ASISO: A Robust and Stable Data Synthesis Method to Optimize Samples

Yukun Du , Yitao Cai , Xiao Jin , Hongxia Wang * , Yao Li , Min Lu

Posted Date: 18 August 2023

doi: 10.20944/preprints202308.1378.v1

Keywords: data synthesis; unknown noise; interpolation; sample optimization; robust

Article

ASISO: A Robust and Stable Data Synthesis Method to Optimize Samples

Yukun Du †, Yitao Cai †, Xiao Jin †, Hongxia Wang *, Yao Li and Min Lu

School of Statistics and Data Science, Nanjing Audit University, Nanjing 211815, China
* Correspondence: hxwang@nau.edu.cn
† These authors contributed equally.

Abstract: Most existing data synthesis methods are designed to tackle problems such as dataset imbalance, data anonymization and insufficient sample size. There is a lack of effective synthesis methods for the limited number of datasets which contain a large of features and unknown noise to expand the size of the dataset. We propose a data synthesis method, named Adaptive Subspace Interpolation for Sample Optimization (ASISO). The idea is to divide the original feature space into several subspaces with an equal number of samples, and then perform interpolation for the samples in the adjacent subspaces. This method can adaptively adjust the size of the dataset containing unknown noise, and the expanded data typically contain minimal error with actual. Moreover, it adjusts the structure of the samples, which can significantly reduce the proportion of samples with large errors. In addition, the hyperparameters of this method have an intuitive explanation and usually require little calibration. Experimental results on artificial data and benchmark data sets demonstrate that ASISO is a robust and stable method to optimize samples.

Keywords: data synthesis; unknown noise; interpolation; sample optimization; robust

MSC: 6211; 68T09

1. Introduction

Synthetic data presents an effective solution to the challenges of inadequate or low-quality sample, particularly in the era of big data. Through the use of data synthesis methods to generate synthetic data, it provides a cost-effective and efficient alternative to collecting and labeling large amounts of real-world data. Furthermore, these methods can address privacy concerns associated with real-world data, making it safer to share and analyze [1]. Recently, there has been a surge in the use of synthetic data in machine learning, with various synthetic methods being developed [2,3].

Representative data synthesis methods can be classified into three categories. The first category entails techniques such as interpolation, extrapolation, and other methods [4] that generate additional data points representative of the underlying distribution. These methods improve the quality of the dataset and model generalization. For image data, deep learning-based methods such as VAE and GAN can also be used to generate new data [5,6]. These methods are useful for improving the quality of the dataset and model generalization. The second category is to address the issue of dataset imbalance, such as SMOTE [7], and some of its enhancements [8–10]. These methods can synthesize minority class samples to balance the dataset and improve the model's performance. Finally, data sharing and research require sensitive data privacy protection. Synthetic data can help protect sensitive information while still enabling data sharing and research. Normally, we can add random noise to protect original data, generating synthetic data for sharing and research purposes like differential privacy methods [11]. Overall, these methods help optimize the representation and use of data in various applications.

For some actual tasks, original data often contains a multitude of features and unknown noise [12]. Since synthetic method involves generating new data based on existing data, the quality of the new points depends on the quality and quantity of the original data. If the quality of the original

 doi:10.20944/preprints202308.1378.v1

data is poor or the quantity is insufficient, then the synthetic data may have limitations in terms of its quality [13]. However, there exists a lack of effective synthetic methods for datasets with a restricted size and complex noise to expand the size of data set.

Based on the purpose of improving the quality and quantity of the dataset, and motivated by piecewise linear interpolation and spline interpolation, we propose a robust and stable data synthesis method named Adaptive Subspace Interpolation for Sample Optimization (ASISO), which aims to adaptively adjust the sample size and structure of the original dataset containing unknown noise. The idea is to divide the original feature space into several subspaces with an equal number of samples, and then perform linear interpolation for the samples in the adjacent subspaces. This method achieves sample optimization in two aspects. First, it can adaptively adjust the size of the dataset, and the expanded data typically contains minimal error with actual. Second, it adjusts the structure of the samples, which can significantly reduce the proportion of samples with large errors, thereby minimizing the impact of noise for model generalization.

The rest of this paper is organized as follows. Section 2 reviews the research of existing interpolation. Section 3 details the concept of the proposed ASISO method and provides proof for the effectiveness of this method. Experimental results are presented and analyzed in Section 4. Finally, we conclude this paper in Section 5.

2. Related Work

Traditional interpolation methods are based on the function values of known data points for extrapolation and prediction. For a given data set, there are generally two cases. In the case of two known data points and interpolation in between, interpolation can be selected based on distance, such as nearest-neighbor interpolation [14]. If it is assumed that the unknown point between these two data points is consistent with the straight line between them, a linear function can be used to fit and interpolate, such as linear interpolation, piecewise linear interpolation [15,16]. For interpolation among multiple given points, a commonly used method is to predict the value of the unknown point by constructing a high-degree polynomial based on the given points, such as Lagrange interpolation, Newton interpolation [17,18]. However, with an increase in the number of data points and the degree of the polynomial, there is a risk of overfitting and numerical instability (Runge's phenomenon) [19]. Another solution is to construct a global smooth function by fitting a low-degree polynomial in a local region. In comparison to high-degree polynomial interpolation methods, this method has better smoothness and numerical stability, such as spline interpolation [20]. Nevertheless, as it is necessary to fit multiple local low-degree polynomials, this method can lead to relatively high computational complexity.

When expanding the sample size using interpolation methods, the selection of node positions and quantities has a significant impact on the accuracy and stability of interpolation results. Typically, equidistant nodes [21] are used for interpolation position selection, where nodes are equally spaced within the interpolation interval. Chebyshev nodes [22] are selected within the interpolation interval to satisfy certain conditions for better fitting of the function. In addition, the choice of the number of interpolations can lead to instability of the data and insufficient validation accuracy. One way to determine the appropriate number of interpolations is by comparing the fitting degree of the model trained under different interpolation numbers with the original data, but this may lead to overfitting and poor generalization ability of the model.

3. Proposed Method

In this section, we will explain the proposed ASISO method in detail. ASISO introduces two algorithms: K-Space and K-Match. To clearly illustrate the proposed method, we will first provide an overview of ASISO and then introduce the specific algorithms involved.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 18 August 2023 doi:10.20944/preprints202308.1378.v1

3.1. Overview

The ASISO is mainly based on the linear interpolation to increase the size and improve the quality of the original dataset. The idea is to divide the original feature space into several subspaces with an equal number of samples, and then perform linear interpolation for the samples in the adjacent subspaces. This method requires two hyperparameters (k and η) in advance. Interpretation of parameter k is the number of samples existing in each feature subspace, while η is the number of equidistant nodes interpolated per unit distance in the linear interpolation of the samples. This proposed method is illustrated in Figure 1.

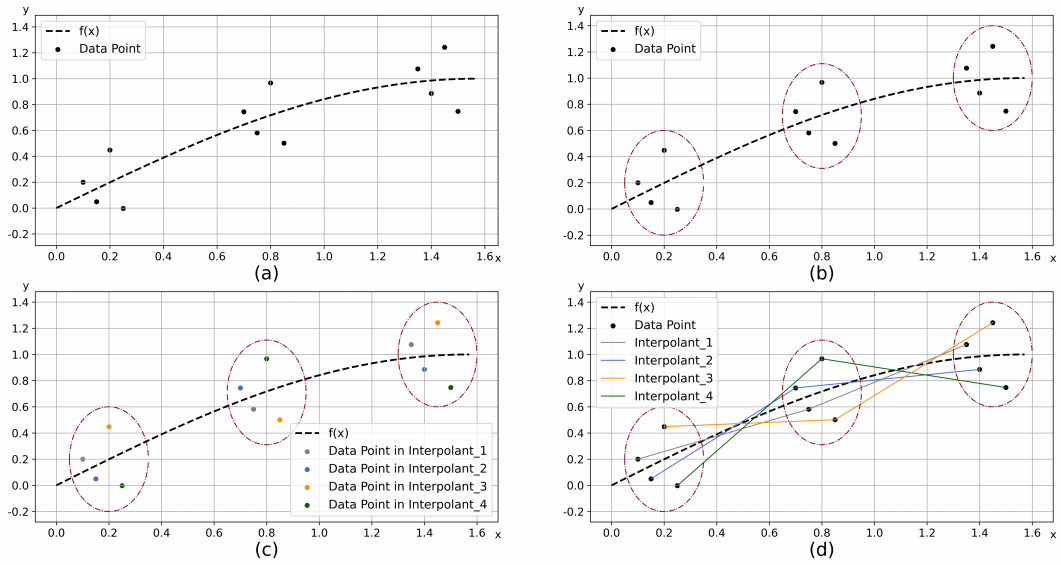

Figure 1. Overview of the proposed approach for ASISO. (a) The dataset contains multiple noisy samples, and the true functional relationship between x and y is f(·). (b) In the first step, we use the K-Space algorithm to perform unsupervised clustering on the dataset and divide the original feature space into several subspaces. (c) Interpolation matching of samples between adjacent subspaces is performed using the K-Match algorithm, and sample points with the same color belong to the same class to be interpolated. (d) Piecewise linear interpolation is performed on samples under different classes, and inserting equidistant sample points on lines of different colors in adjacent subspaces.

The data set $D = \{\boldsymbol{x}_i, y_i\}_{i=1}^n$ has been given and is assumed to be contaminated with unknown noise, where $\boldsymbol{x}_i \in \mathcal{X} = R^p$ and $y_i \in \mathcal{Y} = R$. Assuming $\dot{y}_i = f(\dot{\boldsymbol{x}}_i)$ where $\dot{\boldsymbol{x}}_i$ is the actual value of $\boldsymbol{x}_i$, $\dot{y}_i$ is the actual value of y_i and $f(\cdot)$ is a continuous function, represents the relationship in reality between $\boldsymbol{x}_i$ and y. Consider the model:

$$\dot{y}_i + \epsilon_{i,y} = f(\dot{\boldsymbol{x}}_i + \boldsymbol{\epsilon}_{i,x}) + \epsilon_i, \tag{1}$$

where $\epsilon_{i,y}$ is the noise in $\dot{y}_i$, $\boldsymbol{\epsilon}_{i,x}$ is the noise in $\dot{\boldsymbol{x}}_i$, and ϵ_i represents the error term. The expression (1) can be rewritten as:

$$y_i = f(\boldsymbol{x}_i) + \epsilon_i. \tag{2}$$

Let $\boldsymbol{x}_0 = \{x_0^1, \ldots, x_0^p\}$, we have $x_0^j = inf\{x_i^j\}_{i=1}^n$ for $\forall j = 1, \ldots, p$, and call $\boldsymbol{x}_0$ the sample minimum point.

Given the hyperparameter k, we provide an unsupervised clustering method called K-Space. As it is shown in Figure 1(b), the space can be partitioned into n/k subspaces, each containing k samples. i.e., $\mathcal{X} = \cup_{s=1}^{n/k} \mathcal{X}_s$, $\mathcal{X}_i \cap \mathcal{X}_j = \emptyset, i, j = 1, 2, \ldots, \frac{n}{k}$ and $i \neq j$. The datasets corresponding to different subspaces $D = \cup_{s=1}^{n/k} D_s, D_s = \{(\boldsymbol{x}_i^s, y_i^s)\}_{i=1}^k, \boldsymbol{x}_i^s \in \mathcal{X}_s$. For two adjacent subspaces, since $f(\cdot)$ is

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 18 August 2023 doi:10.20944/preprints202308.1378.v1

a continuous function, we assume that it can be approximated as a linear function $g(\cdot)$, and then (2) can be transformed into:

$$y_i = g(\boldsymbol{x}_i) + \epsilon_i + \epsilon_i', \tag{3}$$

where ϵ_i' is the linear fitting error term. When the distance between two adjacent subspaces approaches zero and the measurements of subspaces tend to zero, obtain $\epsilon_i' \to 0$. Next, we will perform sample interpolation between adjacent subspaces.

We need to calculate the centers of each cluster, as follow:

$$\overline{\boldsymbol{x}}^s = \frac{1}{k} \sum_{\boldsymbol{x}_i^s \in D_s} \boldsymbol{x}_i^s. \tag{4}$$

To make $\epsilon_i' \to 0$, we need to ensure that the interpolation is performed between clusters that are close in distance as much as possible. Among $\{D_s\}_{s=1}^{n/k}$, we define $D_{(1)}$ whose cluster center has the minimum distance to the sample minimum point $\boldsymbol{x}_0$, and define $D_{(d)}$ whose cluster center has the minimum distance to the center of $D_{(d-1)}$, $D_{(d-1)} \neq D_{(1)}, \ldots, D_{(d-1)}$ and $d > 1$.

$$D_{(1)} = \underset{D_s \in D}{argmin}\, dist(\boldsymbol{x}_0, \overline{\boldsymbol{x}}^s), \tag{5}$$

$$D_{(d)} = \underset{\{D_s \in D\}, D_s \neq D_{(1)}, \ldots, D_{(d-1)}}{argmin} dist(\overline{\boldsymbol{x}}_0^{(d-1)}, \overline{\boldsymbol{x}}^s), \tag{6}$$

where $\overline{\boldsymbol{x}}^{(d-1)}$ is the center of $D_{(d-1)}$. Perform interpolation in $\{D_{(d)}\}_{d=1}^{n/k}$ sequentially according to the order of d values, and interpolate only between adjacent subspaces (i.e., interpolate between $D_{(1)}$ and $D_{(2)}$, between $D_{(2)}$ and $D_{(3)}$, and so on).

When performing linear interpolation between adjacent subspaces, we should pair the k samples from the first subspace with an equal number of samples from the second subspace. The interpolation rules between adjacent subspaces are as follows:

1. Linear interpolation can only be performed between two samples belonging to different adjacent subspace sets.
2. Interpolation must be performed for each sample.
3. Participation of each sample point is restricted to a single interpolation instance.

The number of matching schemes is $k!$. As it is shown in Figure 1(c), we provide a matching method called K-Match. Suppose $\epsilon_i' \to 0$, this method can select a good-performing matching scheme $\{(\boldsymbol{x}_i^{(d)}, y_i^{(d)}), (\boldsymbol{x}_i^{(d+1)}, y_i^{(d+1)})\}_{i=1}^{k}$ from $k!$.

Assuming $\boldsymbol{x}$ and y are continuous variables. Given another hyperparameter η, the number of samples inserted using linear interpolation method between $D_{(d)}$ and $D_{(d+1)}$ is $\sum_{i=1}^{k} \lfloor \eta \cdot dist(\boldsymbol{x}_i^{(d)}, \boldsymbol{x}_i^{(d+1)}) \rfloor$. Taking $(\boldsymbol{x}_i^{(d)}, y_i^{(d)}) \in D_{(d)}$ and $(\boldsymbol{x}_i^{(d+1)}, y_i^{(d+1)}) \in D_{(d+1)}$ as example, $\{(\boldsymbol{x}_{(d,d+1)}^{(m,i)}, y_{(d,d+1)}^{(m,i)})\}_{m=1}^{\lfloor \eta \cdot dist(\boldsymbol{x}_i^{(d)}, \boldsymbol{x}_i^{(d+1)}) \rfloor}$ is the set of inserted samples, the linear interpolation formula is defined as:

$$\boldsymbol{x}_{(d,d+1)}^{(m,i)} = \boldsymbol{x}_i^{(d)} + m \cdot \frac{\boldsymbol{x}_i^{(d+1)} - \boldsymbol{x}_i^{(d)}}{\lfloor \eta \cdot dist(\boldsymbol{x}_i^{(d)}, \boldsymbol{x}_i^{(d+1)}) \rfloor + 1}, \tag{7}$$

$$y_{(d,d+1)}^{(m,i)} = y_i^{(d)} + m \cdot \frac{y_i^{(d+1)} - y_i^{(d)}}{\lfloor \eta \cdot dist(y_i^{(d)}, y_i^{(d+1)}) \rfloor + 1}. \tag{8}$$

After ASISO processing, the original dataset will be optimized. The main steps of the ASISO algorithm are summarized in Algorithm 1.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 18 August 2023 doi:10.20944/preprints202308.1378.v1

Algorithm 1: ASISO

Input: Data set $D = \{(x_i, y_i\}_{i=1}^{n}$; hyperparameters k and η
Output: Optimized data set D'

Perform unsupervised clustering using K-Space obtain: $D = \{D_s\}_{s=1}^{n/k}$, where $D_s = \{x_i^s, y_i^s\}_{i=1}^{k}$
Calculate culster centers of each cluster: $\{\overline{x}^s\}_{s=1}^{n/k}$
Obtain $\{D_{(d)}\}_{d=1^{n/k}}$ based on (5) and (6)
for $d = 1, \ldots, \frac{n}{k} - 1$ **do**
    Matching samples between $D_{(d)}$ and $D_{(d+1)}$ using K-Match obtain $\{(x_i^{(d)}, y_i^{(d)}), (x_i^{(d+1)}, y_i^{(d+1)})\}$
    for $i = 1, \ldots, k$ **do**
        Synthesize new samples $\{(x_{(d,d+1)}^{(m,i)}, y_{(d,d+1)}^{(m,i)})\}_{m=1}^{\lfloor \eta \cdot dist(x_i^{(d)}, x_i^{(d+1)}) \rfloor}$ using (7) and (8)
        $D' \leftarrow \{(x_{(d,d+1)}^{(m,i)}, y_{(d,d+1)}^{(m,i)})\}_{m=1}^{\lfloor \eta \cdot dist(x_i^{(d)}, x_i^{(d+1)}) \rfloor}$
    end
end

The assumptions of ASISO are as follows:

1. $f(\cdot)$ is a continuous function.
2. the linear fitting error $\epsilon_i' \to 0$.
3. x and y are continuous variables.

3.2. K-Space

The implementation of ASISO requires an unsupervised clustering method to partition the feature space into multiple subspaces, each containing k samples. Based on this, we propose the K-Space clustering method. The clustering method has the following performance:

1.Each subspace contains an equal number of samples, i.e., $D = \cup_{s=1}^{n/k} D_s, D_s = \{(x_i^s, y_i^s)\}_{i=1}^{k}$;
2.Each sample belongs to only one subset, i.e., $D_i \cap D_j = \emptyset, i \neq j$.

Maintaining continuity and similarity between adjacent subspaces is essential for synthesizing data via multiple linear interpolations in ASISO. Our objective is to minimize the linear fitting error ϵ_i', which helps to satisfy ASISO assumption 2 as much as possible.

To determine the sample set D_s for subspace $\mathcal{X}_s$, it is necessary to determine the first sample x_1^s in D_s.

$$x_1^s = \underset{x: x \in D, x \notin D_1, \ldots, D_{s-1}}{argmin} dist(x, \overline{x}^{s-1}), \tag{9}$$

where $s = 1, \ldots, \frac{n}{k}$, $\overline{x}^{s-1}$ is the cluster center of D_{s-1}, $\overline{x}^0 = x_0$. We define $D_s = \{x_1^s\}$ and determine x_d^s as follows:

$$x_d^s = \underset{x: x \in D, x \notin D_1, \ldots, D_s}{argmin} dist(x, \overline{x}^s), \tag{10}$$

where $d = 2, \ldots, k$. Obtain x_d^s and update $D_s \leftarrow D_s \cup \{x_d^s\}$.

The main steps of the K-Space algorithm are summarized in Algorithm 2.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 18 August 2023 doi:10.20944/preprints202308.1378.v1

Algorithm 2: K-Space

Input: Data set $D = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}$; hyperparameter k
Output: $\{D_s\}_{s=1}^{n/k}$

Obtain the sample minimum point $\boldsymbol{x}_0$
for $s = 1, \ldots, \frac{n}{k}$ **do**
  $\boldsymbol{x}_1^s = \underset{\boldsymbol{x}:\boldsymbol{x}\in D, \boldsymbol{x}\notin D_1,\ldots,D_{s-1}}{argmin} dist(\boldsymbol{x}, \overline{\boldsymbol{x}}^{s-1})$
  $D_s = \{\boldsymbol{x}_1^s\}$
  for $d = 2, \ldots, k$ **do**
    $\boldsymbol{x}_d^s = \underset{\boldsymbol{x}:\boldsymbol{x}\in D, \boldsymbol{x}\notin D_1,\ldots,D_s}{argmin} dist(\boldsymbol{x}, \overline{\boldsymbol{x}}^{s})$
    $D_s \leftarrow D_s \cup \{\boldsymbol{x}_d^s\}$
  end
end

3.3. K-Match

We can calculate the total error of the matching scheme to measure the quality of the scheme, for the sake of simplicity, let $\mathcal{X} = R$, as follow:

$$\sum_{i=1}^{k} S(x_i^{(d)}, x_i^{(d+1)}) = \sum_{i=1}^{k} \int_{x_i^{(d)}}^{x_i^{(d+1)}} |f(x) - L_i(x)| dx, \tag{11}$$

where $L_i(x)$ is the linear expression passing through the points $(x_i^{(d)}, y_i^{(d)})$ and $(x_i^{(d+1)}, y_i^{(d+1)})$.

Theorem 1. *Let $\mathcal{X}_{(d)}$ and $\mathcal{X}_{(d+1)}$ be two adjacent subspaces, the datasets corresponding to different subspaces are $D_{(d)}, D_{(d+1)}$, and $(x_i^{(d)}, y_i^{(d)}) \in D_{(d)}, (x_i^{(d+1)}, y_i^{(d+1)}) \in D_{(d+1)}$. Consider the model $y_i = f(x_i) + \epsilon_i$, let $\epsilon_i^{(d)} = y_i^{(d)} - f(x_i^{(d)})$. For $\forall i = 1, 2, \ldots, k$, suppose that $\epsilon_i' \to 0$, then $E\left(\frac{S(x_i^{(d)}, x_i^{(d+1)})}{|x_i^{(d)} - x_i^{(d+1)}|}\right) < E\left(\frac{|\epsilon_i^{(d)}| + |\epsilon_i^{(d+1)}|}{2}\right)$.*

Proof of Theorem 1. Since $\epsilon_i' \to 0$, and according to (3), the model can be transformed into:

$$y_i = g(x_i) + \epsilon_i,$$

where $g(\cdot)$ is a linear function. According to (11), it follows that:

$$S(x_i^{(d)}, x_i^{(d+1)}) = \int_{x_i^{(d)}}^{x_i^{(d+1)}} |g(x) - L_i(x)| dx.$$

When $\epsilon_i^{(d)} \cdot \epsilon_i^{(d+1)} < 0$, let (x', y') be the intersection point between $y = L_i(x)$ and $y = g(x)$. We can simplify $S(x_i^{(d)}, x_i^{(d+1)})$ using basic geometric area calculations, and according to Law of Iterated Expectations (LIE):

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 18 August 2023 doi:10.20944/preprints202308.1378.v1

$$
\begin{aligned}
E\left(\frac{S(x_i^{(d)}, x_i^{(d+1)})}{|x_i^{(d)} - x_i^{(d+1)}|}\right) &= \frac{E\left(S(x_i^{(d)}, x_i^{(d+1)})|\epsilon_i^{(d)} \cdot \epsilon_i^{(d+1)} \geq 0\right) P(\epsilon_i^{(d)} \cdot \epsilon_i^{(d+1)} \geq 0)}{|x_i^{(d)} - x_i^{(d+1)}|} \\
&+ \frac{E\left(S(x_i^{(d)}, x_i^{(d+1)})|\epsilon_i^{(d)} \cdot \epsilon_i^{(d+1)} < 0\right) P(\epsilon_i^{(d)} \cdot \epsilon_i^{(d+1)} < 0)}{|x_i^{(d)} - x_i^{(d+1)}|} \\
&= \frac{E\left(|\epsilon_i^{(d)}| + |\epsilon_i^{(d+1)}|\right) P(\epsilon_i^{(d)} \cdot \epsilon_i^{(d+1)} \geq 0)}{2} \\
&+ \frac{\left(h_1 \cdot E|\epsilon_i^{(d)}| + h_2 \cdot E|\epsilon_i^{(d+1)}|\right) P(\epsilon_i^{(d)} \cdot \epsilon_i^{(d+1)} < 0)}{2},
\end{aligned}
$$

where $h_1 = \frac{|x' - x_i^{(d)}|}{|x_i^{(d)} - x_i^{(d+1)}|}, h_2 = \frac{|x' - x_i^{(d+1)}|}{|x_i^{(d)} - x_i^{(d+1)}|}$. Since $P(\epsilon_i^{(d)} \cdot \epsilon_i^{(d+1)} \geq 0) + P(\epsilon_i^{(d)} \cdot \epsilon_i^{(d+1)} < 0) = 1$, and $h_1 + h_2 = 1$, it follows that $E\left(\frac{S(x_i^{(d)}, x_i^{(d+1)})}{|x_i^{(d)} - x_i^{(d+1)}|}\right) < E\left(\frac{|\epsilon_i^{(d)}| + |\epsilon_i^{(d+1)}|}{2}\right)$.

□

If our approach is to randomly select a matching scheme, the validity of this method can be proved by Theorem 1. However, randomly selecting a matching scheme does not guarantee the uniqueness of the results, and it also does not guarantee that we will necessarily select the good-performing matching scheme. We found that for $x_i^{(d)}$ and $x_i^{(d+1)}$, if $\epsilon_i^{(d)} \cdot \epsilon_i^{(d+1)} < 0$, there will be a better interpolation effect.

Theorem 2. *Let* $y_i^{(d)} = f(x_i^{(d)}) + \epsilon_i^{(d)}, y_i^{(d+1)} = f(x_i^{(d+1)}) + \epsilon_i^{(d+1)}$. *Suppose that* $\epsilon_i' \to 0$, *then we have* $E\left(S(x_i^{(d)}, x_i^{(d+1)})|\epsilon_i^{(d)} \cdot \epsilon_i^{(d+1)} < 0\right) < E\left(S(x_i^{(d)}, x_i^{(d+1)})|\epsilon_i^{(d)} \cdot \epsilon_i^{(d+1)} \geq 0\right)$.

Proof of Theorem 2. Since $\epsilon_i' \to 0$, based on the proof of theorem 1, we have:

$$
\begin{aligned}
&E\left(S(x_i^{(d)}, x_i^{(d+1)})|\epsilon_i^{(d)}\epsilon_i^{(d+1)} \geq 0\right) \\
&= \frac{|x_i^{(d)} - x_i^{(d+1)}| \cdot E|\epsilon_i^{(d)}| + |x_i^{(d)} - x_i^{(d+1)}| \cdot E|\epsilon_i^{(d+1)}|}{2},
\end{aligned}
$$

$$
\begin{aligned}
&E\left(S(x_i^{(d)}, x_i^{(d+1)})|\epsilon_i^{(d)}\epsilon_i^{(d+1)} < 0\right) \\
&= \frac{|x' - x_i^{(d)}| \cdot E|\epsilon_i^{(d)}| + |x' - x_i^{(d+1)}| \cdot E|\epsilon_i^{(d+1)}|}{2}.
\end{aligned}
$$

It follows that

$$
E\left(S(x_i^{(d)}, x_i^{(d+1)})|\epsilon_i^{(d)}\epsilon_i^{(d+1)} < 0\right) < E\left(S(x_i^{(d)}, x_i^{(d+1)})|\epsilon_i^{(d)}\epsilon_i^{(d+1)} \geq 0\right).
$$

□

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 18 August 2023 doi:10.20944/preprints202308.1378.v1

According to Theorem 2, we can match samples with opposite signs of ϵ_i to achieve a good data synthesis effect. Therefore, the core idea of K-match is to make a judgment on the positive or negative sign of ϵ_i for each sample, and then interpolate the samples with opposite signs as much as possible.

In K-Match, we need to choose an appropriate linear regression method to fit the dataset $D_{(d)} \cup D_{(d+1)}$ based on the performance of the noise. For example, Lasso regression, Locally Weighted Linear Regression (LWLR) [23], and other methods can be used [24,25]. In our experiments, we use OLS or SVR method to fit and obtain $\hat{g}(\cdot)$. Specially, the kernel function is Linear in SVR. According to (3), and suppose the linear fitting error $\epsilon_i' \to 0$, for dataset $D_{(d)} \cup D_{(d+1)}$, we have:

$$\epsilon_i = y_i - \hat{g}(\boldsymbol{x}_i). \tag{12}$$

Then, we sort the samples in dataset $D_{(d)}$ in ascending order according to the value of ϵ_i, and obtain $\{(\boldsymbol{x}_i^{(d)}, y_i^{(d+1)})\}_{i=1}^k$; we sort the samples in dataset $D_{(d+1)}$ in descending order and obtain $\{(\boldsymbol{x}_i^{(d+1)}, y_i^{(d+1)})\}_{i=1}^k$. As it is shown in Figure 1(d), combine the sorted datasets $D_{(d)}$ and $D_{(d+1)}$ into the matching scheme $\{(\boldsymbol{x}_i^{(d)}, y_i^{(d)}), (\boldsymbol{x}_i^{(d+1)}, y_i^{(d+1)})\}_{i=1}^k$.

Algorithm 3: K-Match

Input: Subset $D_{(d)} = \{(\boldsymbol{x}_i^{(d)}, y_i^{(d)})\}_{i=1}^k, D_{(d+1)} = \{(\boldsymbol{x}_i^{(d+1)}, y_i^{(d+1)}\}_{i=1}^k$

Output: Matching scheme $\{(\boldsymbol{x}_i^{(d)}, y_i^{(d)}), (\boldsymbol{x}_i^{(d+1)}, y_i^{(d+1)})\}_{i=1}^k$

1 Fitting the dataset $D_{(d)} \cup D_{(d+1)}$ and obtain $\hat{g}(\cdot)$
2 Obtain $\{\epsilon_i\}$ using (10)
3 Sorting the samples in $D_{(d)}$ and $D_{(d+1)}$ according to the value of ϵ_i, obtain $D'_{(d)}$ and $D'_{(d+1)}$
4 Combine $D'_{(d)}$ and $D'_{(d+1)}$ into $\{(\boldsymbol{x}_i^{(d)}, y_i^{(d)}), (\boldsymbol{x}_i^{(d+1)}, y_i^{(d+1)})\}_{i=1}^k$

3.4. *Supplements*

The proposed method can effectively expand the size of dataset and adjust the dataset structure, reducing the proportion of samples that deviate significantly from the actual distribution, thereby improving the model generalization. Refer to Figure 2.

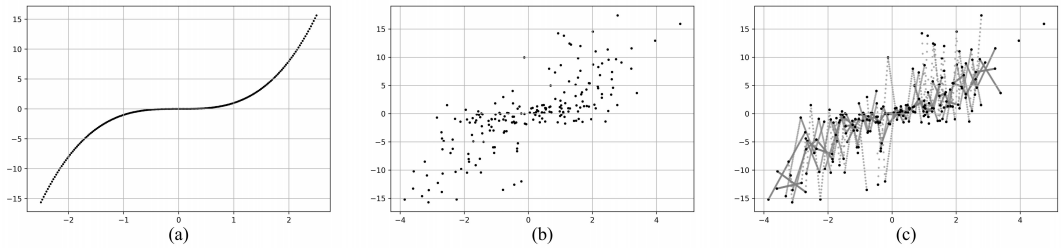

Figure 2. The synthetic data by ASISO. (a) The true relationship between x and y, $y = x^3$. The sample size is 200. (b) Adding Gaussian noise. (c) Let $k = 6, \eta = 100$, processing with ASISO, the sample size was increased to 3808.

The supplements about ASISO are given below:

1. The choice of the hyperparameter k is crucial as different datasets require different values of k. Conversely, hyperparameter η tends to exhibit better performance as its value increases, which will be illustrated in the following experimental results.

2. It is necessary to normalize the data if there is a significant difference in the dimensional scale between the features of the data. It avoids the issue of generating an excessive number of samples.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 18 August 2023 doi:10.20944/preprints202308.1378.v1

3. In most cases, n/k is not an integer, and for the excess samples, we usually have two solutions of handling them. The first one is to use the LOF algorithm [26] to filter out the excess samples that will not participate in the ASISO, as it is shown in Figure 2(c). Another solution is to treat the excess samples as a dataset D' of a subspace, $D' = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n \bmod k}$.When interpolating between D' and other subspaces D'', choose an appropriate linear regression method to fit the dataset $D' \cup D''$, and obtain $\hat{g}(\cdot)$. Then, use the same method to sort D' and D''. Only $n \bmod k$ interpolations are performed, with each sample in D' being interpolated, while for D'', only $n \bmod k$ samples are interpolated. Moreover, interpolate the samples with opposite signs of ϵ_i as much as possible, as it is shown in Figure 3.

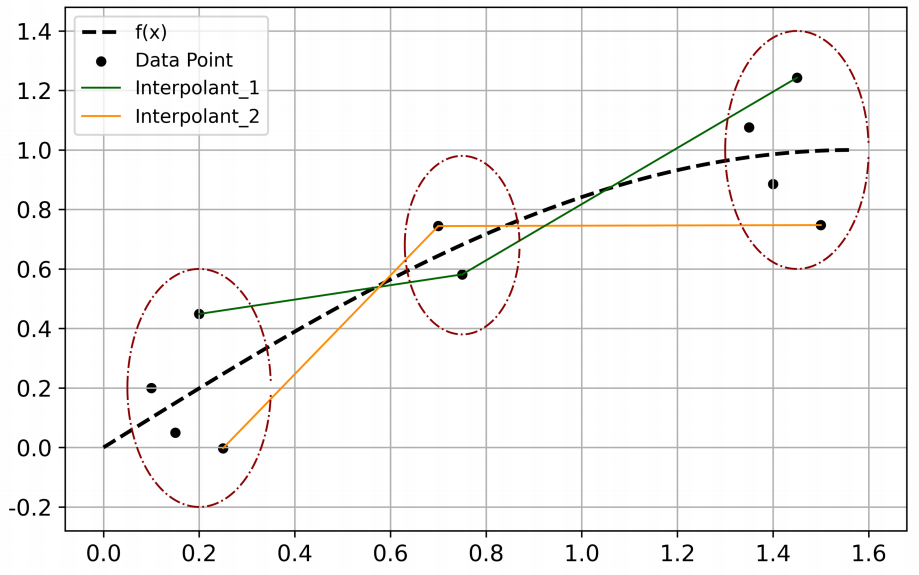

Figure 3. The synthetic data by ASISO. (a) The true relationship between x and y, $y = x^3$. The sample size is 200. (b) Adding Gaussian noise. (c) Let $k = 6, \eta = 100$, processing with ASISO, the sample size was increased to 3808.

4. Experiments

For Artificial data, as $f(\cdot)$ can be known, we investigated the hyperparameter selection, and optimization performance after processing by ASISO. For benchmark data, we studied the prediction performance using this method.

4.1. Artificial Data Sets

Some verification indicators, the proportion of samples with ϵ_i greater than α, and the mean square error (MSE), can be selected to test the optimization effect of the ASISO on the original sample after processing, as follows:

$$p(\alpha) = \frac{1}{n}\sum_{i=1}^{n} I(|f(\boldsymbol{x}_i) - y_i| > \alpha), \tag{13}$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(f(\boldsymbol{x}_i) - y_i)^2. \tag{14}$$

For artificial data sets, $\{\boldsymbol{x}_i\}_{i=1}^{n}$ are generated by $N_p(\boldsymbol{0}, \boldsymbol{E})$. Let $\boldsymbol{W}_1 \in \mathbb{R}_{(p \times p_1)}$ and $\boldsymbol{W}_2 \in \mathbb{R}_{(p_1 \times 1)}$, where all elements of both $\boldsymbol{W}_1$ and $\boldsymbol{W}_2$ are independently and identically distributed as $N(0, 1)$. Consider $y_i = f(\boldsymbol{x}_i) + \epsilon_i = tanh(\boldsymbol{x}_i' \boldsymbol{W}_1)\boldsymbol{W}_2 + \epsilon_i$.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 18 August 2023 doi:10.20944/preprints202308.1378.v1

$$tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}.$$

For a given data set, we typically cannot ascertain the distribution of noise. Hence, we construct datasets that contain unknown noises [12]. The unknown noises are simulated by mixture noises that contain uniform noises and Gaussian noises. The generation of ϵ_i referenced to [27]. We have fixed the random effects of the generated datasets; experiments are repeated 100 times in each artificial dataset. Table 1 shows 6 artificial datasets.

Table 1. Artificial Data Sets.

Artificial Data Sets	ϵ_i Distribution	Samples	(P, P_1)
D_1	20%-N(0,64) 30%-U(-8,8) 50%-N(0,0.04)	500	(5,3)
D_2	20%-N(0,64) 30%-U(-8,8) 50%-N(0,0.04)	200	(5,3)
D_3	20%-N(0,64) 30%-U(-8,8) 50%-N(0,0.04)	1500	(5,3)
D_4	20%-N(0,64) 30%-U(-8,8) 50%-N(0,0.04)	500	(1,3)
D_5	20%-N(0,64) 30%-U(-8,8) 50%-N(0,0.04)	500	(20,10)
D_6	40%-N(0,64) 45%-U(-8,8) 15%-N(0,0.04)	500	(5,3)

4.1.1. Hyperparameter Selection

First, we investigate the selection of parameter k. To ensure that the number of feature subspaces is sufficient. We set the hyperparameter $\eta = 1$, let $k = 1, 2, \ldots, \lfloor \frac{n}{2} \rfloor$. Calculate the changes in MSE of the datasets at different values of k, obtain $k' = \underset{k:k=1,2,\ldots,\lfloor \frac{n}{2} \rfloor, \eta=1}{argmin} MSE$. The change trend of the MSE index before and after the ASISO processing is shown in Figure 4. Then, let $\eta = 1, 2, \ldots, 30, k = k'$, calculate the changes in MSE of the datasets (Figure 5), obtain $\eta' = \underset{\eta:\eta=1,2,\ldots,30, k=k'}{argmin} MSE$. At last, let $k = k', \eta = 1$ and η', calculate the changes in $p(\alpha)$ of the artificial datasets (Figure 6).

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 18 August 2023 doi:10.20944/preprints202308.1378.v1

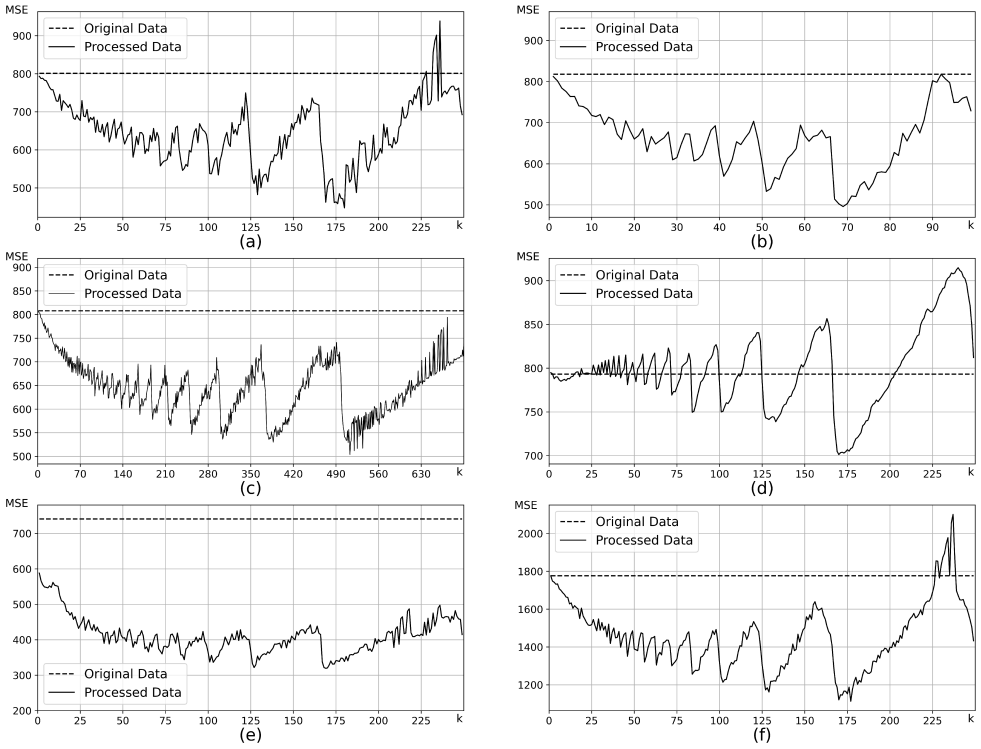

Figure 4. Changes in the MSE under different k values.

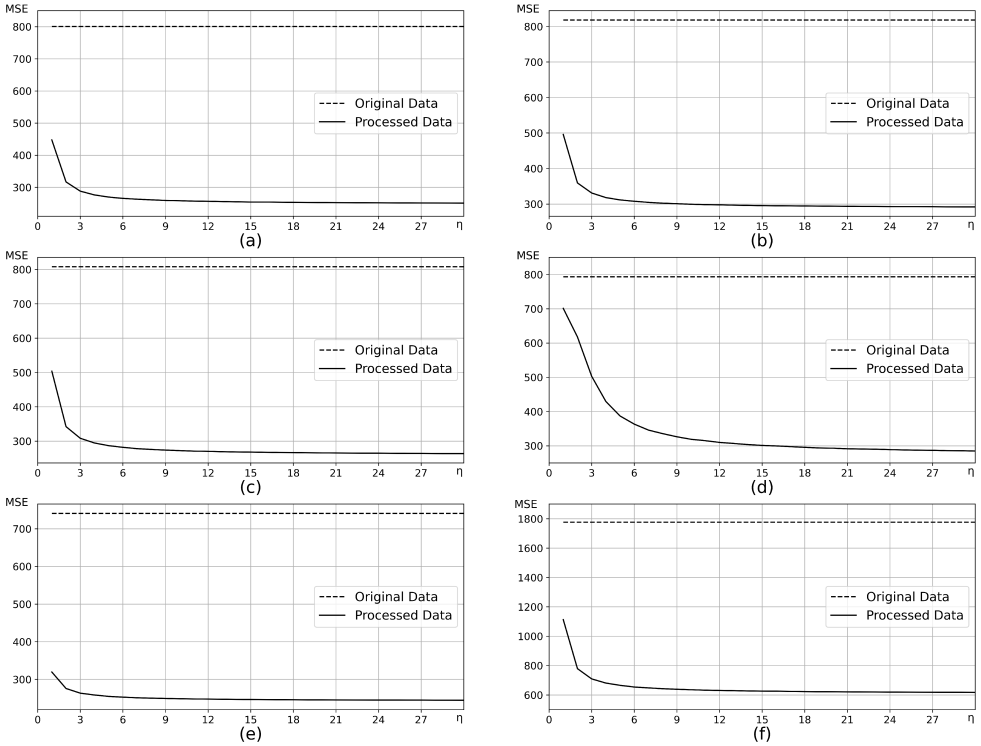

Figure 5. Changes in the MSE under different η values.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 18 August 2023 doi:10.20944/preprints202308.1378.v1

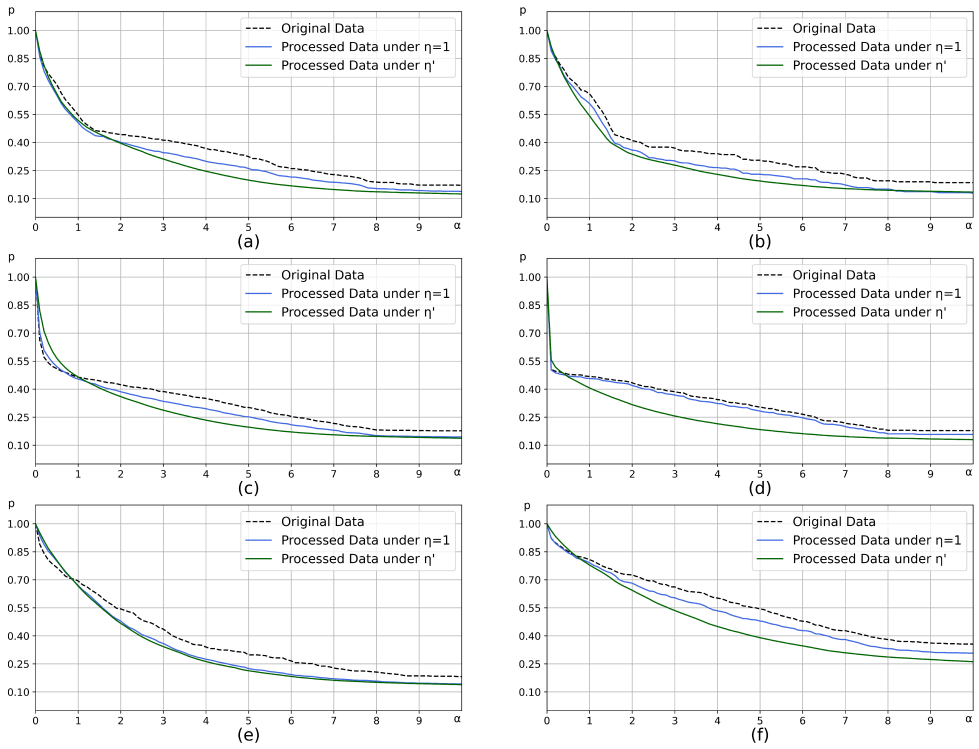

Figure 6. Changes in the $p(\alpha)$ under different η values.

As can be seen from Figure 4, ASISO demonstrates good optimization performance for datasets with varying sample sizes or feature dimensions. Experimental results also show that the performance of ASISO does not decline dramatically as the content of noise with large variances increases. Hence, it can deal with unknown noise better and has good robustness. In addition, it can be shown that ASISO exhibits good optimization performance for all datasets from experimental results, it is also a stable data synthesis method. As can be seen from Figure 5, the hyperparameter η has a generally monotonically decreasing relationship with MSE, and the larger the value of η, the more significant the effect. From Figure 6, it can be observed that ASISO can adaptively adjust the sample structure, which reduces the proportion of samples with large errors.

4.1.2. Comparison of Optimization Performance

To verify the optimization performance of ASISO, it is compared three methods: piecewise linear interpolation, linear extrapolation, and nearest neighbor interpolation. Specifically, let $k = k', \eta = \eta'$ in ASISO. Based on the given dataset, we use the samples generated by ASISO as interpolation points to calculate the output values in linear extrapolation and nearest neighbor interpolation. Moreover, let $k = 1$ and $\eta = \eta'$ in ASISO, we can regard it as piecewise linear interpolation. The experimental results show that ASISO has the smallest MSE among all methods for each dataset (Figure 7).

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 18 August 2023 doi:10.20944/preprints202308.1378.v1

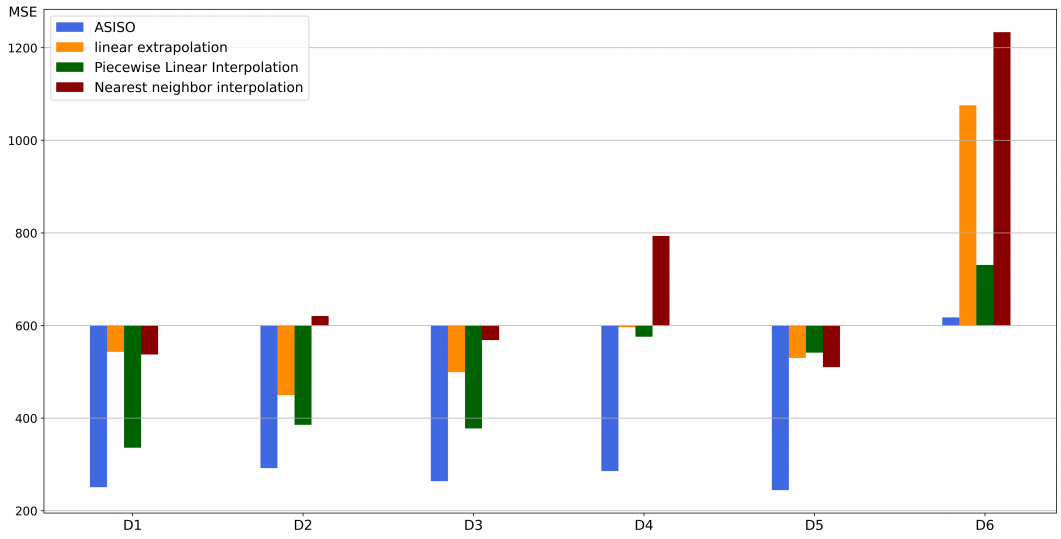

Figure 7. Comparison of MSE on artificial data sets.

4.2. Benchmark Data Sets

To verify the prediction performance of ASISO, 4 benchmark data sets are used [28,29]. We partitioned the dataset into training and testing sets using a 7:3 ratio and normalize the data using the min-max normalization method. We preprocessed the training data using ASISO, train multiple machine learning models on the training set, and evaluated the prediction performance of the models on the testing set. The evaluation metric we used was mean absolute error (MAE).

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|. \tag{15}$$

In addition, we removed features that cannot be directly used, such as "Datetime" for bike sharing demand, "Month" for forest fires, and so on. We choose K-Nearest Neighbor (KNN), Random Forest (RF), Gradient Boosting Decision Tree (GBDT), Multilayer Perceptron (MLP) and Support Vector Regression (SVR) as the machine learning prediction models. Specifically, the kernel function is Radial Basis Function (RBF) in SVR. Moreover, set the number of hidden layers to 3 for the MLP and use different numbers of neurons based on the input dimensionality and sample size.

Table 2. Experimental Results on Benchmark Data Sets.

Data sets	Processing	Hyperparameter	Testing MAE (10^{-2})				
			KNN	RF	MLP	SVR	GBDT
Bike Sharing	-	-	2.20	**0.12**	0.54	4.26	0.23
	ASISO	$k = 150, \eta = 10$	2.00	**0.06**	0.25	4.12	0.19
Facebook	-	-	6.60	2.25	8.19	7.34	**1.33**
	ASISO	$k = 2, \eta = 10$	6.41	1.59	2.02	7.36	**1.12**
Air Quality	-	-	2.99	**2.57**	4.80	3.87	2.58
	ASISO	$k = 20, \eta = 100$	2.89	2.55	**2.08**	3.84	2.77
Forest Fires	-	-	**4.06**	4.93	10.58	7.36	4.49
	ASISO	$k = 10, \eta = 100$	**3.89**	4.32	4.80	10.05	4.14

The experimental results of the five models are shown in Table 2. It is evident that ASISO performs well on each benchmark dataset. This method is highly applicable to all five models, and in most cases, it can improve the predictive performance. It is worth mentioning that the dataset contains many categorical features, which are not continuous variables. Moreover, for sparse samples (Facebook Metrics and Forest Fires), it is difficult to guarantee that the linear fitting error $\epsilon_i' \to 0$

when interpolating between subspaces. This indicates that even if there are violations of the ASISO assumptions in practical applications, this method may still achieve good optimization results. This further demonstrates that it is robust and stable.

5. Conclusions

In this paper, we propose a data synthesis method, ASISO, which can adaptively adjust the size of the dataset, and the expanded data typically contain minimal error with actual. Moreover, it can adjust the structure of the samples, which can significantly reduce the proportion of samples with large errors. The experimental results on artificial data sets demonstrate that ASISO can optimize the samples, and compared to other methods, the generated data using this method has smaller errors. It can deal with unknown noise better and has good robustness. The results on benchmark data sets show the proposed method is applicable to many machine models, and in most cases, it can improve the model generalization.

References

1. ALRikabi H T S, Hazim H T. Enhanced data security of communication system using combined encryption and steganography. iJIM, **2021**, 15(16): 145.
2. Kollias D. ABAW: learning from synthetic data & multi-task learning challenges. European Conference on Computer Vision. Cham: Springer Nature Switzerland, **2022**: 157-172.
3. Mahesh B. Machine learning algorithms-a review. International Journal of Science and Research (IJSR).[Internet], **2020**, 9(1): 381-386.
4. Lepot M, Aubin J B, Clemens F H L R. Interpolation in time series: An introductive overview of existing methods, their performance criteria and uncertainty assessment. Water, **2017**, 9(10): 796.
5. Chlap P, Min H, Vandenberg N, et al. A review of medical image data augmentation techniques for deep learning applications. Journal of Medical Imaging and Radiation Oncology, **2021**, 65(5): 545-563.
6. Shorten C, Khoshgoftaar T M. A survey on image data augmentation for deep learning. Journal of big data, **2019**, 6(1): 1-48.
7. Chawla N V, Bowyer K W, Hall L O, et al. SMOTE: synthetic minority over-sampling technique. Journal of artificial intelligence research, **2002**, 16: 321-357.
8. Dablain D, Krawczyk B, Chawla N V. DeepSMOTE: Fusing deep learning and SMOTE for imbalanced data. IEEE Transactions on Neural Networks and Learning Systems, **2022**.
9. Han H, Wang W Y, Mao B H. Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning.International conference on intelligent computing. Berlin, Heidelberg: Springer Berlin Heidelberg, **2005**: 878-887.
10. Bunkhumpornpat C, Sinapiromsaran K, Lursinsap C. Safe-level-smote: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem. Advances in Knowledge Discovery and Data Mining: 13th Pacific-Asia Conference, PAKDD 2009 Bangkok, Thailand, April 27-30, 2009 Proceedings 13. Springer Berlin Heidelberg, **2009**: 475-482.
11. Ha T, Dang T K, Dang T T, et al. Differential privacy in deep learning: an overview. 2019 International Conference on Advanced Computing and Applications (ACOMP). IEEE, **2019**: 97-102.
12. Meng D, De La Torre F. Robust matrix factorization with unknown noise. Proceedings of the IEEE international conference on computer vision. **2013**: 1337-1344.
13. Raghunathan T E. Synthetic data. Annual review of statistics and its application, **2021**, 8: 129-140.
14. Sibson R. A brief description of natural neighbour interpolation. Interpreting multivariate data, **1981**: 21-36.
15. Tachev G T. Piecewise linear interpolation with nonequidistant nodes. Numerical Functional Analysis and Optimization, **2000**, 21(7-8): 945-953.
16. Blu T, Thévenaz P, Unser M. Linear interpolation revitalized. IEEE Transactions on Image Processing, **2004**, 13(5): 710-719.
17. Berrut J P, Trefethen L N. Barycentric lagrange interpolation. SIAM review, **2004**, 46(3): 501-517.
18. Musial J P, Verstraete M M, Gobron N. Comparing the effectiveness of recent algorithms to fill and smooth incomplete and noisy time series. Atmospheric chemistry and physics, **2011**, 11(15): 7905-7923.

19. Fornberg B, Zuev J. The Runge phenomenon and spatially variable shape parameters in RBF interpolation. Computers & Mathematics with Applications, **2007**, 54(3): 379-398.
20. Rabbath C A, Corriveau D. A comparison of piecewise cubic Hermite interpolating polynomials, cubic splines and piecewise linear functions for the approximation of projectile aerodynamics. Defence Technology, **2019**, 15(5): 741-757.
21. Habermann C, Kindermann F. Multidimensional spline interpolation: Theory and applications. Computational Economics, bm2007, 30: 153-169.
22. Ganzburg M I. The Bernstein constant and polynomial interpolation at the Chebyshev nodes. Journal of Approximation Theory, **2002**, 119(2): 193-213.
23. Cleveland W S. Robust locally weighted regression and smoothing scatterplots. Journal of the American statistical association, **1979**, 74(368): 829-836.
24. Lichti D D, Chan T O, Belton D. Linear regression with an observation distribution model. Journal of geodesy, **2021**, 95: 1-14.
25. Liu C, Li B, Vorobeychik Y, et al. Robust linear regression against training data poisoning. Proceedings of the 10th ACM workshop on artificial intelligence and security. **2017**: 91-102.
26. Breunig M M, Kriegel H P, Ng R T, et al. LOF: identifying density-based local outliers. Proceedings of the 2000 ACM SIGMOD international conference on Management of data. **2000**: 93-104.
27. Guo Y, Wang W, Wang X. A robust linear regression feature selection method for data sets with unknown noise. IEEE Transactions on Knowledge and Data Engineering, **2021**, 35(1): 31-44.
28. Cukierski W. Bike Sharing Demand. Kaggle, https://kaggle.com/competitions/bike-sharing-demand, **2014**.
29. Dua D, Craff C. UCI Machine Learning Repository, http://archive.ics.uci.edu/ml, **2017**.