

Article

An Adaptive Multipath Linear Interpolation Method for Sample Optimization

Yukun Du, Xiao Jin, Hongxia Wang and Min Lu

Special Issue

Advances in Computational Statistics and Applications

Edited by

Prof. Dr. José Antonio Roldán-Nofuentes



Article

An Adaptive Multipath Linear Interpolation Method for Sample Optimization

Yukun Du, Xiao Jin, Hongxia Wang * and Min Lu

School of Statistics and Data Science, Nanjing Audit University, Nanjing 211815, China

* Correspondence: hxwang@nau.edu.cn

Abstract: When using machine learning methods to make predictions, the problem of small sample sizes or highly noisy observation samples is common. Current mainstream sample expansion methods cannot handle the data noise problem well. We propose a multipath sample expansion method (AMLI) based on the idea of linear interpolation, which mainly solves the problem of insufficient prediction sample size or large error between the observed sample and the actual distribution. The rationale of the AMLI method is to divide the original feature space into several subspaces with equal samples, randomly extract a sample from each subspace as a class, and then perform linear interpolation on the samples in the same class (i.e., K -path linear interpolation). After the AMLI processing, valid samples are greatly expanded, the sample structure is adjusted, and the average noise of the samples is reduced so that the prediction effect of the machine learning model is improved. The hyperparameters of this method have an intuitive explanation and usually require little calibration. We compared the proposed method with a variety of machine learning prediction methods and demonstrated that the AMLI method can significantly improve the prediction result. We also propose an AMLI plus method based on the linear interpolation between classes by combining the idea of AMLI with the clustering method and present theoretical proofs of the effectiveness of the AMLI and AMLI plus methods.

Keywords: multipath; linear interpolation; sample optimization; predicted effects

MSC: 6211; 68T09



Citation: Du, Y.; Jin, X.; Wang, H.; Lu, M. An Adaptive Multipath Linear Interpolation Method for Sample Optimization. *Mathematics* **2023**, *11*, 768. <https://doi.org/10.3390/math11030768>

Academic Editor: José Antonio Roldán-Nofuentes

Received: 3 January 2023

Revised: 31 January 2023

Accepted: 1 February 2023

Published: 3 February 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

When using machine learning models to make predictions, a series of problems, such as insufficient sample sizes, missing parts of data, or large observation errors, are common. Especially for small sample datasets, how to improve model performance by using effective sample optimization techniques is crucial.

Improving model performance by increasing the number of numerical data can be traced back to the interpolation method at the earliest. Boor and Carl [1] proposed the cubic spline interpolation method, and Mitás and Mitásova [2] proposed the spatial interpolation method. Lu and Wong [3] proposed an adaptive inverse distance spatial interpolation algorithm, which uses the inverse proportional relationship between the distance between neighbors and the interpolation weight. Efron [4] proposed using the bootstrap resampling method based on jackknife; Chawla et al. [5] proposed the synthetic minority oversampling technique (SMOTE); Pan and Yang [6] proposed a transfer learning method to simultaneously model different types of label samples to increase sample size for model training. Fernandez et al. [7] proposed the SMOTE smoothing method, which constructs a new sample by randomly selecting a sample and randomly selecting multiple samples from its K -nearest neighbor.

With the advent of the era of big data, there are also some studies on sample optimization in machine learning and deep learning. Zhu (2005) [8] proposed active learning and

semi-supervised learning using the existing samples in the original sample space and using certain algorithms to label the unlabeled samples with high quality to achieve the effect of sample optimization. Eisenberger et al. [9] proposed an unsupervised shape interpolation method based on a neural network.

Recent technology research found that Few-Shot Learning has become a promising future development direction. Wang et al. [10] pointed out that the most important point in Few-Shot Learning was the use of prior knowledge. Prior knowledge of Few-Shot Learning comes from three sources: data, models, and algorithms. Current machine learning methods are in stark contrast to human perception, and their sample learning efficiency is very low. Few-Shot Learning is an exciting area of machine learning; it can solve the problem of low sample learning efficiency. Based on matching neural networks (matching learning), Vinyals et al. [11] proposed an LSTM to calculate the supported FCE, and they also optimized the sample by adding another LSTM to modify the embedding of the query samples. Snell et al. [12] proposed a striking inductive bias in the form of a class of prototypes in prototypical networks, and its achieved Few-Shot Learning performance can exceed matched networks without FCE complexity. Kokol et al. (2022) [13] proposed the synthetic data learning method and demonstrated that small samples can be better than large samples of low quality in the context of statistical machine learning. Zhou et al. (2022) [14] proposed a new improved multiscale edge-labeling graph neural network (MEGNN) to address the small sample size problem by acquiring as much feature information as possible.

When the sample size is expanded, the distribution of the added samples often deviates greatly from that of the actual samples (unknown). Moreover, the data observed daily often contain noise, and the expansion of samples with noise often aggravates the influence of the noise on the prediction result. The adaptive multipath linear interpolation AMLI method proposed in this paper can effectively solve these problems and ensure that most of the added samples are valid ones (i.e., their distribution generally deviates only slightly from the actual distribution). The AMLI method is mainly based on the linear interpolation method to expand the samples of the original data. The idea is to divide the original feature space into several subspaces with an equal number of samples, extract one sample from each subspace as a class, and then perform linear interpolation for the samples in the same class, which is K -path linear interpolation. This method requires two hyperparameters (K and η) in advance. The visual interpretation of parameter K is the number of samples existing in each feature subspace, while η is the number of samples interpolated per unit distance in the linear interpolation of the samples. In the simulation and empirical research, we found that the selection of parameter K is critical, which varies with different samples. By selecting appropriate hyperparameters, many valid samples can be expanded, and the proportion of samples in which the observed value deviates greatly from the actual value is reduced so that the composition structure of samples with error is adjusted and sample optimization is achieved; consequently, the impact of observation noise on the prediction result is also greatly reduced.

2. Research Hypothesis and Methodology Statement

This section describes the steps of the AMLI method and some assumptions that should be satisfied when using the AMLI method.

For a given training dataset:

$$T = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\},$$

where $x_i = (x_i^{(1)}, \dots, x_i^{(n)}) \in \mathcal{X} \subseteq R^n$ is the feature vector of the example, and n is the feature dimension. $y_i \in R$ is the corresponding output; $i = 1, 2, \dots, N$, where N represents the sample size. Assuming

$$y_i = f(x_i - \epsilon_i) + \tilde{\epsilon}, i = 1, 2, \dots, N,$$

where $f(\cdot)$ is a continuous function; ϵ_i is the set of independent and identically distributed observation noise; and $\tilde{\epsilon}$ is the model error. The specific steps of the AMLI method are as follows.

First, the hyperparameter K is determined, and the feature space \mathcal{X} is divided into N/K feature subspaces, each subspace containing K observation samples. Second, a sample is randomly selected from each subspace to form a set S_d , $d = 1, 2, \dots, K$, and then, we have:

$$T = \{S_1, S_2, \dots, S_K\},$$

where each S_d contains N/K samples.

$\exists x_0 \in \mathcal{X}$, for $\forall i$, we have $x_0^{(i)} = \inf_{x \in \mathcal{X}} \{x^{(i)}\}$ and call x_0 the feature space minimum point. Let $L(\cdot)$ be the distance measurement function of the feature space \mathcal{X} ($S_d = \{x^{(d,1)}, \dots, x^{(d,N/K)}\}$, $x^{(d,1)} = \operatorname{argmin}_{x \in S_d} L(x, x_0)$); we call the point in each sample class nearest to x_0 the minimum point of the sample class ($d = 1, \dots, K$); and $x^{(d,h+1)} = \operatorname{argmin}_{\{x: x \in S_d, x \neq x^{(d,1)}, \dots, x^{(d,h)}\}} L(x, x^{(d,h)})$, in which S_d is the observation point having the second nearest distance to $x^{(d,h)}$ (after $x^{(d,1)}, \dots, x^{(d,h)}$) ($h = 1, 2, \dots, N/K - 1$).

After determining the minimum point $x^{(d,1)}$ of the sample class, all the samples in the set are searched to find the $x^{(d,1)}$ point in the feature space nearest to $x^{(d,2)}$, and $x^{(d,3)}, \dots, x^{(d,N/K)}$. We define the unit distance filling parameter η and use linear interpolation in the feature space to interpolate $\sum_{i=1}^{N/K-1} \lfloor \eta \cdot L(x^{(d,i)}, x^{(d,i+1)}) \rfloor$ (for the sake of convenience, the number of interpolated samples in this paper is rounded down and will not be specifically mentioned below) virtual samples, and the interpolated samples are equally spaced. $x_d^{(h,h+1,i)}$ represents the i th dummy sample interpolated between $x^{(d,h)}$ and $x^{(d,h+1)}$ ($i = 1, \dots, \eta \cdot L(x^{(d,h)}, x^{(d,h+1)})$) in S_d , and for $x_d^{(h,h+1,i)}$ and its corresponding output $y_d^{(h,h+1,i)}$, it satisfies:

$$x_d^{(h,h+1,i)} = x^{(d,h)} + i \cdot \frac{x^{(d,h+1)} - x^{(d,h)}}{\eta \cdot L(x^{(d,h)}, x^{(d,h+1)}) + 1},$$

$$y_d^{(h,h+1,i)} = y^{(d,h)} + i \cdot \frac{y^{(d,h+1)} - y^{(d,h)}}{\eta \cdot L(x^{(d,h)}, x^{(d,h+1)}) + 1}.$$

The above linear interpolation is performed on all sample classes, the number of interpolations is $\sum_{d=1}^K \sum_{i=1}^{N/K-1} \eta \cdot L(x^{(d,i)}, x^{(d,i+1)})$, and all dummy samples are added to the dataset.

The AMLI method divides the original feature space into N/K subspaces with equal samples and then randomly selects a sample in each subspace as a sample class. When performing sample classification operations in practical applications, we can calculate the distances of all observation samples in the dataset to the minimum point of the feature space and classify each sample into its respective sample class in ascending order. The specific implementation process of the AMLI method is detailed in Algorithm 1.

Algorithm 1: AMLI

Input: Hyperparameters K, η
Distance measurement method $L(\cdot)$. e.g., Euclidean distance minimum point in x_0 . feature space set S_1, S_2, \dots, S_k

Output: T (Resulting dataset)

```

1 for  $d = 1, \dots, K$  do
2    $x^{(d,1)} = \underset{x \in S_d}{\operatorname{argmin}} L(x, x_0)$ 
3   for  $h = 1, \dots, N/K - 1$  do
4      $x^{(d,h+1)} = \underset{\{x: x \in S_d, x \neq x^{(d,1)}, \dots, x^{(d,h)}\}}{\operatorname{argmin}} L(x, x^{(d,h)})$ 
5     for  $i = 1, 2, \dots, \eta \cdot L(x^{(d,h)}, x^{(d,h+1)})$  do
6        $x_d^{(h,h+1,i)} = x^{(d,h)} + i \cdot \frac{x^{(d,h+1)} - x^{(d,h)}}{\eta \cdot L(x^{(d,h)}, x^{(d,h+1)}) + 1}$ 
7        $y_d^{(h,h+1,i)} = y^{(d,h)} + i \cdot \frac{y^{(d,h+1)} - y^{(d,h)}}{\eta \cdot L(x^{(d,h)}, x^{(d,h+1)}) + 1}$ 
8       Add  $(x_d^{(h,h+1,i)}, y_d^{(h,h+1,i)})$  to dataset  $T$ 
9     end
10  end
11 end

```

3. Simulation Experiments**3.1. Monte Carlo Simulations**

In this section, we will demonstrate the optimization effect of the samples added by using the AMLI method on the overall samples by using six groups of Monte Carlo simulations. For the sake of simplicity and to achieve a better visualization effect, the feature dimension $n = 1$ is defined, and the real function relationship selected is $y = x^3$. Since there is a certain error between the observed value of the sample and the actual value, to simulate this effect, we add noise to the samples after data generation.

The distribution of data and the settings of noise and sample size are as follows:

Simulation 1: $N = 200, x \sim U(-2.5, 2.5), \epsilon \sim N(0, 1)$;

Simulation 2: $N = 500, x \sim U(-2.5, 2.5), \epsilon \sim N(0, 1)$;

Simulation 3: $N = 800, x \sim U(-2.5, 2.5), \epsilon \sim N(0, 1)$;

Simulation 4: $N = 200, x \sim N(0, 2.5), \epsilon \sim N(0, 1)$;

Simulation 5: $N = 200, x \sim t(5), \epsilon \sim N(0, 1)$;

Simulation 6: $N = 200, x \sim U(-2.5, 2.5), \epsilon \sim U(-1.732, 1.732)$.

Simulation 1 is used as the control group, and simulations 2 and 3 are used as experimental groups with different sample sizes; simulations 4 and 5 are experimental groups with different feature distributions; simulation 6 is an experimental group with different types of noise (noise distribution parameters are selected to unify the noise variance and to ensure that the expectation is 0). Some verification indicators (e.g., the proportion of samples with an error greater than 0.5, 1, 1.5, 2, or 2.5; and the mean square error (MSE) between the sample observation values and the actual value) are selected to test the optimization effect of the AMLI method on the original sample after processing.

$$p_{(\alpha)} = \frac{\sum_{i=1}^{N + \sum_{d=1}^K \sum_{i=1}^{N/K-1} \eta \cdot L(x^{(d,i)}, x^{(d,i+1)})} I(|x_i - x_i^*| > \alpha)}{N + \sum_{d=1}^K \sum_{i=1}^{N/K-1} \eta \cdot L(x^{(d,i)}, x^{(d,i+1)})},$$

$$MSE(x, x^*) = \frac{\sum_{i=1}^N (x_i - x_i^*)^2}{N},$$

where x is the observed value of the sample; x^* is the actual value of the sample; $N + \sum_{d=1}^K \sum_{i=1}^{N/K-1} \eta \cdot L(x^{(d,i)}, x^{(d,i+1)})$ is the number of total samples after AMLI optimization;

and $\alpha = 0.5, 1, 1.5, 2, 2.5$. Due to the randomness of each experiment, we selected different hyperparameters to perform multiple calibrations in each simulation and calculated the average value of each of the verification indicators for every 100 experiments.

To reflect the optimization effect of the AMLI method more visually, below, we describe the AMLI processing process of simulation 1 in detail. First, 200 samples are generated evenly in the interval of $(-2.5, 2.5)$ (Figure 1(1)). Second, noise that obeys the standard normal distribution is added to the feature variables (Figure 1(2)). Third, to achieve a better visualization effect, original samples are divided into only four categories (just to achieve a better visualization effect, and the verification index is not necessarily optimal) (i.e., the hyperparameter $K = 4$) with labels of different colors (Figure 1(3)). Fourth, given the small interval of the definition domain, high values of the unit distance filling parameter are selected, and sample filling is performed after setting the hyperparameter $\eta = 100$. After the filling, the sample size reaches 3035 (Figure 1(4)).

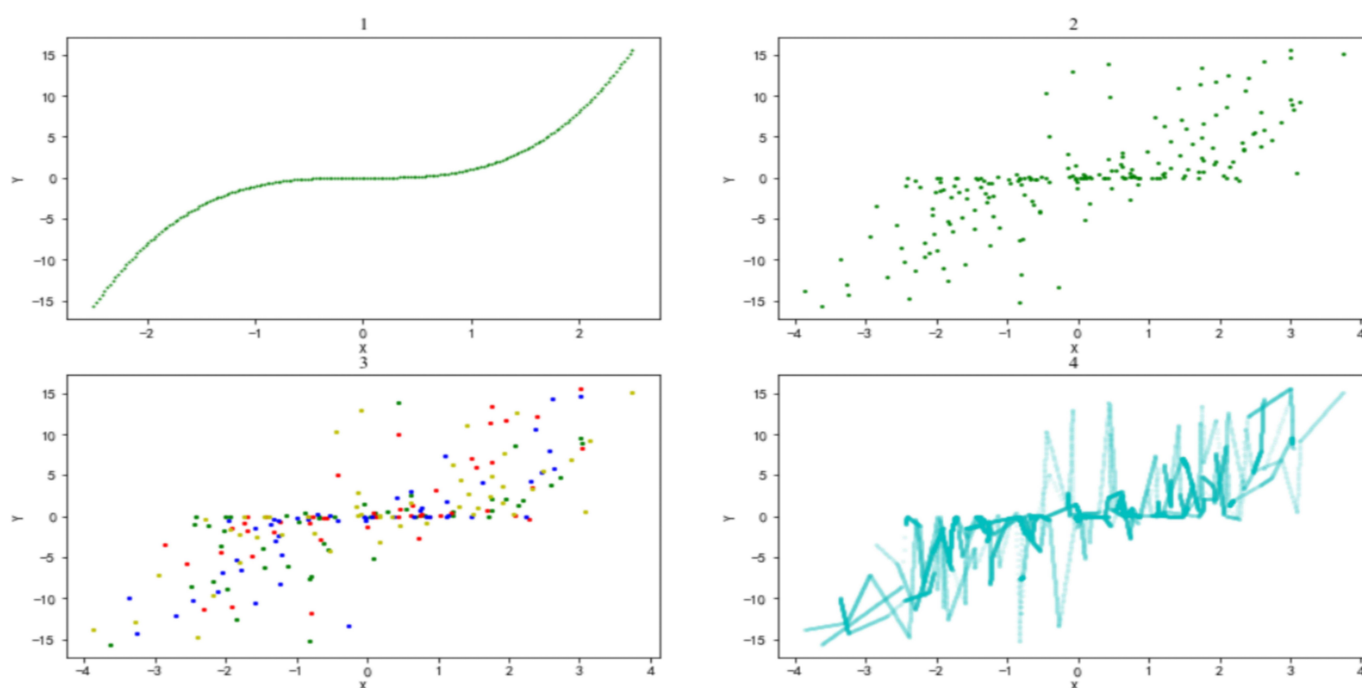


Figure 1. Steps of AMLI processing.

The comparison of the visualization effects in Figure 1(2) and Figure 1(4) indicates that after processing with the AMLI method, the samples can adaptively fit the functional relationship between x and y .

By using the above method, the parameters of the six simulations are calibrated, and the verification indicators are calculated, as shown in Table 1.

Table 1. Monte Carlo simulation results.

Simulation	MSE		$P(0.5)$		$P(1)$		$P(1.5)$		$P(2)$		$P(2.5)$	
	Before	After	Before	After	Before	After	Before	After	Before	After	Before	After
1	0.957	0.762	0.604	0.570	0.316	0.251	0.130	0.083	0.036	0.022	0.010	0.006
2	0.980	0.774	0.614	0.567	0.313	0.258	0.132	0.089	0.043	0.023	0.011	0.005
3	0.977	0.789	0.619	0.571	0.308	0.254	0.131	0.090	0.043	0.025	0.011	0.005
4	0.990	0.701	0.609	0.549	0.317	0.232	0.133	0.073	0.044	0.017	0.011	0.003
5	0.982	0.756	0.611	0.552	0.306	0.245	0.127	0.085	0.038	0.024	0.010	0.005
6	0.987	0.742	0.706	0.632	0.411	0.297	0.126	0.057	0.000	0.000	0.000	0.000

Remark 1. The optimal effect of each simulation is determined by the lowest MSE after AMLI processing by traversing all the hyperparameter values according to the grid search method.

Clearly, after the AMLI processing, the MSE of the samples and the proportion of various error samples have been optimized, most of the added dummy samples are valid ones, the increase in the sample size does not significantly weaken the AMLI's optimization effect, the samples that obey the normal distribution have better optimization performance, and even the uniformly distributed noise shows good robustness. In many cases, the AMLI method performs well in data optimization.

3.2. Analysis of Hyperparameter Taking Values

In this section, we discuss the patterns of setting the parameters of K and η in the AMLI method. First, we examine the setting of parameter K . For the above simulation, we set the parameter $\eta = 100$; perform traversal iterations consecutively by setting $K = 1, 2, \dots, 200$; and obtain the average value of the index for one hundred iterations at each setting. The change trend of the MSE index before and after the AMLI processing is shown in Figure 2.

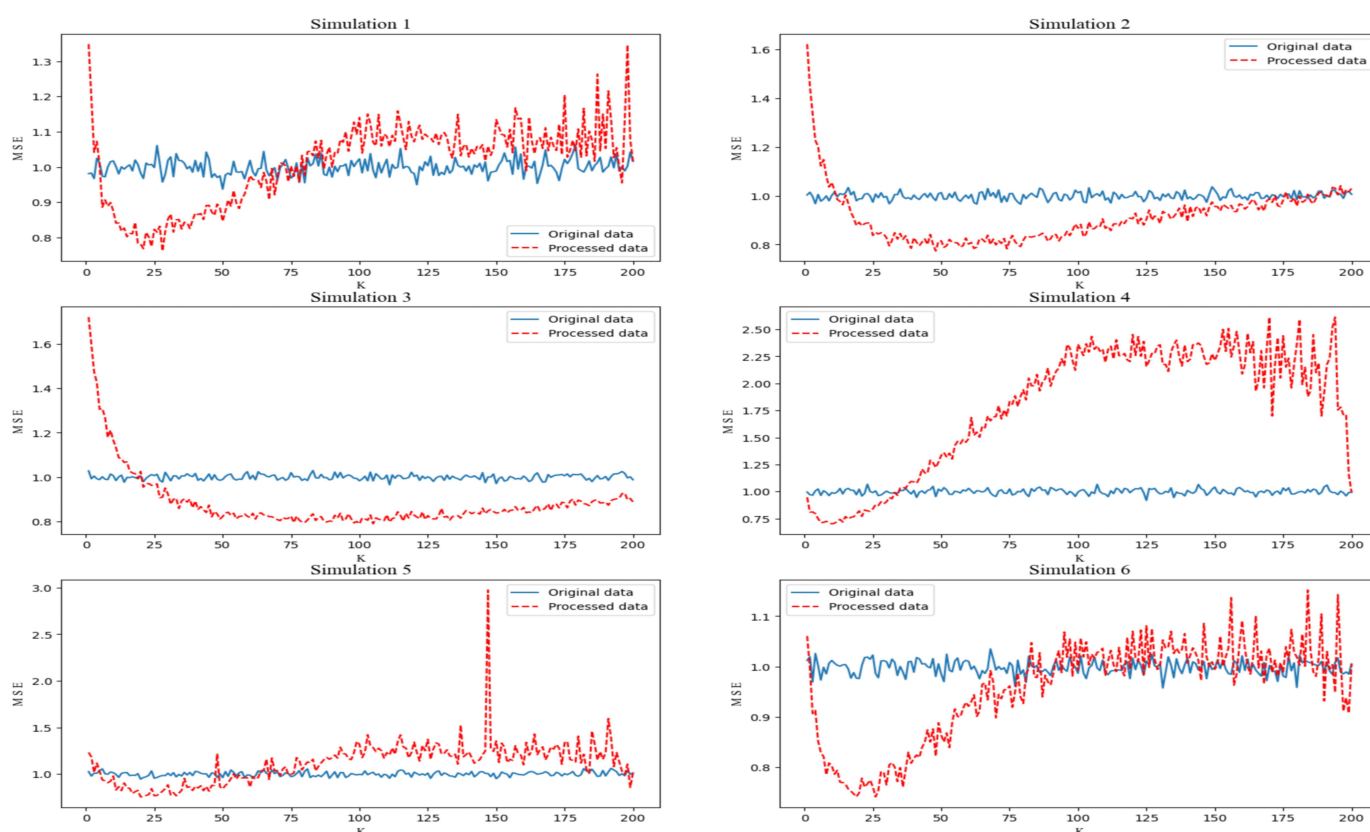


Figure 2. Changes in the MSE under different K values.

At fixed parameter η values, the optimal value of K increases with the increase in the sample size; the value range of K varies under different distributions of variables; and the change in the noise distribution has little impact on the optimal value of K .

Next, we examine the pattern of the setting of parameter η . In the case of simulation 1, under the condition of $K = 21$ to take the optimal value, traversal iterations are performed consecutively by setting the parameter $\eta = 1, 2, \dots, 200$; the results are shown in Figure 3. Clearly, the fluctuation of the sample MSE after optimization by using the AMLI method decreases with the increase in the value of filling parameter η .

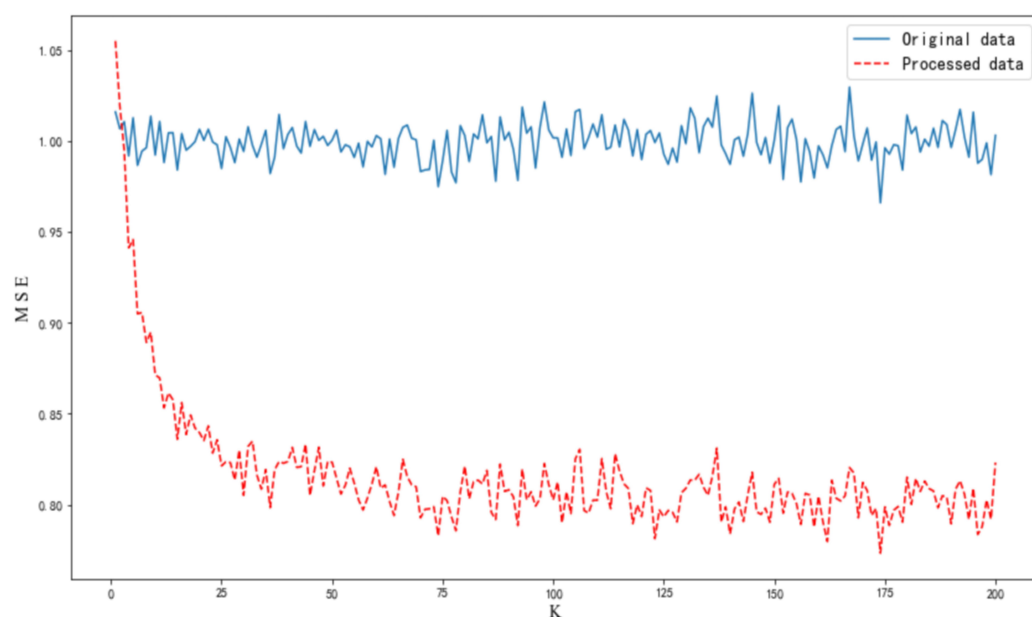


Figure 3. Changes in the *MSE* under different η values.

3.3. Comparison with Other Interpolation Methods

The above experiments show that the AMLI method can greatly expand valid samples and reduce the uniform error of samples as a whole so that the proportion of samples with large errors is low. The AMLI method achieves sample optimization based mainly on the idea of interpolation. The main interpolation methods include linear interpolation, quadratic spline interpolation and cubic spline interpolation. In this section, we will compare the AMLI method with other interpolation methods to demonstrate its superiority.

Similarly, simulation 1 is used as the control group, and simulations 2–6 are used as the experimental group, using various interpolation methods with a fixed interpolation number of 3500–4000. Only the *MSE* between the processed samples and the actual values is selected as the evaluation index, and for each simulation, one hundred experiments are carried out, and their average value is taken. The results are shown in Table 2. In terms of the *MSE*, the AMLI method clearly outperforms the other methods, indicating that the AMLI method has achieved a very significant optimization effect.

Table 2. *MSE* values of samples under different interpolation methods.

Simulation	<i>MSE</i>			
	<i>AMLI</i>	<i>Linear Interpolation</i>	<i>Quadratic Spline</i>	<i>Cubic Spline</i>
1	0.762	1.385	5.542	9.452
2	0.774	1.618	6.434	8.976
3	0.789	1.812	6.441	8.338
4	0.701	1.578	18.156	42.997
5	0.756	1.203	10.322	29.542
6	0.742	0.959	4.802	6.619

4. Application of AMLI Method in Machine Learning

In this section, we will describe the performance of the AMLI method in both simulated and actual data prediction when the method is combined with various machine learning models. Using the side-by-side method, we divide the training set and the test set in the ratio of 7:3; the training set is optimized by using the AMLI method. For the machine learning method, we select the *K*-Nearest Neighbor (*KNN*) method, Feedforward Neural Network (*FNN*), Gradient Boosting Decision Tree (*GBDT*) and Random Forest (*RF*), with the *MSE* as the loss function. The hyperparameters of each model and the parameters of *K* and η of the AMLI method are calibrated multiple times to obtain the optimal values. The Euclidean distance is adopted as the distance function:

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2,$$

$$L(x_i, x_j) = \sqrt{\sum_{l=1}^n (x_i^l - x_j^l)^2}.$$

4.1. Simulated Data Prediction

We assume that for simulated samples with a size of 1000, the simulated data feature dimension $n = 3$, different feature dimensions obey different distributions, $x^1 \sim N(0, 3)$, $x^2 \sim U(-3, 3)$, $x^3 \sim t(5)$, and the weight vector of $w_{(3,2)}^1, w_{(2,1)}^2$ is randomly generated. Let $Y = X \cdot w_{(3,2)}^1 \cdot w_{(2,1)}^2 + \tilde{\epsilon}$. After the data are generated, noise that obeys the Gaussian distribution is added to X , which is then further divided into the test set and the training set. The training set is processed using the AMLI method ($K = 40, \eta = 5$), and the total number of samples after the processing is 9810.

As shown in Table 3, the *MSE* of the trained model using the data processed by the AMLI method is smaller in the prediction, indicating that the prediction result is more accurate.

Table 3. Simulation data prediction results.

<i>MSE</i>	<i>KNN</i>	<i>FNN</i>	<i>GBDT</i>	<i>RF</i>
Before AMLI processing	1.70	0.942	1.210	1.507
After AMLI processing	1.07	0.713	1.008	1.320

4.2. Actual Data Prediction

4.2.1. Demand Forecast for Shared Bike Rental

In this section, we present a case study by predicting the demand for shared bicycle rentals in a certain city. The dataset includes multiple variables, such as season, holiday, temp, and registered (Table 4). We use the AMLI method to optimize samples of actual data. On the one hand, we combine the AMLI method with machine learning methods to examine its optimization performance in actual predictions; on the other hand, given that the dataset contains multiple categorical data, we can explore whether the AMLI method can achieve good optimization in prediction when the assumptions of the AMLI method are violated.

Table 4. Description of variable indicators.

Variable Name	Variable Definition
Season	1 = Spring 2 = Summer 3 = Autumn 4 = Winter
Holiday	1 = Holiday 0 = Non-holiday
Workdays	1 = Working day 0 = Weekend
Weather	1 = Sunny, cloudy 2 = Foggy, overcast 3 = Light snow, drizzle 4 = Heavy rain, heavy snow, heavy fog
Temp	Temperature in Celsius
Atemp	Apparent temperature
Humidity	Humidity
Windspeed	Wind speed
Casual	Number of nonregistered users
Registered	Number of registered users
Count	Total number of car rentals

The dataset contains 7620 observation samples. We select 1000, 3000, and 7620 samples consecutively to examine the prediction optimization of the AMLI method in combination with machine learning methods when the sample size is insufficient, fair, or sufficient.

As shown in Table 5, at various sample size levels, the AMLI method has achieved a certain optimization of prediction effect.

Table 5. Prediction results of shared bicycle rental demand (MSE).

<i>N</i>	Hyper-Parameter	<i>N</i> (After Processing)	AMLI Processing	KNN	FNN	GBDT	RF
1000	$K = 25$	191,057	Before	208.230	0.9677	88.152	231.210
	$\eta = 5$		After	46.340	0.0791	68.394	158.940
3000	$K = 40$	290,856	Before	42.757	0.3342	22.794	43.679
	$\eta = 3$		After	26.020	0.0084	14.301	25.451
7620	$K = 65$	428,277	Before	17.464	0.1168	9.879	23.521
	$\eta = 1$		After	6.798	0.0051	6.981	2.472

4.2.2. Concentration Forecast for $PM_{2.5}$

In this section, we present a case study by predicting the concentration for $PM_{2.5}$ in a certain city to rule out the randomness of the dataset. The dataset includes multiple concentrations of air pollutants such as PM_{10} , CO, SO_2 , NO_2 and O_3 . The dataset contains 3193 observation samples. We select 500, 1500, and 3193 samples consecutively to examine the prediction optimization of the AMLI method. As shown in Table 6, the AMLI method still has a good optimization effect.

Table 6. Prediction results of shared bicycle rental demand ($MSE(10^{-3})$).

<i>N</i>	Hyperparameter	<i>N</i> (After Processing)	AMLI Processing	KNN	FNN	GBDT	RF
500	$K = 17$	1089	Before	3.16	4.45	3.14	3.39
	$\eta = 10$		After	2.89	2.55	2.36	3.01
1500	$K = 23$	7003	Before	2.69	2.54	2.33	2.62
	$\eta = 30$		After	2.18	2.07	1.86	2.33
3193	$K = 50$	24,559	Before	1.77	1.63	1.48	1.54
	$\eta = 60$		After	1.41	1.13	1.07	1.16

5. Proof

In this section, we will present the proof for the samples that satisfy the AMLI hypothesis that after processing with the AMLI method, the average observation error of the samples is reduced, and the proportion of samples with different errors is adjusted.

The rationale of AMLI is to divide the original feature space into N/K subspaces with K samples, randomly extract a sample from each subspace as a class to divide the original dataset into K classes, and then perform linear interpolation between two adjacent samples. In essence, AMLI can be viewed as a method of linear interpolation between two subspaces with close distances through K paths.

For the dataset $T = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$, we select two subspaces that are assumed to have equal samples: $\mathcal{X}^{(1)}, \mathcal{X}^{(2)} \subseteq \mathcal{X} \subseteq R$, and $\mathcal{X}^{(1)} \cap \mathcal{X}^{(2)} = \emptyset$. In these two subspaces, we assume the following common relation is present:

$$y = f(x^*) + \tilde{\epsilon} = f(x - \epsilon) + \tilde{\epsilon}, \quad (1)$$

where x^* is the actual value of x after removing the observation noise; and where $x = x^* + \epsilon$, where ϵ is the noise term, $\epsilon \sim N(0, \sigma^2)$, and $\tilde{\epsilon}_i$ is the model error.

According to our assumption, $f(\cdot)$ is a continuous function; if $\mathcal{X}^{(1)}, \mathcal{X}^{(2)} \rightarrow 0$, and $\mathcal{X}^{(1)} \cap \mathcal{X}^{(2)} = \emptyset$, $\text{dis}(\mathcal{X}^{(1)}, \mathcal{X}^{(2)}) \rightarrow 0$, $f(\cdot)$ can be approximated as a linear function of $g(\cdot)$, and then (1) can be transformed into

$$y = g(x^*) + \epsilon_1 + \tilde{\epsilon} = g(x - \epsilon) + \epsilon_1 + \tilde{\epsilon}, \quad (2)$$

where ϵ_1 is the linear fitting error term and $\epsilon_1 \rightarrow 0$.

Since the samples we select in $\mathcal{X}^{(1)}, \mathcal{X}^{(2)}$ are equal in size, we assume there are K observation samples; i.e., $\mathbf{x}^{(j)} = (x_1^{(j)}, \dots, x_K^{(j)}) = (x_1^{*(j)} + \epsilon_1^{(j)}, \dots, x_K^{*(j)} + \epsilon_K^{(j)}) \in \mathcal{X}^{(j)}$, $j = 1, 2$. The expectation of the absolute value of the uniform noise in the space at this time is $\mathcal{X}^{(1)}, \mathcal{X}^{(2)}$:

$$\begin{aligned} E\left(\frac{\sum_{j=1}^2 \sum_{i=1}^K |\epsilon_i^{(j)}|}{2K}\right) &= \int_{-\infty}^{+\infty} |t| \frac{1}{\sqrt{2\pi} \cdot \sigma} e^{-\frac{t^2}{2\sigma^2}} dt \\ &= \sqrt{\frac{2}{\pi}} \cdot \sigma \int_0^{+\infty} \frac{t}{\sigma} e^{-\frac{t^2}{2\sigma^2}} d\frac{t}{\sigma} \\ &= \sqrt{\frac{2}{\pi}} \cdot \sigma \int_0^{+\infty} e^{-\frac{t^2}{2\sigma^2}} d\left(\frac{t}{\sigma}\right) \\ &= \sqrt{\frac{2}{\pi}} \cdot \sigma \cdot \Gamma(1) = \sqrt{\frac{2}{\pi}} \cdot \sigma. \end{aligned}$$

The expectation of the proportion of observed samples with a noise greater than 0.5 is

$$\begin{aligned} E\left(\frac{\sum_{j=1}^2 \sum_{i=1}^K I(|x_i^{(j)} - x_i^{*(j)}|) > 0.5}{2K}\right) &= E\left(\frac{\sum_{j=1}^2 \sum_{i=1}^K I(|\epsilon_i^{(j)}|) > 0.5}{2K}\right) \\ &= \frac{\sum_{j=1}^2 \sum_{i=1}^K (P(\epsilon_i^{(j)} > 0.5) + P(\epsilon_i^{(j)} < -0.5))}{2K} \\ &= \frac{\sum_{j=1}^2 \sum_{i=1}^K (P(\frac{\epsilon_i^{(j)}}{\sigma} > \frac{0.5}{\sigma}) + P(\frac{\epsilon_i^{(j)}}{\sigma} < \frac{-0.5}{\sigma}))}{2K} \\ &= \frac{\sum_{j=1}^2 \sum_{i=1}^K \Phi(\frac{-0.5}{\sigma})}{K} = 2\Phi(\frac{-0.5}{\sigma}). \end{aligned}$$

We randomly select two samples from the subspace to perform linear interpolation on them and iterate the process K times. The AMLI method determines the number of interpolation samples according to the distance between two samples. For the sake of

simplicity and convenience, we assume that the number of samples in each interpolation is m , and the rest of the situation is similar and provable. The sample for the i th interpolation is

$$\begin{aligned} x'_i &= (x'_{i,1}, x'_{i,2}, \dots, x'_{i,m}) \\ &= (x_i^{(1)} + \frac{x_i^{(2)} - x_i^{(1)}}{m+1}, x_i^{(1)} + 2 \cdot \frac{x_i^{(2)} - x_i^{(1)}}{m+1}, \dots, x_i^{(1)} + m \cdot \frac{x_i^{(2)} - x_i^{(1)}}{m+1}) \\ &= (x_i^{*(1)} + \epsilon_i^{(1)} + \frac{x_i^{*(2)} - x_i^{*(1)} + \epsilon_i^{(2)} - \epsilon_i^{(1)}}{m+1}, \dots, x_i^{*(1)} + \epsilon_i^{(1)} + m \cdot \frac{x_i^{*(2)} - x_i^{*(1)} + \epsilon_i^{(2)} - \epsilon_i^{(1)}}{m+1}), \end{aligned}$$

where $i = 1, \dots, K$ and the corresponding output is $y'_{i,d} = y_i^{(1)} + d \cdot \frac{y_i^{(2)} - y_i^{(1)}}{m+1}$. Based on (2), the noise of $x'_{i,d}$ is $\epsilon'_{i,d} = \epsilon_i^{(1)} + d \cdot \frac{\epsilon_i^{(2)} - \epsilon_i^{(1)}}{m+1}$, $d = 1, \dots, m$.

After K interpolations, the uniform noise expectation is

$$\begin{aligned} &E\left(\frac{\sum_{j=1}^2 \sum_{i=1}^K |\epsilon_i^{(j)}| + \sum_{i=1}^K \sum_{d=1}^m |\epsilon_i^{(1)} + d \cdot \frac{\epsilon_i^{(2)} - \epsilon_i^{(1)}}{m+1}|}{2K + K \cdot m}\right) \\ &= \frac{\sum_{j=1}^2 \sum_{i=1}^K E(|\epsilon_i^{(j)}|) + \sum_{i=1}^K \sum_{d=1}^m E(|\epsilon_i^{(1)} + d \cdot \frac{\epsilon_i^{(2)} - \epsilon_i^{(1)}}{m+1}|)}{2K + K \cdot m} \\ &= \frac{\sum_{j=1}^2 \sum_{i=1}^K E(|\epsilon_i^{(j)}|) + \frac{1}{m+1} \sum_{i=1}^K \sum_{d=1}^m E(|(m-d+1)\epsilon_i^{(1)} + d\epsilon_i^{(2)}|)}{2K + K \cdot m} \\ &= \frac{2 + \frac{1}{m+1} \sum_{d=1}^m \sqrt{(m-d+1)^2 + d^2}}{2 + m} \cdot \sigma \sqrt{\frac{2}{\pi}} \\ &< \frac{2 + \frac{1}{m+1} \sum_{d=1}^m \sqrt{(m-d+1)^2 + d^2 + 2d(m-d+1)}}{2 + m} \cdot \sigma \sqrt{\frac{2}{\pi}} \\ &= \frac{2 + \frac{1}{m+1} \sum_{d=1}^m \sqrt{(m+1)^2}}{2 + m} \cdot \sigma \sqrt{\frac{2}{\pi}} = \sigma \sqrt{\frac{2}{\pi}}. \end{aligned}$$

The above shows that the uniform noise of the samples after the optimization through the AMLI method is reduced. The expectation for the proportion of samples with a noise greater than 0.5 is

$$\begin{aligned} &E\left(\frac{\sum_{j=1}^2 \sum_{i=1}^K I(|\epsilon_i^{(j)}| > 0.5) + \sum_{i=1}^K \sum_{d=1}^m I(|\epsilon_i^{(1)} + d \cdot \frac{\epsilon_i^{(2)} - \epsilon_i^{(1)}}{m+1}| > 0.5)}{2K + K \cdot m}\right) \\ &= \frac{4K \cdot \Phi\left(\frac{-0.5}{\sigma}\right) + \sum_{i=1}^K \sum_{d=1}^m p(|\epsilon_i^{(1)} + d \cdot \frac{\epsilon_i^{(2)} - \epsilon_i^{(1)}}{m+1}| > 0.5)}{2K + K \cdot m} \\ &= \frac{4K \cdot \Phi\left(\frac{-0.5}{\sigma}\right) + \sum_{i=1}^K \sum_{d=1}^m p\left(\frac{|(m-d+1)\epsilon_i^{(1)} + d\epsilon_i^{(2)}|}{\sqrt{(m-d+1)^2 + d^2} \cdot \sigma} > \frac{0.5(m+1)}{\sqrt{(m-d+1)^2 + d^2} \cdot \sigma}\right)}{2K + K \cdot m} \\ &= \frac{4\Phi\left(\frac{-0.5}{\sigma}\right) + 2 \sum_{d=1}^m \Phi\left(-\frac{0.5(m+1)}{\sqrt{(m-d+1)^2 + d^2} \cdot \sigma}\right)}{2 + m} \\ &< \frac{4\Phi\left(\frac{-0.5}{\sigma}\right) + 2 \sum_{d=1}^m \Phi\left(-\frac{0.5(m+1)}{\sqrt{(m-d+1)^2 + d^2 + 2d(m-d+1)} \cdot \sigma}\right)}{2 + m} = 2\Phi\left(\frac{-0.5}{\sigma}\right). \end{aligned}$$

Thus, after being processed by the AMLI method, the proportion of samples with an error greater than 0.5 in the data decreases.

6. Extension

6.1. AMLI Plus

The above simulation experiments indicate that the selection of parameter K in the AMLI algorithm is very important. The selection of the optimal value of parameter K involves many parameter adjustment calculations while under the influence of randomness, thereby making it difficult to guarantee that the selected K value results in optimal performance at all times. If the AMLI method fails to achieve a good result, we can consider another interpolation method that combines the clustering method to perform linear interpolation between classes; we name this method AMLI plus. Below, we describe specific steps of the AMLI plus method.

First, all observed samples are clustered according to their distribution. Assuming that the number of clusters is K , and accordingly, the virtual space is divided into K subspaces, each containing all observed samples of the same category:

$$T = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_k, y_k)\} = \{G_1, G_2, \dots, G_K\},$$

where G_1 represents the class whose center is closest to the minimum point of the feature space \mathbf{x}_0 ; the center of the class is $\bar{\mathbf{x}}^{(d)} = \frac{1}{n^{(d)}} \sum_{i=1}^{n^{(d)}} \mathbf{x}_i^{(d)}$, $d = 1, \dots, K$, $n^{(d)}$ $G_d, d = 1, \dots, K$; $n^{(d)}$ is the number of samples in G_d ; and G_{d+1} satisfies $\bar{\mathbf{x}}^{(d+1)} = \underset{\{\bar{\mathbf{x}}: \bar{\mathbf{x}} \neq \bar{\mathbf{x}}^{(1)}, \dots, \bar{\mathbf{x}}^{(d)}\}}{\operatorname{argmin}} (L(\bar{\mathbf{x}}^{(d)}, \bar{\mathbf{x}}))$.

The choice of clustering method can be diverse; additionally, the K -means, the X -means or the DBSCAN method, which can eliminate noise points according to Ester et al. [15], can be selected.

Second, interpolation is performed between classes, in which the unit distance filling parameter η is defined; additionally, $n^{(d)} \cdot n^{(d+1)}$ linear interpolations are performed between $\mathbf{x}_i^{(d)} \in G_d$ and all samples in G_{d+1} . The number of interpolation samples is $\sum_{j=1}^{n^{(d+1)}} \sum_{i=1}^{n^{(d)}} \eta \cdot L(\mathbf{x}_i^{(d)}, \mathbf{x}_j^{(d+1)})$, and the interpolated samples are also equally spaced.

The specific implementation process of the AMLI method is detailed in Algorithm 2.

Algorithm 2: AMLI plus

Input: unit distance filling parameter η
distance measurement method $L(\cdot)$
minimum point of feature space \mathbf{x}_0
set G_1, G_2, \dots, G_K

Output: T (Resulting dataset)

```

1 for  $d = 1, \dots, K - 1$  do
2   for  $h = 1, \dots, n^{(d)}$  do
3     for  $j = 1, \dots, n^{(d+1)}$  do
4       for  $i = 1, \dots, \eta \cdot L(\mathbf{x}^{(d+1,j)}, \mathbf{x}^{(d,h)})$  do
5          $\mathbf{x}_d^{(h,j,i)} = \mathbf{x}^{(d,h)} + i \cdot \frac{\mathbf{x}^{(d+1,j)} - \mathbf{x}^{(d,h)}}{\eta \cdot L(\mathbf{x}^{(d+1,j)}, \mathbf{x}^{(d,h)}) + 1}$ ,
           $y_d^{(h,j,i)} = y^{(d,h)} + i \cdot \frac{y^{(d+1,j)} - y^{(d,h)}}{\eta \cdot L(y^{(d+1,j)}, y^{(d,h)}) + 1}$ .
6         Add  $(\mathbf{x}_d^{(h,j,i)}, y_d^{(h,j,i)})$  to dataset  $T$ 
7       end
8     end
9   end
10 end
```

6.2. The Proof of AMLI Plus

In this section, we will present evidence of the effectiveness of the AMLI plus method. The proof idea is essentially the same as that of the AMLI method, and we will focus on the differences between the two.

Unlike the AMLI method, the AMLI plus method divides the virtual space into K subspaces according to the number of clusters, and each subspace contains all observation samples of the same category; thus, the sample size of each subspace may vary. It is assumed that two adjacent subspaces, namely, $\mathcal{X}^{(1)}, \mathcal{X}^{(2)}$, contain $n^{(1)}, n^{(2)}$ samples, respectively; additionally, it is assumed that $n^{(1)} \cdot n^{(2)}$ linear interpolations are performed. For the sake of simplicity, assuming that the number of samples interpolated each time is m , then the uniform noise after the interpolation is

$$\begin{aligned} & E\left(\frac{\sum_{i=1}^{n^{(1)}} |\epsilon_i^{(1)}| + \sum_{i=1}^{n^{(2)}} |\epsilon_i^{(2)}| + \sum_{i=1}^{n^{(1)}} \sum_{j=1}^{n^{(2)}} \sum_{d=1}^m \left| \epsilon_i^{(1)} + d \cdot \frac{\epsilon_j^{(2)} - \epsilon_i^{(1)}}{m+1} \right|}{n^{(1)} + n^{(2)} + n^{(1)}n^{(2)}m}\right) \\ &= \frac{\sum_{i=1}^{n^{(1)}} E(|\epsilon_i^{(1)}|) + \sum_{i=1}^{n^{(2)}} E(|\epsilon_i^{(2)}|) + \sum_{i=1}^{n^{(1)}} \sum_{j=1}^{n^{(2)}} \sum_{d=1}^m E\left(\left| \epsilon_i^{(1)} + d \cdot \frac{\epsilon_j^{(2)} - \epsilon_i^{(1)}}{m+1} \right| \right)}{n^{(1)} + n^{(2)} + n^{(1)}n^{(2)}m} \\ &= \frac{n^{(1)}\sigma\sqrt{\frac{2}{\pi}} + n^{(2)}\sigma\sqrt{\frac{2}{\pi}} + \frac{1}{m+1} \sum_{i=1}^{n^{(1)}} \sum_{j=1}^{n^{(2)}} \sum_{d=1}^m E\left(\left| (m-d+1)\epsilon_i^{(1)} + d\epsilon_j^{(2)} \right| \right)}{n^{(1)} + n^{(2)} + n^{(1)}n^{(2)}m} \\ &= \frac{n^{(1)} + n^{(2)} + \frac{n^{(1)}n^{(2)}}{m+1} \sum_{d=1}^m \sqrt{(m-d+1)^2 + d^2}}{n^{(1)} + n^{(2)} + n^{(1)}n^{(2)}m} \cdot \sigma\sqrt{\frac{2}{\pi}} \\ &< \frac{n^{(1)} + n^{(2)} + \frac{n^{(1)}n^{(2)}}{m+1} \sum_{d=1}^m \sqrt{(m-d+1)^2 + d^2 + 2d(m-d+1)}}{n^{(1)} + n^{(2)} + n^{(1)}n^{(2)}m} \cdot \sigma\sqrt{\frac{2}{\pi}} \\ &= \frac{n^{(1)} + n^{(2)} + \frac{n^{(1)}n^{(2)}}{m+1} \sum_{d=1}^m m + 1}{n^{(1)} + n^{(2)} + n^{(1)}n^{(2)}m} \cdot \sigma\sqrt{\frac{2}{\pi}} = \sigma\sqrt{\frac{2}{\pi}}. \end{aligned}$$

Thus, the uniform noise after optimization by the AMLI plus method is reduced. The proportion of samples with a noise error greater than 0.5 is expected to be

$$\begin{aligned} & E\left(\frac{\sum_{i=1}^{n^{(1)}} I(|\epsilon_i^{(1)}| > 0.5) + \sum_{i=1}^{n^{(2)}} I(|\epsilon_i^{(2)}| > 0.5) + \sum_{i=1}^{n^{(1)}} \sum_{j=1}^{n^{(2)}} \sum_{d=1}^m I\left(\left| \epsilon_i^{(1)} + d \cdot \frac{\epsilon_j^{(2)} - \epsilon_i^{(1)}}{m+1} \right| > 0.5\right)}{n^{(1)} + n^{(2)} + n^{(1)}n^{(2)}m}\right) \\ &= \frac{\sum_{i=1}^{n^{(1)}} P(|\epsilon_i^{(1)}| > 0.5) + \sum_{i=1}^{n^{(2)}} P(|\epsilon_i^{(2)}| > 0.5) + \sum_{i=1}^{n^{(1)}} \sum_{j=1}^{n^{(2)}} \sum_{d=1}^m P\left(\left| \epsilon_i^{(1)} + d \cdot \frac{\epsilon_j^{(2)} - \epsilon_i^{(1)}}{m+1} \right| > 0.5\right)}{n^{(1)} + n^{(2)} + n^{(1)}n^{(2)}m} \\ &= \frac{2(n^{(1)} + n^{(2)})\Phi\left(\frac{-0.5}{\sigma}\right) + \sum_{i=1}^{n^{(1)}} \sum_{j=1}^{n^{(2)}} \sum_{d=1}^m P\left(\frac{|(m-d+1)\epsilon_i^{(1)} + d\epsilon_j^{(2)}|}{\sqrt{(m-d+1)^2 + d^2} \cdot \sigma} > \frac{0.5(m+1)}{\sqrt{(m-d+1)^2 + d^2} \cdot \sigma}\right)}{n^{(1)} + n^{(2)} + n^{(1)}n^{(2)}m} \\ &= \frac{2(n^{(1)} + n^{(2)})\Phi\left(\frac{-0.5}{\sigma}\right) + 2n^{(1)}n^{(2)} \sum_{d=1}^m \Phi\left(-\frac{0.5(m+1)}{\sqrt{(m-d+1)^2 + d^2} \cdot \sigma}\right)}{n^{(1)} + n^{(2)} + n^{(1)}n^{(2)}m} \\ &< \frac{2(n^{(1)} + n^{(2)})\Phi\left(\frac{-0.5}{\sigma}\right) + 2n^{(1)}n^{(2)} \sum_{d=1}^m \Phi\left(-\frac{0.5(m+1)}{\sqrt{(m-d+1)^2 + d^2 + 2d(m-d+1)} \cdot \sigma}\right)}{n^{(1)} + n^{(2)} + n^{(1)}n^{(2)}m} = 2\Phi\left(\frac{-0.5}{\sigma}\right). \end{aligned}$$

Therefore, after the samples are processed by the AMLI plus method, the proportion of samples with an error greater than 0.5 in the data decreases.

Remark 2. *The proof rationale of the AMLI plus method is to divide the clustered samples of different categories into different subspaces and to assume that neighboring subspaces have a common linear relationship; therefore, when the number of clusters and the clustering method are selected, the above requirement should be satisfied as much as possible.*

7. Conclusions

In this study, we propose a multipath sample interpolation method based on the idea of linear interpolation, which can solve the problem of insufficient sample sizes or large errors between observed samples and actual distribution when making predictions. The AMLI method, simple to implement and flexible, can greatly expand valid samples, reduce the influence of sample noise, and thus significantly improve the prediction effect. Finally, we propose the AMLI plus method, another class-to-class-based linear interpolation method, which can also achieve good optimization results. In general, we find that the AMLI method is robust, effective, and very suitable for addressing a series of problems in machine learning, e.g., insufficient sample sizes and large amounts of observation noise.

Author Contributions: Methodology, H.W.; Software, Y.D.; Writing—original draft, X.J.; Writing—review & editing, M.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the National Social Science Fund of China under Grant No. 22BTJ021, “Qinglan project” of Colleges and Universities of Jiangsu Province and Postgraduate Research & Practice Innovation Program of Jiangsu Province under Grant No. KYCX22_2148.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to thank the editor and reviewers for the valuable advice given in order to develop the article.

Conflicts of Interest: The authors declare no conflict of interest.

References

- De Boor, C. *A Practical Guide to Splines*; Springer: New York, NY, USA, 1978.
- Mitas, L.; Mitasova, H. Spatial interpolation. *Geogr. Inf. Syst. Princ. Tech. Manag. Appl.* **1999**, *1*, 481–492.
- Lu, G.Y.; Wong, D.W. An adaptive inverse distance weighting spatial interpolation technique. *Comput. Geosci.* **2008**, *34*, 1044–1055. [[CrossRef](#)]
- Efron, B. Bootstrap methods: Another look at the jackknife. In *Breakthroughs in Statistics*; Springer: New York, NY, USA, 1992; pp. 569–593.
- Chawla, N.V.; Bowyer, K.W.; Hall, L.O.; Kegelmeyer, W.P. SMOTE: Synthetic minority over-sampling technique. *J. Artif. Intell. Res.* **2002**, *16*, 321–357. [[CrossRef](#)]
- Pan, S.J.; Yang, Q. A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.* **2010**, *22*, 1345–1359. [[CrossRef](#)]
- Fernández, A.; Garcia, S.; Herrera, F.; Chawla, N.V. SMOTE for learning from imbalanced data: Progress and challenges, marking the 15-year anniversary. *J. Artif. Intell. Res.* **2018**, *61*, 863–905. [[CrossRef](#)]
- Zhu, X.J. *Semi-Supervised Learning Literature Survey*; University of Wisconsin-Madison: Madison, WI, USA, 2005.
- Eisenberger, M.; Novotny, D.; Kerchenbaum, G.; Labatut, P.; Neverova, N.; Cremers, D.; Vedaldi, A. Neuromorph: Unsupervised shape interpolation and correspondence in one go. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 7473–7483.
- Wang, Y.; Yao, Q.; Kwok, J.T.; Ni, L.M. Generalizing from a few examples: A survey on few-shot learning. *ACM Comput. Surv.* **2020**, *53*, 1–34. [[CrossRef](#)]
- Vinyals, O.; Blundell, C.; Lillicrap, T.; Wierstra, D. Matching networks for one shot learning. *Adv. Neural Inf. Process. Syst.* **2016**, *29*, 3637–3645.
- Snell, J.; Swersky, K.; Zemel, R. Prototypical networks for few-shot learning. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 4077–4087.
- Kokol, P.; Kokol, M.; Zagoranski, S. Machine learning on small size samples: A synthetic knowledge synthesis. *Sci. Prog.* **2022**, *105*, 1–15. [[CrossRef](#)] [[PubMed](#)]

14. Zhou, Y.; Zhi, G.; Chen, W.; Qian, Q.; He, D.; Sun, B.; Sun, W. A new tool wear condition monitoring method based on deep learning under small samples. *Measurement* **2022**, *189*, 110622. [[CrossRef](#)]
15. Ester, M.; Kriegel, H.P.; Sander, J.; Xu, X. A density-based algorithm for discovering clusters in large spatial databases with noise.kdd. *KDD* **1996**, *96*, 226–231.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.