	25.376	29.965	18.701	24.961	31.936	25.405	34.045	47.398	25.205	35.533	52.104
OLS-CL	0.752	1.609	3.186	1.346	2.568	4.513	1.524	3.971	5.051	1.856	4.441	7.466	2.39	4.689	10.849	2.818	6.558	14.001	1.915	6.338	11.552	2.097	4.5	13.422
(p-value)	<(0.001)	<(0.001)	(0.001)	<(0.001)	<(0.001)	(0.033)	<(0.001)	<(0.001)	(0.053)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)
LSPD	7.645	7.669	7.386	8.23	7.559	7.11	9.793	12.046	8.912	17.89	21.194	27.522	17.928	25.051	30.201	18.731	24.976	32.115	25.697	33.569	47.688	25.52	35.904	51.636
LSPD-CL	1.492	2.258	3.612	2.436	2.929	4.545	2.744	4.795	5.728	2.473	5.033	7.928	2.912	5.367	11.446	3.164	7.062	14.802	2.673	6.921	12.695	2.764	4.703	14.555
(p-value)	<(0.001)	<(0.001)	(0.001)	<(0.001)	<(0.001)	(0.05)	<(0.001)	(0.001)	(0.086)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)
LASSO	7.798	8.409	7.049	8.028	7.566	7.109	9.745	12.414	6.997	17.788	21.601	27.361	18.011	25.394	30.045	18.797	24.946	33.342	25.533	34.16	47.754	25.223	35.593	52.959
LASSO-CL	0.888	1.239	2.863	1.387	2.459	5.283	1.716	3.931	5.214	1.88	4.459	6.735	2.416	5.624	12.326	3.382	6.999	14.528	1.909	5.764	11.203	2.321	4.76	13.978
(p-value)	<(0.001)	<(0.001)	(0.001)	<(0.001)	<(0.001)	(0.058)	<(0.001)	<(0.001)	(0.083)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)
Kmeans	1.822	4.847	13.545	2.355	8.571	16.997	2.598	6.659	18.116	4.42	7.092	21.05	4.729	8.856	25.58	6.308	14.677	26.239	4.727	10.346	20.862	4.838	11.242	22.267
(p-value)	<(0.001)	<(0.001)	<(0.001)	(0.009)	<(0.001)	<(0.001)	(0.008)	(0.096)	<(0.001)	<(0.001)	(0.002)	<(0.001)	(0.096)	(0.002)	<(0.001)	<(0.001)	<(0.001)	(0.002)	<(0.001)	(0.050)	(0.094)	(0.001)	<(0.001)	(0.072)
Boostr	7.654	7.491	7.192	8.057	7.487	7.006	9.63	11.909	8.531	17.907	21.882	27.061	18.877	25.895	29.923	18.877	25.004	32.182	26.497	35.249	49.32	26.247	37.054	53.803
(p-value)	<(0.001)	<(0.001)	(0.001)	<(0.001)	<(0.001)	(0.025)	<(0.001)	<(0.001)	(0.05)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)
Comb	5.273	5.866	6.986	6.253	6.683	8.126	7.583	7.632	8.257	15.247	16.04	24.833	15.289	19.234	28.538	15.627	20.16	29.114	18.257	31.635	40.538	18.814	28.54	50.077
(p-value)	<(0.001)	<(0.001)	(0.001)	<(0.001)	<(0.001)	(0.009)	<(0.001)	(0.001)	(0.037)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)	<(0.001)

<https://doi.org/10.1371/journal.pone.0223529.t004>

Table 5. Variance of prediction errors by the aCL method and k-means clustering.

	DGP 1	DGP 2	DGP 3	DGP 4
OLSaCL	13.884	3.680	3.047	5.964
Kmeans	14.875	7.365	6.344	7.438

NOTE: Prediction errors from Tables 1–4 are used

<https://doi.org/10.1371/journal.pone.0223529.t005>

DGP 2. For the robustness check, the change point is estimated across the various values of N , H_0 , and Sig . Well-estimated change points should be approximately half of the sample size. As shown in Table 6, the change points estimated by the aCL method are close to the true change points, whereas the change points estimated by k-means clustering are downward-biased because k-means clustering usually produces a small cluster size by the effect of the noise.

Fig 10 depicts the histograms of estimated change points by OLSaCL and Kmeans through 100 Monte Carlo iterations. We can see that the histogram of OLSaCL shows a central tendency similar to a normal distribution, unlike Kmeans.

Real data study

Stock market index. As the first application using real data, the S&P 500 index is predicted by the USD/EUR exchange rate and interest rate (effective federal funds rate in the US). Financial data have been the subject of numerous studies. In particular, the relation between stock market index and the exchange rate has long been studied [32, 33]. Additionally, the interest rate is known to affect the stock market because it is usually considered as a substitute for stock market returns [34].

More recently, machine learning methods such as the nearest neighbor algorithm have been applied to predict the stock market index [35], the exchange rate [36], and the interest rate [37] by focusing on the nonlinear characteristics of financial data. Meanwhile, most financial data have stylized facts that make prediction difficult [38], such as the fat tail distribution, clustering volatility, etc. These financial data characteristics frequently result in unpredictable spikes or clusters in time-series data which degrade the prediction power of a model using the whole dataset. Therefore, the nearest neighbor algorithm tends to be much influenced by the outliers leading to difficulty in establishing the predictive model, which might bring about the varying prediction performances depending on the range of the dataset [37]. Therefore, partitioning a dataset can be effective for these types of data. Some works reported improved prediction performances by adopting the k-means clustering method [39, 40, 41]. However, most of the related works estimate the number of clusters numerically and directly use the obtained clusters. So one may be able to test the many numbers of clusters until the desirable result is obtained [41], which makes the size of the cluster somewhat unreliable. On the other hand, because the aCL method applies the bootstrap method, it is expected to yield more reliable results.

We adopted a dataset comprising a total of 250 weeks of data from Sep. 30, 2011 to Sep. 23, 2016. To control the effect of the data size, the dataset is divided into six equal-sized subsets. Each subset includes 100 instances, from which the last 30 are predicted. The first-order difference data are used as in the usual financial data analysis [42].

Given a dataset of size N , we predict the values at $N+1$, $N+2$ and $N+3$. As seen in Table 7, the aCL methods largely outperform their preceding optimizations and competing methods. K-means clustering yields very poor prediction performances in several cases, for example, in the third subset; similarly, the prediction performance of the aCL method is relatively low in the third subset compared to others. Note that standard deviations in prediction errors of the aCL

Fig 6. Prediction error comparison by sample size ($H_0 = 2, \text{Sig} = 2$).

<https://doi.org/10.1371/journal.pone.0223529.g006>

Fig 7. Prediction error comparison by horizon (N = 200, Sig = 2).

<https://doi.org/10.1371/journal.pone.0223529.g007>

Fig 8. Prediction error comparison by variance (N = 200, Ho = 2).

<https://doi.org/10.1371/journal.pone.0223529.g008>

Fig 9. Prediction error comparison by changes of coefficients.

<https://doi.org/10.1371/journal.pone.0223529.g009>

Table 6. Change points estimated by OLSaCL and Kmeans (DGP 2).

Factor	Value	OLSaCL	Kmeans
N (Ho = Sig = 1)	100	44.5	18.8
	200	100.5	25.2
	300	147.2	48.7
Ho (N = 100, Sig = 1)	1	41.3	18.2
	2	47.1	6.1
	3	50.4	1.5
Sig (N = 100, Ho = 1)	1	43.1	6.4
	2	40.5	24.0
	3	45.0	18.7

NOTE: The change point is the average over Monte Carlo iterations

<https://doi.org/10.1371/journal.pone.0223529.t006>

methods over all the prediction horizons are lower than other methods (Fig 11), which implies the low variance of the prediction error in the aCL methods.

Home price index. The home price index (S&P/Case-Shiller U.S. National Home Price Index) is predicted by using the mortgage rate (30-year fixed rate mortgage average in the US) and the expected inflation (survey by University of Michigan).

A total of 250 monthly data points from Feb. 1990 to Dec. 2015 are included, and the data are divided into the six equal subsets as in the previous example. The last 30 instances out of 100 instances in each subset are predicted, and the prediction horizon ranges from 1 to 3.

There exists a vast literature predicting home prices. Many studies on this topic adopt time-series analysis tools [43, 44]. With regard to the factors influencing home price, the expected inflation can have a positive effect on home prices because the inflation rate usually reflects home prices [45]. However, a rise in inflation results in an increase in interest rates and a corresponding reduction in home prices [46]. Hence, the expected inflation has different influences on home prices depending on the relevant economic environment. In general, the mortgage rate is known to have a negative influence on home prices [47, 48]. As the time-series model enables achieving a good prediction performance, the selection of the lag parameter in the time-series model still remains a factor that is not easy to be determined. Moreover, as previous studies [43, 44] adopted the method of combining the different models in order to obtain the stable prediction performance, there also still remains the problem about how to combine models or which models to choose. Therefore, a clustering algorithm could be an alternative for the prediction problem when it is not easy to determine a comprehensive time-series model explaining the whole dataset.

Similar to the case of the stock market index, we can also check the low standard deviations of prediction errors of the aCL methods in Fig 12.

Table 8 shows that the aCL methods mainly achieve good prediction performances; however, in the fourth and sixth subsets, the aCL method is unable to yield good prediction performances and the k-means clustering also results in high prediction errors. For this phenomenon, a clue can be found in the heat maps in Fig 13, which show the dissimilarity matrix of the instances from the fourth to the sixth subsets. In the fifth subset, where the aCL method results in low prediction error, the instances from approximately the 50th to 100th appear well-clustered according to the dissimilarity level; thus, the prediction targets from the 70th to the 100th instances can be predicted well by the aCL methods. In contrast, in the fourth and

Fig 10. Histograms of change points estimated by OLSaCL and Kmeans.

<https://doi.org/10.1371/journal.pone.0223529.g010>

Table 7. Comparison of prediction error (S&P 500 index).

Subset	1			2			3			4			5			6		
	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3
Ho																		
OLS	1,623.641	1,456.661	1,583.565	1,006.735	1,044.876	795.876	5,150.994	5,351.771	5,600.641	4,570.235	4,343.653	4,392.129	5,056.139	4,986.251	4,904.958	5,006.36	4,930.956	4,933.17
OLSaCL	529.096	424.337	243.216	553.64	333.66	357.637	610.101	196.099	266.285	2,213.057	1,671.478	1,543.156	950.783	895.095	548.783	658.434	827.888	1,213.569
(p-value)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)
LSPD	1,542.055	1,392.232	1,494.012	1,110.979	1,149.076	935.288	4,816.755	4,994.519	5,216.836	3,933.531	3,770.43	3,783.216	4,315.94	4,268.341	4,188.265	4,197.11	4,142.642	4,132.846
LSPDaCL	522.096	448.278	292.519	633.418	419.701	459.249	296.197	85.188	125.423	1,542.501	1,196.477	1,049.984	664.488	644.935	338.042	450.153	599.778	866.493
(p-value)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)
LASSO	1,389.534	1,257.328	1,354.365	777.932	803.751	596.907	3,980.313	4,116.747	4,200.481	4,428.887	4,716.819	4,279.51	4,713.972	4,792.391	4,648.729	6,022.143	5,743.751	5,356.878
LASaCL	640.095	324.586	191.067	479.821	291.952	273.487	528.755	171.586	203.63	1,982.531	3,512.937	1,270.665	2,552.292	912.82	690.8	2,249.648	2,826.237	1,796.834
(p-value)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(0.167)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(0.099)
Kmeans	2,106.306	1,528.799	1,642.423	896.124	1,432.912	196.805	687.919	935.397	2,028.618	8,666.065	8,603.418	7,978.947	5,706.817	5,039.112	5,268.619	4,690.76	4,681.7	4,832.931
(p-value)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(1)	(0.327)	(0.03)	(0.004)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)
Bootstr	1,627.618	1,459.733	1,534.486	964.687	959.824	711.141	5,115.023	5,308.184	5,607.416	5,142.519	4,557.184	4,470.688	5,229.428	5,405.18	5,419.757	5,263.469	4,759.555	5,259.559
(p-value)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)
Comb	1,626.753	1,457.796	1,584.437	1,380.856	1,475.762	1,351.905	4,892.685	5,209.421	5,214.018	1,915.456	1,845.557	1,836.637	2,128.457	2,083.357	2,070.776	1,731.392	1,700.678	1,706.512
(p-value)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(0.91)	(0.01)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(0.001)	(0.148)

<https://doi.org/10.1371/journal.pone.0223529.t007>

Fig 11. Comparison of standard deviations of prediction errors over the prediction horizons.

<https://doi.org/10.1371/journal.pone.0223529.g011>

sixth subsets, the instances around the prediction targets do not show well-clustered shapes; thus, we can conclude that methods using the clustering algorithm do not work well at these subsets.

Fig 12. Comparison of standard deviations of prediction errors over the prediction horizons.

<https://doi.org/10.1371/journal.pone.0223529.g012>

Table 8. Comparison of prediction error (Home price index).

Subset	1			2			3			4			5			6		
	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3
Ho																		
OLS	7.828 (0.008)	10.981 (0.125)	13.477 (0.489)	104.179 (0.001)	122.637 (0.001)	159.995 (0.001)	535.21 (0.001)	675.131 (0.001)	860.787 (0.001)	27.34 (0.258)	22.612 (0.934)	27.207 (0.999)	405.931 (0.001)	433.311 (0.001)	477.964 (0.001)	72.431 (0.216)	59.753 (0.964)	68.189 (0.724)
OLSaCL	1.415 (0.008)	3.605 (0.125)	13.186 (0.489)	14.503 (0.001)	59.207 (0.001)	88.479 (0.001)	16.915 (0.001)	89.869 (0.001)	156.927 (0.001)	22.389 (0.258)	37.38 (0.934)	59.258 (0.999)	35.495 (0.001)	36.581 (0.001)	27.162 (0.001)	63.628 (0.216)	139.413 (0.964)	99.749 (0.724)
LSPD	12.156 (0.008)	16.989 (0.125)	19.974 (0.489)	77.154 (0.001)	88.39 (0.001)	113.302 (0.001)	402.313 (0.001)	491.3 (0.001)	607.367 (0.001)	9.664 (0.258)	14.131 (0.934)	17.752 (0.999)	330.297 (0.001)	348.942 (0.001)	380.806 (0.001)	75.899 (0.216)	63.338 (0.964)	71.756 (0.724)
LSPDaCL	3.056 (0.001)	2.571 (0.009)	0.64 (0.027)	20.338 (0.001)	52.732 (0.001)	65.159 (0.001)	26.75 (0.001)	85.028 (0.001)	125.856 (0.001)	3.74 (0.014)	4.397 (0.006)	10.576 (0.134)	26.787 (0.001)	26.469 (0.001)	18.86 (0.001)	74.36 (0.442)	139.601 (0.971)	106.155 (0.762)
LASSO	6.664 (0.011)	9.908 (0.066)	12.219 (0.398)	106.23 (0.001)	123.888 (0.001)	159.673 (0.001)	585.304 (0.001)	728.003 (0.001)	961.346 (0.001)	23.275 (0.258)	20.403 (0.934)	24.667 (0.999)	413.923 (0.001)	437.733 (0.001)	477.003 (0.001)	79.21 (0.832)	64.433 (0.832)	76.155 (0.907)
LaSaCL	1.274 (0.011)	2.98 (0.066)	9.705 (0.398)	33.559 (0.001)	78.001 (0.001)	87.183 (0.001)	24.095 (0.001)	129.964 (0.001)	233.149 (0.001)	20.15 (0.258)	30.896 (0.934)	43.616 (0.999)	82.133 (0.001)	48.193 (0.001)	26.764 (0.001)	90.636 (0.832)	201.612 (0.832)	148.198 (0.907)
Kmeans	17.135 (0.001)	28.166 (0.001)	65.534 (0.001)	25.309 (0.001)	38.026 (0.001)	89.703 (0.001)	52.252 (0.025)	120.376 (0.235)	302.769 (0.001)	97.486 (0.005)	121.879 (0.01)	184.943 (0.053)	4.441 (0.001)	4.316 (0.001)	1.604 (0.001)	218.579 (0.001)	157.704 (0.329)	296.049 (0.001)
Bootstr	9.311 (0.006)	14.371 (0.093)	13.942 (0.473)	118.511 (0.001)	136.676 (0.001)	171.779 (0.001)	582.779 (0.001)	758.247 (0.001)	993.253 (0.001)	29.41 (0.191)	24.612 (0.896)	29.126 (0.998)	396.858 (0.001)	421.812 (0.001)	466.379 (0.001)	72.878 (0.203)	61.841 (0.96)	67.481 (0.733)
Comb	7.795 (0.008)	10.854 (0.127)	13.35 (0.494)	99.291 (0.001)	117.107 (0.001)	155.867 (0.001)	516.993 (0.001)	657.46 (0.001)	858.033 (0.001)	27.171 (0.26)	22.242 (0.942)	27.019 (0.999)	403.307 (0.001)	429.392 (0.001)	475.835 (0.001)	74.532 (0.165)	60.194 (0.965)	68.216 (0.726)

<https://doi.org/10.1371/journal.pone.0223529.t008>

Fig 13. Heat maps of the dissimilarity matrix of instances from the fourth to the sixth subset. The dissimilarity metric is the Euclidean distance between the instances.

<https://doi.org/10.1371/journal.pone.0223529.g013>

Conclusions

Partitioning data produces a more accurate prediction than using a whole dataset when a true function is not known or when setting an estimator for the whole dataset is difficult. Meanwhile, the small-sized cluster produced by simply partitioning data causes high variance problems, which leads to high prediction errors. In this study, we showed that adjusting the size of the last cluster avoids high prediction error. To adjust the size of the last cluster between the whole dataset and the last cluster, we applied the bootstrap method, which has the effect that a given model is trained multiple times by different bootstrapped samples generated from a certain DGP; hence, the model can estimate the reliable location of the last cluster that is optimal for prediction. Therefore, we believe that this paper contributes to understanding that the adjusted last cluster could be optimal for prediction based on the balance between bias and variance by the bootstrap method.

In this paper, the aCL method was shown to reduce prediction errors using the numerical results of both simulated and real data. Notice that the aCL method is not for establishing a complete model fitting a whole dataset but for selecting an optimal subsample for prediction. Therefore, an estimator of simple functional form such as a linear function can be easily used with the aCL method for prediction, and this advantage of the aCL method would serve a practical use in research or application. In addition, the aCL method yielded the size-adjusted last cluster by the bootstrap method, which is easy to be adopted in studies using a clustering algorithm, and thus produces a more reliable subsample for prediction.

Nevertheless, this study has some limitations. The datasets were numerically partitioned, which usually leads to the problem of time consumption that occurs with other clustering algorithms. As described in the second real data example, for the subsamples not suitable for the clustering algorithm, the k-means clustering yielded very high prediction errors; hence, the aCL method was also unable to achieve a good prediction performance. To solve this problem, a more elaborate technique needs to be developed. However, observing the degree of clustering in data before using a clustering algorithm could be helpful to avoid this problem. In addition, as future work, a theoretical study on the aCL method should be done to rigorously prove the reliability of the proposed method.

In summary, we proposed a new method that adjusts the size of a last cluster to achieve accurate prediction. By applying the bootstrap method to find the most reliable aCL, this method could yield low prediction errors. More specifically, the aCL method can be useful in many applications using clustering algorithms for a stable prediction performance.

Acknowledgments

I would like to express deep gratitude to Professor Jaehwan Kim for his constructive suggestions and useful critique of this study.

Author Contributions

Conceptualization: Jeongwoo Kim.

Data curation: Jeongwoo Kim.

Formal analysis: Jeongwoo Kim.

Software: Jeongwoo Kim.

Visualization: Jeongwoo Kim.

Writing – original draft: Jeongwoo Kim.

Writing – review & editing: Jeongwoo Kim.

References

1. Dietterich TG, Kong EB. Machine learning bias, statistical bias, and statistical variance of decision tree algorithms. Technical report, Department of Computer Science, Oregon State University, 1995. Available from: <http://www.cems.uwe.ac.uk/~irjohnso/coursenotes/uqc832/tr-bias.pdf>.
2. Bennett PN. Neighborhood-based local sensitivity: In European Conference on Machine Learning; 2007 Sep: Springer, Berlin, Heidelberg; 2007. 30–41.
3. Bubeck S, Von Luxburg U. Overfitting of clustering and how to avoid it. Preprint. 2007. 1–39. Available from: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.527.6339&rep=rep1&type=pdf>.
4. Ernst J, Nau G. J, Bar-Joseph Z. Clustering short time series gene expression data. *Bioinformatics*. 2005; 21(suppl_1):i159–i168.
5. Thrun M. C. Projection-based clustering through self-organization and swarm intelligence: combining cluster analysis with the visualization of high-dimensional data: Springer; 2018. pp. 151–152.
6. Pruessner G. Self-organised criticality: theory, models and characterisation. Cambridge University Press; 2012. p. 218.
7. Hastie T, Tibshirani R, Friedman J. The Elements of Statistical Learning: Data Mining, Inference, and Prediction: Springer New York; 2013. p. 38, 199.
8. Oh KJ, Kim K-j. Analyzing stock market tick data using piecewise nonlinear model. *Expert Syst Appl*. 2002; 22(3):249–255.
9. Bishop C, Bishop CM. Neural networks for pattern recognition: Oxford university press; 1995. p. 333.
10. Elliott G. Forecasting when there is a single break. Manuscript, University of California at San Diego. 2005. Available from: <http://www.uh.edu/~cmurray/TCE/papers/Elliott.pdf>.
11. Pesaran MH, Timmermann A. Selection of estimation window in the presence of breaks. *J Econometrics*. 2007; 137(1):134–161.
12. Clark TE, McCracken MW. Improving forecast accuracy by combining recursive and rolling forecasts. Federal Reserve Bank of Kansas City, 2004. Available from: <https://files.stlouisfed.org/files/htdocs/wp/2008/2008-028.pdf>.
13. Horowitz JL. The bootstrap. *Handbook of econometrics*. 5: Elsevier; 2001. p. 3163.
14. Breiman L. Bias, variance, and arcing classifiers. 1996. Available from: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.115.7931&rep=rep1&type=pdf>.

15. Smith RS, Windeatt T, editors. The bias variance trade-off in bootstrapped error correcting output code ensembles: International Workshop on Multiple Classifier Systems; 2009: Springer, Berlin, Heidelberg; 2009. 1–10.
16. Bauer E, Kohavi R. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Mach Learn.* 1998; 36(1–2):105–139.
17. Fumera G, Roli F, Serrau A, editors. Dynamics of variance reduction in bagging and other techniques based on randomization: International Workshop on Multiple Classifier Systems; 2005: Springer, Berlin, Heidelberg; 2005. 316–325.
18. Hassanien AE, Oliva DA. *Advances in Soft Computing and Machine Learning in Image Processing*: Springer International Publishing; 2017. p. 121.
19. Kim JW, Kim JC, Kim JH. Adjusted k-nearest neighbor algorithm. *J. Korean Soc. of Marine Engineering.* 2018; 42(2):127–135.
20. Diebold FX, Chen C. Testing structural stability with endogenous breakpoint a size comparison of analytic and bootstrap procedures. *J Econometrics.* 1996; 70(1):221–241.
21. Goeman JJ. L1 penalized estimation in the Cox proportional hazards model. *Biom J.* 2010; 52(1):70–84. <https://doi.org/10.1002/bimj.200900028> PMID: 19937997
22. Greene WH. *Econometric Analysis*: Pearson Education, India; 2003. p. 193.
23. Kim JW, Kim JC, Kim JT, Kim JH. Forecasting greenhouse gas emissions in the Korean shipping industry using the least squares adjusted with pseudo data. *J. Korean Soc. of Marine Engineering.* 2017; 41(5):452–460.
24. Schuster EF. Incorporating support constraints into nonparametric estimators of densities. *Commun Stat Theory Methods.* 1985; 14(5):1123–1136.
25. Cowling A, Hall P. On pseudodata methods for removing boundary effects in kernel density estimation. *J R Stat Soc Series B Stat Methodol.* 1996:551–563.
26. Breiman L. Using convex pseudo-data to increase prediction accuracy. *breast (Wis).* 1998; 699(9):2.
27. Zou H. The adaptive lasso and its oracle properties. *J Amer Statistical Assoc.* 2006; 101(476):1418–1429.
28. Pham DT, Dimov SS, Nguyen CD. Selection of K in K-means clustering. *Proc. Inst. Mech. Eng. Pt. C J. Mechan. Eng. Sci.* 2005; 219(1):103–119.
29. Hansen BE. Least-squares forecast averaging. *J Econometrics.* 2008; 146(2):342–350.
30. Diebold FX, Mariano RS. Comparing predictive accuracy. *J Bus Econ Stat.* 2002; 20(1):134–144.
31. Tseng GC. Penalized and weighted K-means for clustering with scattered objects and prior information in high-throughput biological data. *Bioinformatics.* 2007; 23(17):2247–2255. <https://doi.org/10.1093/bioinformatics/btm320> PMID: 17597097
32. Ma CK, Kao GW. On exchange rate changes and stock price reactions. *J Bus Finan Account.* 1990; 17(3):441–449.
33. Kwon CS, Shin TS. Cointegration and causality between macroeconomic variables and stock market returns. *Global Finance J.* 1999; 10(1):71–81.
34. Chen N-F, Roll R, Ross SA. Economic forces and the stock market. *J Bus.* 1986:383–403.
35. Ou P, Wang H. Prediction of stock market index movement by ten data mining techniques. *Mod Appl Sci.* 2009; 3(12):28.
36. Fernández-Rodríguez F, Sosvilla-Rivero S, Andrada-Félix J. Nearest-neighbour predictions in foreign exchange markets. *Computational Intelligence in Economics and Finance*: Springer, Berlin, Heidelberg; 2004. pp. 297–325.
37. Barkoulas J, Baum CF, Chakraborty A. Nearest-neighbor forecasts of US interest rates. *International Journal of Banking and Finance.* 2003; 1(1):119–135.
38. Cont R, Nitons D. Statistical properties of financial time series. 1999. Available from: <http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=FF27A44E7634D28DD37321E26DD7EAF3?doi=10.1.1.32.9235&rep=rep1&type=pdf>.
39. He H, Chen J, Jin H, Chen S-H, editors. *Stock Trend Analysis and Trading Strategy*. JCIS; 2006. Available from: <http://users.cecs.anu.edu.au/~hdjin/publications/2006/CIEF-165.pdf>.
40. Pavlidis NG, Plagianakos VP, Tasoulis DK, Vrahatis MN. Financial forecasting through unsupervised clustering and neural networks. *Operational Research.* 2006; 6(2):103–127.
41. Cai F, Le-Khac N-A, Kechadi T. Clustering approaches for financial data analysis: a survey. *arXiv preprint arXiv:160908520.* 2016. Available from: <https://arxiv.org/pdf/1609.08520>.
42. Bhar R, Hamori S. *Empirical Techniques in Finance*: Springer Berlin Heidelberg; 2006. p. 46.

43. Rapach DE, Strauss JK. Forecasting real housing price growth in the eighth district states. Federal Reserve Bank of St Louis Regional Economic Development. 2007; 3(2):33–42. Available from: <http://research.stlouisfed.org/publications/red/2007/02/Rapach.pdf>.
44. Gupta R, Kabundi A, Miller SM. Forecasting the US real house price index: Structural and non-structural models with and without fundamentals. *Econ Modelling*. 2011; 28(4):2013–2021.
45. Piazzesi M, Schneider M. Inflation and the price of real assets: Federal Reserve Bank of Minneapolis, Research Department; 2009. Available from: <https://pdfs.semanticscholar.org/8cc9/15dc446e7d4d07198d9d51b0072b75336134.pdf>.
46. Lessard DR, Modigliani F. Inflation and the housing market: Problems and potential solutions. 1975. p. 15. Available from: <https://www.bostonfed.org/-/media/Documents/conference/14/conf14c.pdf>.
47. Kearl JR, Mishkin FS. Illiquidity, the demand for residential housing, and monetary policy. *J Finance*. 1977; 32(5):1571–1586.
48. Hendershott PH, Bosworth BP, Jaffee DM. Real user costs and the demand for single-family housing. *Brookings Pap Econ Act*. 1980; 1980(2):401–452.