一种鲁棒的基于特征子空间插值的数据合成方法

摘 要

数据噪音是指在数据收集、处理或传输过程中引入的任何错误或无关信息，它对预测模型和数据分析的准确性构成了重大挑战。因此，解决数据噪音问题不仅有助于提高模型的预测准确度，还可以增强模型的鲁棒性，使其更能适应实际应用场景的需求。然而，目前的数据噪音处理方法，如数据清洗、数据平滑、噪音识别技术等，虽然可以减少数据中的随机波动，但也可能导致数据过度泛化。这种过度泛化可能会模糊数据中的重要特征和细节，从而影响数据分析与预测的准确性。在这种背景下，本文提出了一种创新的对于含噪数据集处理的数据合成方法，基于特征子空间插值的思想，命名为 RSIS（Robust Subspace Interpolation Synthesis）。这种方法可以从两方面对样本进行优化：首先，RSIS 方法可以合成误差较小的样本以降低原始数据集平均误差，并且减小了大误差样本的占比；其次，由于 RSIS 方法是在特征子空间之间进行多次等距且线性插值的，该方法可以提升原始数据的多样性以及代表性，从而增强了变量之间的函数映射关系。通过这两方面的作用，RSIS 达到了优化样本的目的。与传统的含噪数据集噪声处理方法相比，RSIS 更适用于机器学习领域的样本噪音处理，即使是在处理含有未知复杂噪音的高维数据时，也可以有效提升机器学习模型的泛化能力。此外，本文还将 RSIS 方法结合卷机神经网络应用到计算机视觉领域，通过实验结果可知，RSIS 方法同样可以有效提升图像分类模型的泛化能力。

关键词：多阶段最小权匹配；含噪数据集；鲁棒的；特征子空间

一、绪论

在当今的数据驱动时代，数据质量对于数据科学领域的影响日益显著，尤其是在大数据环境下，数据噪声问题成为了一个不可忽视的挑战。数据噪声指的是数据中的任何不准确、不相关或误导性信息，这可能源于多种原因，如仪器误差、数据录入错误、传输问题，甚至是数据采集过程中的系统性偏差。在大多数实际问题中，噪声的分布都是复杂且未知的，这些噪声可能导致数据分析和机器学习模型的性能大打折扣，尤其是在需要高度精准预测和分析的应用场景中，如医疗影像分析、金融市场预测、环境监测等。在这些领域，即使是微小的数据误差也可能导致严重的后果。因此，有效地识别和处理数据中的噪声，已成为提高预测模型准确性和可靠性的关键步骤[1–3]。

为了解决数据噪音的问题，研究者们开发了多种方法，包括数据清洗、数据平滑和异常值检测等。这些方法主要集中于减轻或移除数据中的噪声，但它们通常不会从根本上提升数据质量。例如，数据清洗过程可能难以判断数据噪声与真实的异常值。异常值在某些情况下可能是数据分析中的重要信息源，但数据清洗可能错误地将其视为噪声并将其移除，这种区分的困难可能导致关键信息的丢失。同样的，数据平滑技术，如移动平均或高斯平滑，虽然可以减少数据中的随机波动，但也可能导致数据过度泛化。这种过度泛化可能会模糊数据中的重要特征和细节，从而影响数据分析的准确性。此外，当处理复杂以及高维度的噪音数据集时，传统的噪声处理方法具有局限性。在高维数据中，噪声和有用信息之间的界限可能更加模糊，有效地识别和处理噪声变得更加困难，这就会导致在降噪过程中可能误删掉有价值的数据，或者保留过多的噪声。

在这种背景下，本文提出了一种创新的可对含噪数据集优化的数据合成方法，基于特征子空间插值的思想，命名为 RSIS（Robust Subspace Interpolation Synthesis）。这种方法可以从两方面对原始样本进行优化：首先在不损失样本原始信息的前提下，通过合成误差较小的样本，降低数据集整体的噪音水平，并减少高噪音样本的比例；其次，RSIS 通过特征子空间等距且线性的插值，合成了更多具有代表性、多样性的样本，强化了数据集变量之间真实的函数映射关系。与传统的含噪数据集噪声处理方法相比，RSIS 更适用于机器学习领域的样

本噪音处理，即使是在处理含有未知复杂噪音的高纬数据时，该方法也可以有效提升机器学习模型的泛化能力。

RSIS 方法包含几个关键步骤。首先，对原始数据集进行迭代层次聚类，该方法可以将原始数据集划分为多个样本量相近的子集，这一步骤会将原始特征空间划分为多个具有相近样本量的子空间；其次，根据聚类结果，RSIS 方法引入了旅行商问题的思想对特征子空间进行排序；随后，本文结合了软参数多任务学习的机制，根据排序结果对相邻子空间之间进行考虑了全局信息的线性拟合；最后，本文提出了一种创新的多阶段最小权匹配启发式方法，通过该方法可以得到一种较优的插值匹配策略，通过该策略对样本进行线性插值可以生成具有较小误差的样本。

目前许多研究都假设样本的噪音是同分布的，但是这种情况对于实际应用中是较为少见的，本文从理论上证明了 RSIS 方法即使面对不同分布的噪音样本依旧可以表现出很好的优化效果。基于噪音不同分布的场景，本文还模拟了样本中含有未知且不同分布的复杂噪音的情况，RSIS 方法在实际模拟实验中依旧可以具有良好的鲁棒性，并且对样本进行显著的优化。此外，本文还将 RSIS 方法结合卷机神经网络应用到计算机视觉领域，通过实验结果可知，RSIS 方法同样可以有效提升图像分类模型的泛化能力。

二、文献综述

目前已有的数据合成方法数据合成的目的是通过生成非真实但统计上具有代表性的数据来增强、扩展或模拟数据集。这些方法在数据稀缺、隐私保护或模型训练等场景中被广泛应用。目前，数据合成的主要方法包括：

1.基于规则的方法。这类方法依赖于预定义的规则或算法来修改现有数据或生成新的数据实例。例如，线性插值和多项式插值用于在已知数据点之间生成新点。图像数据增强，如旋转、缩放、翻转等，也属于此类[14]。此外，模拟法如蒙特卡洛模拟[15]和代理模型则通过随机抽样或简化模型来模拟数据分布。

2.基于统计的方法。这些方法使用统计模型来估计数据分布，并据此生成数据。这类方法可以进一步细分为参数化方法和非参数化方法。在参数化方法中，如高斯混合模型[16]和贝叶斯网络，数据合成过程依赖于特定的分布假设。这些模型假定数据遵循某种已知的概率分布，如正态分布或其他参数化分布，然后利用这些分布的参数来生成新的数据点。非参数化方法，如核密度估计、直方图方法以及 SMOTE 方法和其变体，则不依赖于固定的分布模型[17]。它们直接从数据本身推断出数据分布的形状，而不是依赖于预先定义的数学形式。

3. 基于机器学习的方法，尤其是生成对抗网络（GANs）和变分自编码器（VAEs），通过学习数据分布来生成新数据。GANs 通过对抗过程训练生成器和判别器[18]，而 VAEs 使用编码器-解码器架构来学习数据的潜在表示。自回归模型，如 PixelRNN 和 PixelCNN，通过学习数据中的像素依赖关系来生成图像[19]。

4. 最后，基于混合方法的数据合成结合了以上方法的优点，使用多种技术来生成数据[20]。这包括模型集成，即结合多个模型的输出来生成数据，以及多阶段生成，即先使用一种方法生成初步数据，再用另一种方法进行精细化处理。

目前已有的数据合成方法对于噪音数据集并不具有良好的鲁棒性，因为大部分数据合成方法的设计初衷并不是用于处理样本噪音的问题，如 SMOTE 算法，其目的是为了针对分类任务用于平衡样本数据；以及差分隐私算法是一种旨在提供强大隐私保护的技术，同时允许对数据集进行有用的统计分析。RSIS 作为一种创新的基于混合方法的数据合成方法，不仅解决了含噪数据集的特定问题，也通过创新的合成技术来优化数据质量，进而提升机器学习模型在处理

复杂数据时的性能和泛化能力。

三、基于特征子空间插值的数据合成方法

RSIS 方法分为四个步骤：

1.通过无监督聚类算法将原始特征空间划分为多个子空间，RSIS 方法要求这些子空间的样本数量近乎相等;

2.随后对子空间之间进行排序，该部分以最小化距离总和为目标将子空间插值排序问题转化为旅行商问题进行求解;

3.RSIS 方法融合了软参数共享机制的原理，对相邻子空间中的样本进行线性拟合。在这一框架下，虽然每个任务都拥有其独特的模型和权重，但这些任务特定模型的参数之间的差异被纳入到联合目标函数中。这种方法确保了拟合函数考虑到了全球信息，促进了模型跨任务的总体泛化能力;

4.最后，本文提出了一个多阶段最小权匹配算法，该算法可以针对相邻子空间之间的样本快速得到效果较好的插值匹配策略,并根据求得的插值匹配策略对相邻的子空间之间的样本进行多次线性插值。通过特征子空间之间插值合成的样本具有相比原始样本更小的噪音误差，达到扩充数据、优化样本并且强化拟合效果的目的。RSIS 方法整体流程图如图 3.1 所示：

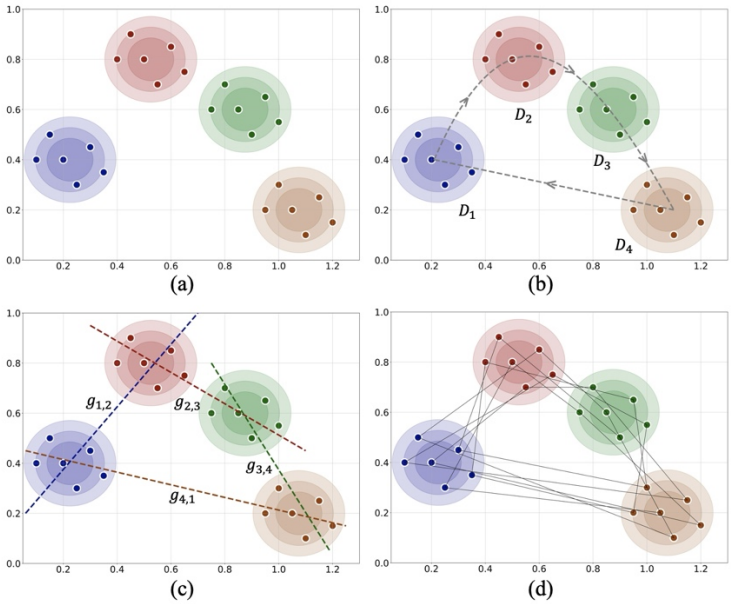

图 3.1 RSIS 方法整体流程图

图（a）中通过无监督聚类算法将原始特征空间划分为多个具有相近样本的特征子空间；图（b）中考虑了两维特征空间的情况对 TSP 问题进行求解，得到

子空间插值排序方案；图（c）对相邻排序的子空间之间进行线性拟合；图（d）根据线性拟合的结果，通过多阶段最小权匹配算法求得相邻子空间之间的样本插值匹配策略并进行插值，合成误差较小的样本。

3.1 特征子空间的划分

在 RSIS 方法中，首先需要对原始数据集进行聚类，本文提出了一种无监督聚类算法，该方法可以将特征空间划分为多个等量样本的子空间，该算法分为两个阶段，迭代聚类阶段以及 K 近邻优化阶段。

3.1.1 迭代层次聚类

迭代层次聚类过程的设计主要基于贪心策略的思想，通过多次循环迭代进行聚合聚类的无监督算法, 需要先给定超参数k，超参数k的直观解释是原始特征空间划分的子空间的数量。

首次迭代过程，给定样本特征的有序集合$C = \{c_1, c_2, \dots, c_n\} = \{\boldsymbol{x}_i\}_{i=1}^{n}$, 可得集合$C$中距离最近的两个元素$c_p, c_q$， $p, q = \underset{p,q}{\operatorname{argmin}}\, dist\left(c_p, c_q\right)$,并且定义集合$D_1 = c_p, c_q$, 作为第一个初始的簇，根据公式（3.1）计算簇心：

$$\overline{x^s} = \frac{\sum_{x^s \in D_s} x^s}{\mathrm{num}^{(D_s)}}, \tag{3.1}$$

其中，$\mathrm{num}(D_s)$为当前集合D_s中的样本数量。首次迭代的最后，将c_p, c_q从集合C中剔除，$C' = c_i \in C | i \neq p, q$, 并且更新集合$C' \leftarrow C' \cup \{\overline{\boldsymbol{x}}^1\}$.

在随后的迭代过程中，都需要对于集合C求得最小距离的两个元素c_p, c_q. 对于c_p, c_q分为三种情况进行讨论：c_p, c_q都是样本；c_p, c_q一个是样本另一个是簇心；c_p, c_q两个都是簇心。

在c_p, c_q都是原始样本的情况下，定义集合$D_s = c_p, c_q$生成新的簇并计算簇心$\overline{\boldsymbol{x}}^s$, 将$c_p, c_q$从集合$C$中剔除并且将$\overline{\boldsymbol{x}}^s$添加到集合$C$中，即$C' = \{c_i \in C | i \neq p, q\} \cup \{\overline{\boldsymbol{x}}^s\}$.

在c_p:$\boldsymbol{x}_p$是样本，c_q:$\overline{\boldsymbol{x}}^s$是簇心的情况下，更新$D_s \leftarrow D_s \cup \boldsymbol{x}_p$, 将$c_p$从集合$C$中剔除并且根据公式（3.1）重新计算并更新集合$C$中$D_s$的簇心，即$C' = \{c_i \in C | i \neq p, q\} \cup \overline{\boldsymbol{x}}^{s\prime}$. 对于$c_p$是簇心，$c_q$是样本的情况同理。

若c_p:$\overline{\boldsymbol{x}}^s$，$c_q$:$\overline{\boldsymbol{x}}^l$皆为簇心，则将两个簇进行合并$D_s \leftarrow D_s \cup D_l$，并且更新$C' = c_i \in C | i \neq p, q \cup \overline{\boldsymbol{x}}^{s\prime}$.

每次迭代结束后，如果$\mathrm{num}(D_s) \geq \lceil n/k \rceil$，则将$\overline{\boldsymbol{x}}^s$从$C$中剔除，并且引入 LOF 离群点检测[35]算法对集合D_s中的样本进行离群点检测，剔除掉多余的样本点$\{\boldsymbol{x}_i\}_{i=1}^{\mathrm{num}(D_s)-\lceil n/k \rceil}$，将$D_s = \{\boldsymbol{x}_i^s\}_{i=1}^{\lceil n/k \rceil}$作为一个完整的簇不参与后续迭代。一方面保证每个簇的样本数量为$\lceil n/k \rceil$，另一方面确保每个簇的样本在特征空间中不会过于分散以此进一步优化聚类的效果。最后将通过 LOF 算法检测出来的多余的样本点重新添加到集合C中，$C' = \{c_i \in C | c_i \neq \overline{\boldsymbol{x}}^s\} \cup \{\boldsymbol{x}_i\}_{i=1}^{\mathrm{num}(D_s)-\lceil n/k \rceil}$. 若集合$C = \emptyset$，则停止迭代。该部分具体的迭代流程详见图 3.2。

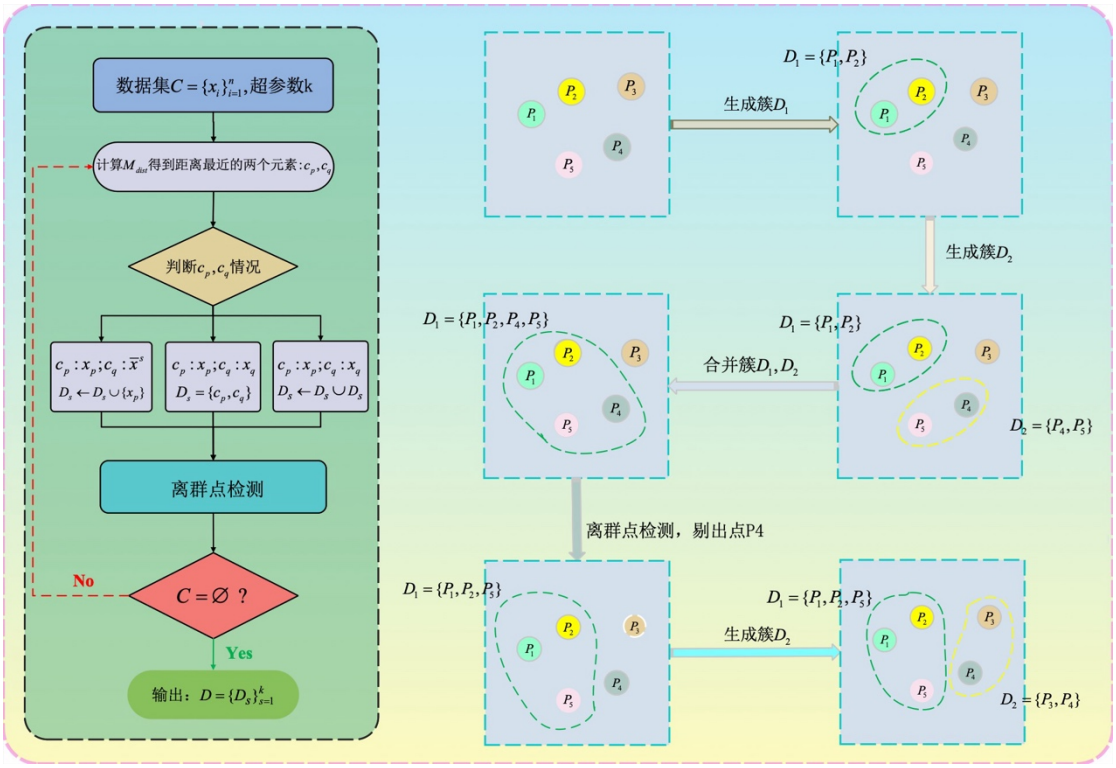

图 3.2 迭代层次聚类过程

综上，该聚类过程可以将原始数据集划分为$k-1$个具有等量样本和 1 个不

超过$\lceil n/k \rceil$个样本的子集，即$D = \bigcup_{s=1}^{k} D_s, D_s = \{\boldsymbol{x}_i^s\}_{i=1}^{l}, 1 \le l \le \lceil n/k \rceil$, 且将原始特征空间划分为$k$个子空间$\mathcal{X} = \bigcup_{s=1}^{k} \mathcal{X}^s$.

表 3.1 迭代层次聚类过程

算法 1: 迭代层次聚类
输入：特征数据集$C = \boldsymbol{x}_{i\,i=1}^{\,n}$；超参数$k$ 输出：$D = \{D_s\}_{s=1}^{k}$
While $C \neq \emptyset$： 得到特征空间中距离最近的两个元素c_p, c_q 对于c_p, c_q分为三种情况进行处理 若两个都是样本： 定义新的集合$D_s = c_p, c_q$ 计算簇心$\bar{\boldsymbol{x}}^s$，并将其添加到集合C中 将样本从集合C中剔除 若一个是样本另一个是簇心$\bar{\boldsymbol{x}}^s$： 将样本添加到集合D_s中 重新计算并且更新集合C中的簇心$\bar{\boldsymbol{x}}^s$ 将样本从集合C中剔除 若两个都是簇心： 合并集合$D_s \leftarrow D_s \cup D_l$ 重新计算并且更新集合C中的簇心$\bar{\boldsymbol{x}}^s$ 将簇心$\bar{\boldsymbol{x}}^l$从集合C中剔除 如果$\text{num}(D_s) \ge \lceil n/k \rceil$： 通过 LOF 算法筛选集合$D_s$中的多余样本 将多余的样本添加的集合$C$中 将簇心$\bar{\boldsymbol{x}}^s$从集合$C$中剔除 $D \leftarrow D \cup D_s$

在迭代层次聚类过程中，需要找到距离最近的两个元素，这会大大提升算法的复杂度，这一环节可以使用 K-d 树（K-dimensional tree）方法进行优化。构建 K-d 树的时间复杂度通常是$O(n \log n)$，在 K-d 树中删除或添加一个点以及查找最近两个元素的平均时间复杂度为$O(\log n)$，但随着迭代的进行，样本量会逐渐减少，因此实际的查找时间会略有下降。在不考虑 LOF 离群点检测将簇中多

余的样本重新放回数据集的情况下，迭代总次数为n次，因此总体时间复杂度应当会接近$O(n\log n)$。

3.1.2 K 近邻优化

K 近邻（K-Nearest Neighbors, KNN）方法是一种基于实例的学习算法，它不显式地进行模型学习，而是直接使用训练数据进行预测。KNN 方法在特征空间中对数据点进行分类或回归，其核心思想是在特征空间中找到与新样本最接近的 K 个已知数据点（即“邻居”），并根据这些邻居的信息来预测新样本的输出。

由于迭代层次聚类的过程是基于贪心策略的思想，这就会导致在迭代后期的聚类的可能存在不合理的情况，影响整体的聚类效果，因此本文设计了 K 近邻优化过程对聚类结果进行微调，一些样本的簇可能会改变，但整体上保持了簇的均匀性，以提高聚类效果。

初始聚类阶段完成后，RSIS 的层次聚类阶段为每个样本分配一个标签，表示其所属的簇。在 KNN 优化阶段，算法对每个样本计算其在特征空间中的$\lceil n/k \rceil$个最近邻样本。每个样本的簇标签随后更新为这些近邻中最常见的簇标签。值得注意的是，此过程可能会导致部分样本的簇标签发生变化，从而在簇内样本数量上产生轻微的波动。这种变化有可能导致样本分布较为分散的簇被吸收至其他簇中，进而消失。然而，此优化过程的设计确保了大多数簇的样本数量保持近似相等。通过结合迭代的层次聚类过程和 KNN 优化，我们得到了最终的聚类结果$\{D_s\}_{s=1}^{k'}$，其中，k'是 K 近邻优化后簇的数量。该算法的流程如图 3.3 所示，能有效地平衡簇内的样本数量，并优化簇的质量。

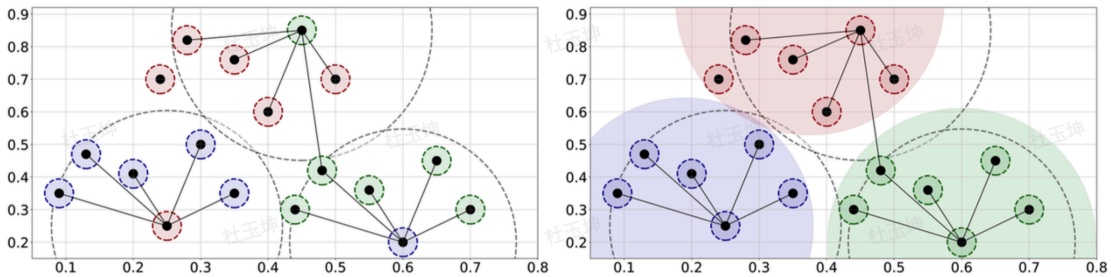

图 3.3 近邻优化流程图

总的来说，K 近邻优化是一个用于改进迭代层次聚类结果的过程，它通过找

到每个样本的最近邻居并将样本重新分配到占比最多的簇，以实现聚类结果的微调，从而在保证了簇的均匀性的同时提高了聚类效果。

3.2 子空间插值路径排序

通过对原始数据进行聚类可以得到多个子集，每个子集对应不同的特征子空间。RSIS 方法是对两个相邻的子空间之间的样本进行插值合成数据，该部分主要针对多个子空间插值路径的设计进行分析。

插值路径的设计思想是从一个子空间出发，经过一次所有的子空间并最终返回起始点，目标是最小化总路径距离。可以转换为旅行商问题（Travelling Salesman Problem, TSP）进行求解，将该问题表示为一个加权图，其中每个子空间样本的簇心可以表示为顶点，簇心之间的直线路径可以表示为加权图的边，路径的距离可以表示为边的权重。设定目标函数：

$$\min \sum_{i=1}^{k'-1} dist\left(\bar{\boldsymbol{x}}^{i}, \bar{\boldsymbol{x}}^{i+1}\right) + dist\left(\bar{\boldsymbol{x}}^{k'}, \bar{\boldsymbol{x}}^{1}\right). \tag{3.2}$$

旅行商问题是组合优化和理论计算机科学中的一个经典问题，也是一个典型的 NP-hard 问题，这意味着没有已知的多项式时间复杂度算法可以解决所有 TSP 实例，随着特征图顶点数量的增加，求解问题所需的时间和资源以指数级增长。在实际应用中，TSP 被用来解决物流、规划、芯片制造等领域的问题[36–39]，但是直接解决 TSP 是非常复杂的，许多研究设计了多种启发式算法和近似算法来找到足够好的解[40–42]，虽然这些解不一定是最优的，这些方法通过不同的策略在可接受的时间内寻找到一个近似最短路径。为了保证能在可接受的时间内得到足够好的解决方案，本文首先使用贪婪算法快速找到一个初始解，然后用 3-opt 方法来优化并得到最终解。最终可以得到一个排序好的集合$D_{sorted} = \left\{D_{(s)}\right\}_{s=1}^{k'}$. 在本文的后续部分，本文将对于序号相邻的两个子集所对应的特征子空间定义为相邻特征子空间。

以上是针对多维度特征空间的情况，若特征空间是一维的，考虑到一维空间的特殊性，插值路径的设计依旧要求经过一次所有的子空间，但不需要重新回到起点，可以直接使用贪心策略进行求解，在一维情况下，这种方法得到的

解一定是最优解。首先，找到插值路径的起点子空间：

$$D_{(1)} = \underset{D_s \in D}{\operatorname{argmin}}\, dist(\bar{x}^s, x_0), \tag{3.3}$$

其中，x_0是特征集合的最小值。对于后续簇的定义如下：

$$D_{(d)} = \underset{\{D_s \in D,\, D_s \neq D_1, \ldots, D_{d-1}\}}{\operatorname{argmin}} dist(\bar{x}^s, \bar{x}^{s-1}). \tag{3.4}$$

除非另有特别说明，本文后续的研究主要集中在多维特征空间情况下进行探讨。

3.3 相邻子空间线性回归拟合

给定含噪数据集$D = \boldsymbol{x}_i, y_i{}_{i=1}^{n}$, 其中$\boldsymbol{x}_i \in \mathcal{X} = \mathbb{R}^d, y_i \in \mathcal{Y} = \mathbb{R}$. 假设$\dot{y}_i = f(\dot{\boldsymbol{x}}_i)$, 其中$\dot{\boldsymbol{x}}_i$为$\boldsymbol{x}_i$的去噪真实值, $\dot{y}_i$为y_i的真实值, RSIS 方法假设关系式$f(\cdot)$为一个连续的函数, 即对于任意的$\boldsymbol{c} \in \mathcal{X}$, 任意的$\varepsilon > 0$, 存在$\delta > 0$使得, 当$\boldsymbol{x} \in \mathcal{X}$且$|\boldsymbol{x} - \boldsymbol{c}| < \delta$时, 总有$|f(\boldsymbol{x}) - f(\boldsymbol{c})| < \varepsilon$.可得：

$$\dot{y}_i + \varepsilon_{i,y} = f\left(\dot{\boldsymbol{x}}_i + \boldsymbol{\varepsilon}_{i,x}\right) + \varepsilon_i, \tag{3.5}$$

其中，$\boldsymbol{\varepsilon}_{i,x}$为$\dot{\boldsymbol{x}}_i$中所含的噪音，$\varepsilon_{i,y}$为$\dot{y}_i$中所含的噪音，$\varepsilon_i$为等式误差项。公式（3.5）可转化为：

$$y_i = f(\boldsymbol{x}_i) + \varepsilon_i. \tag{3.6}$$

通过上述的聚类以及子空间排序算法，可以将数据集D划分为多个有序子集$D_{sorted} = \left\{D_{(s)}\right\}_{s=1}^{k'}$, $D_{(s)} = \{\boldsymbol{x}_i^s, y_i^s\}_{i=1}^{num(D_{(s)})}$, 以及将特征空间划分为多个子空间$\mathcal{X} = \bigcup_{s=1}^{k'} \mathcal{X}_s$. 对于两个相邻的特征子空间，由于 RSIS 方法假设关系式$f(\cdot)$为一个连续的函数,则可以将$f(\cdot)$近似拟合为一个线性的函数$g(\cdot)$，公式（3.3）可转化为：

$$y_i^{s,s+1} = g_{s,s+1}\left(\boldsymbol{x}_i^{s,s+1}\right) + \varepsilon_i + \varepsilon_i', \tag{3.7}$$

其中，$g_{s,s+1}(\cdot)$与,ε_i'分别为相邻特征子空间$\mathcal{X}^{(s)}$与$\mathcal{X}^{(s+1)}$中线性拟合函数以及的

线性拟合误差，$\{\boldsymbol{x}_i^{s,s+1}, y_i^{s,s+1}\} \in D_{(s)} \cup D_{(s+1)}$. 若两个子空间之间的距离足够近并且子空间的测度趋向于 0 时，则线性拟合误差$\varepsilon_i' \to 0$.

可以考虑使用 OLS、Lasso 回归或者局部加权线性回归等方法[27,28]对两个相邻子空间之间的$g(\cdot)$进行估计，但是考虑到 RSIS 的假设$f(\cdot)$为一个连续的函数，仅仅根据两个子空间的样本直接进行线性拟合并不能充分考虑到全局的信息。本文借鉴了软参数多任务学习框架的思想[29–32]，同时对多个子空间之间的线性函数进行估计，并且在损失函数的设计添加了 L1 正则项，以此使得每个线性函数$\hat{g}_{s,s+1}(\cdot)$的估计都考虑了全局性信息：

$$L(\beta) = \sum_{s=1}^{k'} \left\| \boldsymbol{y}^{s,s+1} - X^{s,s+1}\beta^{s,s+1} \right\|_2^2 + \sum_{s=1}^{k'} \left[\frac{\lambda}{\operatorname{dist}(\bar{x}^s, \bar{x}^{s+1}) + 1} \left\| \beta^{s,s+1} - \beta^{s-1,s} \right\|_1 \right] \quad (3.8)$$

其中，$\beta = \left(\beta^{1,2}, \beta^{2,3}, \ldots, \beta^{k',k'+1}\right)$，$\beta^{s,s+1}$为$\hat{g}_{s,s+1}(\cdot)$参数，$\hat{g}_{k',k'+1}(\cdot)$与$\hat{g}_{0,1}(\cdot)$都是子空间$\mathcal{X}^{(k')}$与$\mathcal{X}^{(1)}$的线性拟合函数,即$\beta^{0,1} = \beta^{k',k'+1}$, $\frac{\lambda}{dist(\bar{\boldsymbol{x}}^s, \bar{\boldsymbol{x}}^{s+1})+1}$为正则项系数，根据子空间之间的距离自适应调整，在后文的模拟实验以及实例分析中定义λ=1.

Lemma 1. 对于函数$f(\beta) = \|y - X\beta\|_2^2$，其中$y \in \mathbb{R}^{n\times 1}, X \in \mathbb{R}^{n\times d}, \beta \in \mathbb{R}^{d\times 1}$. 对于$\forall\omega \in [0,1]$, 以及$\forall\beta_1, \beta_2 \in \mathbb{R}^{d\times 1}$,可得:

$$f(\beta_1 + (1-\omega)\beta_2) \le \omega f(\beta_1) + (1-\omega) f(\beta_2).$$

Lemma 2. 对于函数$f(\beta) = \|\beta^1 - \beta^2\|_1$，其中$\beta^1, \beta^2 \in \mathbb{R}^{d\times 1}, \beta = (\beta^1, \beta^2)$，对于$\forall\beta_1, \beta_2$以及$\forall\omega \in [0,1]$，可得：

$$f(\beta_1 + (1-\omega)\beta_2) \le \omega f(\beta_1) + (1-\omega) f(\beta_2).$$

Theorem 1. 根据公式（3.7）, 其中$\beta = \left(\beta^{1,2}, \ldots, \beta^{k',k'+1}\right) \in \mathbb{R}^{d\times k'}, \beta^{0,1} = \beta^{k',k'+1}$. 对于$\forall\omega \in [0,1]$以及$\forall\beta_1, \beta_2 \in \mathbb{R}^{d\times k'}$，可得$L(\beta_1 + (1-\omega)\beta_2) \le \omega L(\beta_1) + (1-\omega)L(\beta_2)$.

Proof of Theorem 1.

$$L(\omega\beta_1+(1-\omega)\beta_2)=\sum_{s=1}^{k'}\Bigg(\|y^{s,s+1}-X^{s,s+1}[\omega\beta_1^{s,s+1}+(1-\omega)\beta_2^{s,s+1}]\|_2^2 + \frac{\lambda}{\mathrm{dist}(\bar{x}^s,\bar{x}^{s+1})+1}\Big\|[\omega\beta_1^{s,s+1}+(1-\omega)\beta_2^{s,s+1}] - [\omega\beta_1^{s-1,s}+(1-\omega)\beta_2^{s-1,s}]\Big\|_1\Bigg)$$

根据 **Lemma 1** 以及 **Lemma 2** 可得：

$$\begin{aligned}L(\omega\beta_1+(1-\omega)\beta_2) &\le \sum_{s=1}^{k'}\Bigg(\omega\|y^{s,s+1}-X^{s,s+1}\beta_1^{s,s+1}\|_2^2+(1-\omega)\|y^{s,s+1}-X^{s,s+1}\beta_2^{s,s+1}\|_2^2 \\ &\quad+\omega\frac{\lambda}{\mathrm{dist}(\bar{x}^s,\bar{x}^{s+1})+1}\|\beta_1^{s,s+1}-\beta_1^{s-1,s}\|_1 \\ &\quad+(1-\omega)\frac{\lambda}{\mathrm{dist}(\bar{x}^s,\bar{x}^{s+1})+1}\|\beta_2^{s,s+1}-\beta_2^{s-1,s}\|_1\Bigg) \\ &=\omega\sum_{s=1}^{k'}\Bigg(\|y^{s,s+1}-X^{s,s+1}\beta_1^{s,s+1}\|_2^2+\frac{\lambda}{\mathrm{dist}(\bar{x}^s,\bar{x}^{s+1})+1}\|\beta_1^{s,s+1}-\beta_1^{s-1,s}\|_1\Bigg) \\ &\quad+(1-\omega)\sum_{s=1}^{k'}\Bigg(\|y^{s,s+1}-X^{s,s+1}\beta_2^{s,s+1}\|_2^2+\frac{\lambda}{\mathrm{dist}(\bar{x}^s,\bar{x}^{s+1})+1}\|\beta_2^{s,s+1}-\beta_2^{s-1,s}\|_1\Bigg) \\ &=\omega L(\beta_1)+(1-\omega)L(\beta_2)\end{aligned}$$

根据 **Theorem 1.**可知，$Loss(\beta)$是一个凸函数，因此结果必然收敛且局部最优解就是全局最优解。

3.4 多阶段最小权匹配

接下来需要对相邻的两个子空间的样本进行线性插值合成新的样本，插值的过程是在两个样本之中进行的，RSIS 方法生成的数据要充分考虑到所有的样本信息，插值规则设计如下：

1. 插值的两个样本应当来自不同的子空间；

2. 每个样本至少进行一次插值；

3. 插值次数为$max\big(num(D_s),\mathrm{num}(D_{s+1})\big)$；

4. 所有样本参与插值的次数必须是均匀的，即每个样本最多插值$\left\lceil\frac{\max(\mathrm{num}(D_s),\mathrm{num}(D_{s+1}))}{\min(\mathrm{num}(D_s),\mathrm{num}(D_{s+1}))}\right\rceil$次。

由于数据集D包含噪音，通过插值合成的样本会受到原始样本噪音的影响也会包含噪音。在对两个相邻子空间进行线性插值时，插值匹配策略的选择需要

考虑到合成的样本相比真实分布的误差。不失一般性，假设$\mathrm{num}\left(D_{(s)}\right)=m$, $\mathrm{num}\left(D_{(s+1)}\right)=n,\ n/2\le m\le n$, 根据 RSIS 方法的插值规则，一共存在$C_m^{n-m}\cdot C_n^2\cdot C_{n-2}^2\ldots C_{n-2(n-m-1)}^2\cdot(n-m)!$种插值匹配策略可供选择。同样不失一般性，不妨假设特征空间的维度$d=1$，可以通过公式（3.9）匹配策略的误差进行度量：

$$\sum S(x^s,x^{s+1})=\sum\int_{x^s}^{x^{s+1}}|f(x)-L(x)|dx, \tag{3.9}$$

其中，$L(x)$是经过$(\mathrm{x^s},\mathrm{y^s}),(x^{s+1},y^{s+1})$两点的线性函数。进一步的，通过公式（3.9）可以求得合成样本的平均误差，见公式（3.10）.

$$\bar{S}(x^s,x^{s+1})=\frac{\int_{x^s}^{x^{s+1}}|f(x)-L(x)|dx}{|x^{s+1}-x^s|}, \tag{3.10}$$

如果用穷举法计算所有策略的平均误差，由于公式（3.9），（3.10）涉及到复杂的积分运算，并且如果子空间样本数量过多，这会消耗大量的计算资源，若特征空间的维度较高，则需要的计算资源会更大。因此，如何仅仅耗费少量的计算资源的情况下快速且高效选择一种较优的插值匹配策略是需要 RSIS 方法重点探讨的问题。

Theorem 2. 对于两个相邻的子空间$\mathcal{X}^s$与$\mathcal{X}^{s+1}$对应的子集D_s,D_{s+1}. $(x^s,y^s)\in D_s$, $x^{s+1},y^{s+1}\in D_{s+1}$. 考虑关系式$y_i=f(x_i)+\varepsilon_i$。假设线性拟合误差$\varepsilon'\to 0$，则有:

$$\mathbb{E}\left(\frac{S(x^s,x^{s+1})}{|x^s-x^{s+1}|}\right)<\mathbb{E}\left(\frac{(|\varepsilon^s|+|\varepsilon^{s+1}|)}{2}\right).$$

Proof of Theorem 2. 由于假设线性拟合误差$\varepsilon'\to 0$，根据公式（3.7）可得：

$$y^{s,s+1}=g_{s,s+1}(x^{s,s+1})+\varepsilon.$$

由于$g_s(\cdot)$为线性函数，根据公式（3.9）以及（3.10）可得：

$$S(x^s,x^{s+1})=\int_{x^s}^{x^{s+1}}|g(x)-L(x)|dx.$$

根据迭代期望定律(LIE)可得:

$$\mathbb{E}\left(\frac{S(x^s,x^{s+1})}{|x^{s+1}-x^s|}\right)=\frac{\mathbb{E}(S(x^s,x^{s+1})|\varepsilon^s\cdot\varepsilon^{s+1}<0)P(\varepsilon^s\cdot\varepsilon^{s+1}<0)}{|x^{s+1}-x^s|}$$

$$+\frac{\mathbb{E}(S(x^s,x^{s+1})|\varepsilon^s\cdot\varepsilon^{s+1}\geq 0)P(\varepsilon^s\cdot\varepsilon^{s+1}\geq 0)}{|x^{s+1}-x^s|}.$$

可根据基础的几何面积计算来简化$S(x^s,x^{s+1})$，当$\varepsilon^s\cdot\varepsilon^{s+1}<0$时，$\exists x'\in(x^s,x^{s+1})$, 使得$g(x')=L(x')$. 可得：

$$\mathbb{E}(S(x^s,x^{s+1})|\varepsilon^s\cdot\varepsilon^{s+1}<0)=\frac{\mathbb{E}(|\varepsilon^s|)\cdot(x'-x^s)+\mathbb{E}(|\varepsilon^{s+1}|)\cdot(x^{s+1}-x')}{2},$$

当$\varepsilon^s\cdot\varepsilon^{s+1}\geq 0$时：

$$\mathbb{E}(S(x^s,x^{s+1})|\varepsilon^s\cdot\varepsilon^{s+1}\geq 0)=\frac{\mathbb{E}(|\varepsilon^s|+|\varepsilon^{s+1}|)\cdot(x^{s+1}-x^s)}{2}.$$

代入原式可得：

$$\begin{aligned}\mathbb{E}\left[\frac{S(x^s,x^{(s+1)})}{|x^{(s+1)}-x^s|}\right]&=\left(\frac{\mathbb{E}[|\varepsilon^s|]\cdot(x'-x^s)+\mathbb{E}[|\varepsilon^{(s+1)}|]\cdot(x^{(s+1)}-x')}{2(x^{(s+1)}-x^s)}\right)\cdot P(\varepsilon^s\cdot\varepsilon^{(s+1)}<0)\\&+\left(\frac{\mathbb{E}[|\varepsilon^s|+|\varepsilon^{(s+1)}|]\cdot(x^{(s+1)}-x^s)}{2(x^{(s+1)}-x^s)}\right)\cdot P(\varepsilon^s\cdot\varepsilon^{(s+1)}\geq 0)\\&=\left(\frac{\mathbb{E}[|\varepsilon^s|]\cdot(x'-x^s)+\mathbb{E}[|\varepsilon^{(s+1)}|]\cdot(x^{(s+1)}-x')}{2(x^{(s+1)}-x^s)}\right)\cdot P(\varepsilon^s\cdot\varepsilon^{(s+1)}<0)\\&+\left(\frac{\mathbb{E}[|\varepsilon^s|+|\varepsilon^{(s+1)}|]}{2}\right)\cdot P(\varepsilon^s\cdot\varepsilon^{(s+1)}\geq 0).\end{aligned}$$

由于$P(\varepsilon^s\cdot\varepsilon^{s+1}<0)+P(\varepsilon^s\cdot\varepsilon^{s+1}\geq 0)=1$, 并且：

$$\frac{\mathbb{E}(|\varepsilon^s|)\cdot(x'-x^s)+\mathbb{E}(|\varepsilon^{s+1}|)\cdot(x^{s+1}-x')}{2(x^{s+1}-x^s)}$$

$$=\frac{\mathbb{E}(|\varepsilon^s|)\cdot\frac{x'-x^s}{x^{s+1}-x^s}+\mathbb{E}(|\varepsilon^{s+1}|)\cdot\frac{x^{s+1}-x'}{x^{s+1}-x^s}}{2}<\frac{\mathbb{E}(|\varepsilon^s|)+\mathbb{E}(|\varepsilon^{s+1}|)}{2},$$

可得$\mathbb{E}\left(\frac{S(x^s,x^{s+1})}{|x^s-x^{s+1}|}\right)<\mathbb{E}\left(\frac{(|\varepsilon^s|+|\varepsilon^{s+1}|)}{2}\right)$.

通过 **Theorem 2.**可知，在假设线性拟合误差$\varepsilon'\to 0$的条件下，即使面对噪音不同分布的情况，对任意两个子空间的样本进行线性插值，则插值合成数据的

平均噪音期望小于这两个样本的平均噪音期望。

Theorem 3. 令$y^s = f(x^s) + \varepsilon^s,\ y^{s+1} = f(x^{s+1}) + \varepsilon^{s+1}$. 假设线性拟合误差$\varepsilon' \to 0$，则：

$$\frac{S(x^s, x^{s+1})}{|x^s - x^{s+1}|} = \frac{|\varepsilon^s + \varepsilon^{s+1}|}{2}$$

Proof of Theorem 3. 当$\varepsilon^s \cdot \varepsilon^{s+1} \geq 0$时，由于线性拟合误差$\varepsilon' \to 0$,可得：

$$\mathbb{E}\left(\frac{S(x^s, x^{s+1})}{|x^s - x^{s+1}|}\middle|\varepsilon^s \cdot \varepsilon^{s+1} \geq 0\right) = \frac{\mathbb{E}(|\varepsilon^s| + |\varepsilon^{s+1}|) \cdot |x^{s+1} - x^s|}{2|x^s - x^{s+1}|} = \frac{\mathbb{E}(|\varepsilon^s| + |\varepsilon^{s+1}|)}{2},$$

$$\mathbb{E}\left(\frac{S(x^s, x^{s+1})}{|x^s - x^{s+1}|}\middle|\varepsilon^s \cdot \varepsilon^{s+1} < 0\right) = \frac{\mathbb{E}(|\varepsilon^s|) \cdot (x' - x^s) + \mathbb{E}(|\varepsilon^{s+1}|) \cdot (x^{s+1} - x')}{2|x^s - x^{s+1}|}$$

$$= \frac{\mathbb{E}(|\varepsilon^s + \varepsilon^{s+1}|)}{2} < \frac{\mathbb{E}(|\varepsilon^s| + |\varepsilon^{s+1}|)}{2}$$

易得：

$$\mathbb{E}\left(\frac{S(x^s,x^{s+1})}{|x^s-x^{s+1}|}\middle|\varepsilon^s \cdot \varepsilon^{s+1} < 0\right) = \mathbb{E}\left(\frac{(|\varepsilon^s|+|\varepsilon^{s+1}|)}{2}\right) < \mathbb{E}\left(\frac{S(x^s,x^{s+1})}{|x^s-x^{s+1}|}\middle|\varepsilon^s \cdot \varepsilon^{s+1} \geq 0\right) = \mathbb{E}\left(\frac{(|\varepsilon^s|+|\varepsilon^{s+1}|)}{2}\right).$$

通过以上证明不难看出，在满足假设条件的情况下，不论两个样本之间插值数量多少，总有：

$$\frac{S(x^s,x^{s+1})}{|x^s-x^{s+1}|} = \frac{|\varepsilon^s+\varepsilon^{s+1}|}{2} \leq \frac{|\varepsilon^s|+|\varepsilon^{s+1}|}{2}.$$

可以根据子空间之间线性拟合函数$\hat{g}_s(\cdot)$估计样本的噪音，通过公式（3.7）可得：

$$\varepsilon_i^{s,s+1} = y_i^{s,s+1} - \hat{g}_{s,s+1}\left(\boldsymbol{x}_i^{s,s+1}\right). \tag{3.11}$$

求得子空间之间的样本误差，下一步只需要考虑如何根据样本的误差信息去确定样本之间的匹配策略，因为不同误差的样本之间进行插值合成的数据噪音也会不同，可以转换为二分图最小权匹配问题进行分析。近些年来，最小权匹配问题在物流、网络设计、资源分配等领域有广泛应用[49–51]；也有部分研究将最小权匹配问题扩展到更复杂的网络中，例如动态网络或随机网络，研究如何在变化的环境中找到最优解[52–54]；对于那些计算上不可行求解精确结果的大

型问题，一些研究可能会寻找能够快速得到近似解的算法[55,56]。

根据插值规则，需要对每个样本至少进行一次插值，并且约束了插值的总次数。经过 K 近邻优化后子空间之间的样本数量大多数情况并不是完全相等的，所以这不能视作一个最小权完美匹配问题。这个问题是一个加权匹配问题的变体，同时融合了平衡匹配和最小权匹配的特点，因为它需要最小化匹配的总权重，并且约束了样本匹配次数的均匀性。目前针对这种问题并没有一个现成的标准算法，因此需要对传统的最小权匹配问题进行改进以适应此目标场景。本文提出了一种启发式的多阶段的最小权匹配方法对该问题进行分析。需要注意的是，这种方法并不能保重所得结果是一个最优解，但是但它可以在极短的时间内得到一个合理的近似解。这种方法的优点在于它相对简单，并且对于大规模问题来说可以提供较好的效率，如果需要找到最优解，可能需要使用更复杂的优化方法，例如整数线性规划（ILP）来精确求解问题，但通常需要更多的计算资源和时间。

依旧延续上文的假设$\mathrm{num}(D_s) = m,\ \mathrm{num}(D_{s+1}) = n,\ n/2 \le m \le n$。对于一个加权二分图$G = (D_s, D_{s+1}, E)$，和一个权重函数$w: E \to \mathbb{R}$, 其中$D_s, D_{s+1}$是两侧顶点的集合，$E \subseteq D_s \times D_{s+1}$是边的集合，表示可能的匹配。由于$\frac{S(x^s, x^{s+1})}{|x^s - x^{s+1}|} = \frac{(|\varepsilon^s + \varepsilon^{s+1}|)}{2}$，定义权重函数$w(\boldsymbol{x}^s, \boldsymbol{x}^{s+1}) = |\varepsilon^s + \varepsilon^{s+1}|/2$. 考虑目标函数：

$$min \sum_{(i,j)\in E} w\left(\boldsymbol{x}_i^s, \boldsymbol{x}_j^{s+1}\right) x_{ij}, \qquad (3.12)$$

$$\text{s.t. } \sum_{(i,j)\in E} x_{ij} = n,$$

$$\forall j, \sum_{i=1}^{n} x_{ij} \in 1,2,$$

其中$x_{ij} \in (0,1)$为二元决策变量。

对于集合D_s的每个样本考虑进行一次或者两次匹配，因此分为两个阶段进行求解。在第一个阶段中，在不考虑目标函数（3.12）约束条件的情况下，使用匈牙利算法直接对二分图G进行求解，最终可得二分图G的一个最大匹配M_1. 此时对于每个$x^s \in D_s$都参与了一次匹配，D_{s+1}中m个顶点参与了一次匹配，还有$(m-n)$个顶点并未参与匹配，对于未参与匹配的顶点定义为D'_{s+1}.

在第二个阶段中，定义一个新的加权二分图$G' = (D_s, D'_{s+1}, E')$，$E' \subseteq D_s \times D'_{s+1}$，依旧使用匈牙利算法对二分图$G'$求解，并得到最大匹配$M_2$. 由于 $n/2 \leq m \leq n$，因此在M_2中对于D'_{s+1}每个顶点都参与了一次匹配，D_s中$(m-n)$个顶点再次参与了一次匹配。最终将M_1与M_2合并得到$M = M_1 \cup M_2$。M是二分图G中的一个匹配，使得D_s中的每个顶点参与一次或两次匹配，且而D_{s+1}中的每个顶点仅参与一次匹配，满足约束要求。

以上是对特定的假设情况$n/2 \leq m \leq n$介绍的二阶段最小权匹配方法，这也是在子空间之间样本匹配中最常见的一种情况。若两个子空间样本数量差距过大，这个问题则需要分为$\left\lceil \frac{max(num(D_s), num(D_{s+1}))}{min(num(D_s), num(D_{s+1}))} \right\rceil$个最小权匹配阶段进行求解（见表 3.2）。

表 3.2 多阶段最小权匹配

算法 2: 多阶段最小权匹配
输入：顶点集$D_s = \{\boldsymbol{x}_i^s\}_{i=1}^m, D_{s+1} = \{\boldsymbol{x}_i^{s+1}\}_{i=1}^n (m \leq n)$; 线性回归函数$\hat{g}_{s,s+1}$. 输出：插值匹配$M$
估计样本的噪音 $\varepsilon_i^s = y_i^s — \hat{g}_s(\boldsymbol{x}_i^s)$ 定义二分图$G = (D_s, D_{s+1}, E)$ 定义权重函数$w(x^s, x^{s+1}) = (\varepsilon^s + \varepsilon^{s+1})/2$ For $t = 1,2, \ldots, \left\lceil \frac{n}{m} \right\rceil$: 使用匈牙利算法对二分图$G$求解得$M_t$ 更新$D'_{s+1} = D_{s+1} \backslash S$, 其中$S \subseteq D_{s+1}$为$M_t$中已匹配的$D_{s+1}$的顶点集 更新二分图$G' = (D_s, D'_{s+1}, E')$ $M \leftarrow M \cup M_t$

3.5 方法整体分析

由于子空间的数量为k'，因此需要进行k'次多阶段最小权匹配计算得到k'个匹配策略，最终可求得所有样本之间插值路径总和$dist_{sum}$. 需要给定另外一个超参数η，该超参数直观解释是合成样本数量与原始样本数量的比值。对于两个

来自相邻且不同子空间的样本$\{\boldsymbol{x}^s, y^s\}, \boldsymbol{x}^{s+1}, y^{s+1}$，合成的样本为$\boldsymbol{x}_{(d)}^{s,s+1}, y_{(d)}^{s,s+1}{}_{d=1}^{\left\lfloor \frac{\eta \cdot n}{dist_{sum}} \cdot dist(\boldsymbol{x}^s, \boldsymbol{x}^{s+1}) \right\rfloor}$. 其中$n$为原始样本量，$\frac{\eta \cdot n}{dist_{sum}}$的直观解释是单位距离插入的样本数量，$\left\lfloor \frac{\eta \cdot n}{d_{sum}} \cdot dist(\boldsymbol{x}^s, \boldsymbol{x}^{s+1}) \right\rfloor$是两个原始样本之间插入样本的总数量。通过这样的设计方式，可以有效控制合成样本的总数量。假设所有的变量都是连续的，线性插值公式设计如下：

$$\boldsymbol{x}_{(\mathrm{d})}^{s,s+1} = \boldsymbol{x}^s + d \cdot \frac{\boldsymbol{x}^s - \boldsymbol{x}^{s+1}}{\left\lfloor \frac{\eta \cdot n}{dist_{sum}} \cdot dist(\boldsymbol{x}^s, \boldsymbol{x}^{s+1}) \right\rfloor + 1} \tag{3.13}$$

$$y_{(\mathrm{d})}^{s,s+1} = y^s + d \cdot \frac{y^s - y^{s+1}}{\left\lfloor \frac{\eta \cdot n}{dist_{sum}} \cdot dist(\boldsymbol{x}^s, \boldsymbol{x}^{s+1}) \right\rfloor + 1} \tag{3.14}$$

通过这种方式可以自适应地根据样本在特征空间的距离去控制生成样本的数量，并且插入的样本都是线性且在两个样本中都是等距的。对相邻的子空间之间通过多阶段最小权匹配算法求得样本插值匹配策略，并求得距离总和$dist_{sum}$，根据公式（3.13），（3.14）进行插值合成新的具有更小噪音的样本，RSIS 方法整体完成。

综上所述，RSIS 方法共有三个假设：

1、$f(\cdot)$为一个连续的函数；

2、线性拟合误差$\varepsilon' \to 0$；

3、样本变量是连续的。

该方法分为多个步骤，整体操作流程以及特征空间一维情况下的直观效果图见表 3.3 以及图 3.5。

表 3.3 RSIS 方法

算法 3: RSIS 方法
输入：数据集 $D = \boldsymbol{x}_i, y_{i}{}_{i=1}^{n}$， 超参数$k, \eta$ 输出：插值合成后的数据集D'
构建特征数据集$C = \boldsymbol{x}_{i}{}_{i=1}^{n}$ 通过聚类算法对样本进行聚类得到$D = \{D_s\}_{s=1}^{k'}$ 构建加权图转换使用贪心算法得到一个初始插值路径解

使用 3-opt 方法来优化并得到最终解

对子空间之间进行线性拟合得到$\hat{g}_{s}{}_{s=1}^{k'}$

For $t = 1,2,3,\ldots,k'$:

估计D_s, D_{s+1}中样本的噪音$\varepsilon_i^s = y_i^s - \hat{g}_s(\boldsymbol{x}_i^s)$

利用多阶段最小权匹配方法得到插值匹配策略

根据匹配策略对样本进行插值合成新的数据

将新合成的数据添加到数据集D中

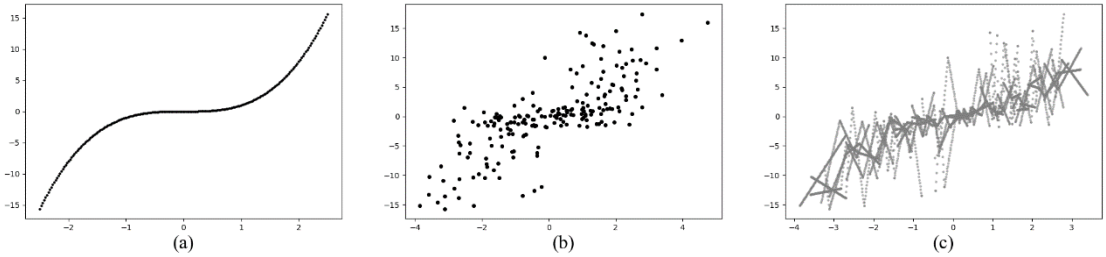

图 3.5 一维特征空间下 RSIS 方法处理效果图

如上图所示，生成实际函数关系为$y = x^3$的样本，样本量为 200. 随后对样本添加高斯噪声。令k =6，η =19，对噪音数据使用 RSIS 方法进行数据合成处理，样本量增加至 3808。

四、RSIS 优化效果实验

本文设计了六组模拟数据集以及六组实例数据，通过模拟数据探究 RSIS 方法对样本噪音的优化效果以及超参数分析，通过实例数据探究 RSIS 方法结合机器学习模型对模型泛化能力的提升效果。

4.1 模拟实验

4.1.1 模拟实验设计

本文设计了平均绝对误差(Mean Absolute Error, MAE)、误差占比指标用以度量 RSIS 方法对于数据集噪音的优化效果：

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - f(\boldsymbol{x_i})| \tag{4.1}$$

$$P(\alpha) = \frac{1}{n}\sum_{i=1}^{n} I(|y_i - f(\boldsymbol{x}_i)| \geq \alpha) \tag{4.2}$$

考虑到 RSIS 方法假设各个变量以及函数关系都是连续的，因此本文使用正态分布生成关系矩阵$M_1 \in \mathbb{R}_{d\times d_1}$, $M_2 \in \mathbb{R}_{d_1\times 1}$. 考虑使用模型$y_i = f(\boldsymbol{x_i}) + \varepsilon_i = \tanh(\boldsymbol{x}_i M_1)\, M_2 + \varepsilon_i$.

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}. \tag{4.3}$$

对于给定的数据集，通常难以判断数据噪音的分布，并且噪音大多数情况下并不是同分布的，本文设计了三种不同分布的噪音按照一定比例添加到模拟数据集中，以模拟现实数据集的实际噪音情况。对于每组实验准备 25 组样本并取平均值作为实验结果。模拟数据集详细信息见表 4.1.

表 4.1 模拟数据集

模拟数据集	噪音分布	样本量	特征维度	特征分布
D_1	$20\% - N(0,64)$ $30\% - U(-8,8)$ $50\% - N(0,1)$	500	10	$N(0,10)$
D_2	20%-N (0,64) 30%-U (-8,8) 50%-N (0,1)	500	10	$T(10)$
D_3	20%-N (0,64)	1500	10	$N(0,10)$

	30%-U (-8,8) 50%-N (0,1) 20%-N (0,64)			
D_4	30%-U (-8,8) 50%-N (0,1) 50%-N (0,64)	500	30	$N(0,10)$
D_5	30%-U (-8,8) 20%-N (0,1) 20%-N (0,64)	500	10	$N(0,10)$
D_6	30%-U (-8,8) 50%-N (0,1)	200	10	$N(0,10)$

4.1.2 优化效果分析

表 4.2 中给出了每个模拟实验的原始数据集以及经过不同超参数下 RSIS 方法处理后数据集的平均绝对误差的计算结果，并且给出了对应的标准差（括号内）。并且还对于数据集噪音是否得到优化进行了 0.05 水平下的威尔科克森符号秩检验（Wilcoxon Signed-Rank Test）。正负号以及约等号分别表示经过 RSIS 方法处理后的数据集均匀绝对误差显著正向优化、显著负向优化、未显著优化。

这种检验用于比较两个相关样本、匹配样本或成对样本的差异，或者比较单个样本与理论中值的差异。它是对成对观测差异的中位数是否为零的检验。符号秩检验不需要数据服从正态分布，因此它对于非正态分布的数据尤其有用。这种检验经常用于医学、心理学和其他社会科学领域的研究，特别是当数据不能假设为正态分布或样本量较小时。

表 4.2 模拟数据集优化效果

数据集	RSIS 处理前	RSIS 处理后								
		k: 5	10	15	5	10	15	5	10	15
		η: 1	1	1	5	5	5	10	10	10
	17.79	13.82+	14.47+	15.48+	13.14+	14.42+	15.75+	13.01+	14.48+	15.89+
D_1	0.290	0.319+	0.315+	0.313+	0.341+	0.320+	0.315+	0.344+	0.319+	0.314+
	5.160	3.908+	3.947+	4.210+	3.919+	3.967+	4.207+	3.974+	4.317+	4.191+
	19.97	16.01+	16.34+	16.97+	15.11+	16.06+	16.71+	14.95+	16.05+	16.78+
D_2	0.241	0.249+	0.259+	0.259+	0.259+	0.271+	0.265+	0.260+	0.273+	0.265+
	6.179	4.484+	4.793+	5.095+	4.671+	4.813+	4.893+	4.603+	4.925+	4.913+
	18.37	13.77+	14.35+	14.95+	13.21+	14.34+	14.89+	13.12+	14.36+	14.99+
D_3	0.313	0.333+	0.339+	0.338+	0.345+	0.345+	0.339+	0.345+	0.344+	0.336+
	5.143	3.446+	3.466+	4.318+	3.187+	3.284+	3.880+	3.185+	3.225+	6.409+
	19.29	19.29≈	15.71+	15.07+	19.29≈	14.26+	14.15+	19.29≈	13.95+	14.01+
D_4	0.280	0.280≈	0.302+	0.312+	0.280≈	0.325+	0.340+	0.280≈	0.331+	0.345+
	6.643	6.643≈	4.973+	4.578+	6.643≈	4.303+	4.097+	6.643≈	4.112+	3.991+
	31.88	22.15+	22.34+	23.00+	21.17+	21.94+	22.88+	21.01+	21.90+	22.87+
D_5	0.190	0.211+	0.216+	0.220+	0.237+	0.239+	0.236+	0.240+	0.239+	0.237+
	9.508	6.221+	6.557+	6.607+	6.149+	6.641+	6.679+	6.102+	6.769+	6.714+
	18.64	15.12+	16.09+	16.89+	14.61+	14.04+	16.91+	14.53+	16.05+	16.99+
D_6	0.307	0.328+	0.329+	0.330+	0.333+	0.332+	0.332+	0.334+	0.332+	0.331+
	3.958	2.943+	3.347+	3.505+	2.841+	3.271+	3.596+	2.795+	3.302+	3.577+
+		(5,5,5)	(6,6,6)	(6,6,6)	(5,5,5)	(6,6,6)	(6,6,6)	(5,5,5)	(6,6,6)	(6,6,6)
-		(0,0,0)	(0,0,0)	(0,0,0)	(0,0,0)	(0,0,0)	(0,0,0)	(0,0,0)	(0,0,0)	(0,0,0)
≈		(1,1,1)	(0,0,0)	(0,0,0)	(1,1,1)	(0,0,0)	(0,0,0)	(1,1,1)	(0,0,0)	(0,0,0)

通过表 4.1 可知：

1. 六组模拟数据集的原始 MAE 在 11.234-33.696 的范围内,说明原始数据中存在不同程度的噪音。RSIS 方法在不同参数设置下,大多数情况下都能够显著减小平均绝对误差,最大降低 13.682. 这验证了 RSIS 方法的有效性。

2. 即使是噪音比例较高的样本，RSIS 方法也表现出良好的优化效果，具有良好的鲁棒性。

3. 即使面对厚尾分布或样本量较少的数据集，RSIS 方法依旧可以对未知噪

音进行显著的优化，说明 RSIS 方法同样适用于多种场景。

4.1.3 超参数分析

RSIS 方法包含两个超参数，从表 4.1 可以看出不同的超参数下 RSIS 方法对数据集的优化效果也不一样，甚至也会出现没有显著优化的情况。接下来重点探讨不同超参数取值对原始数据集优化效果的影响。

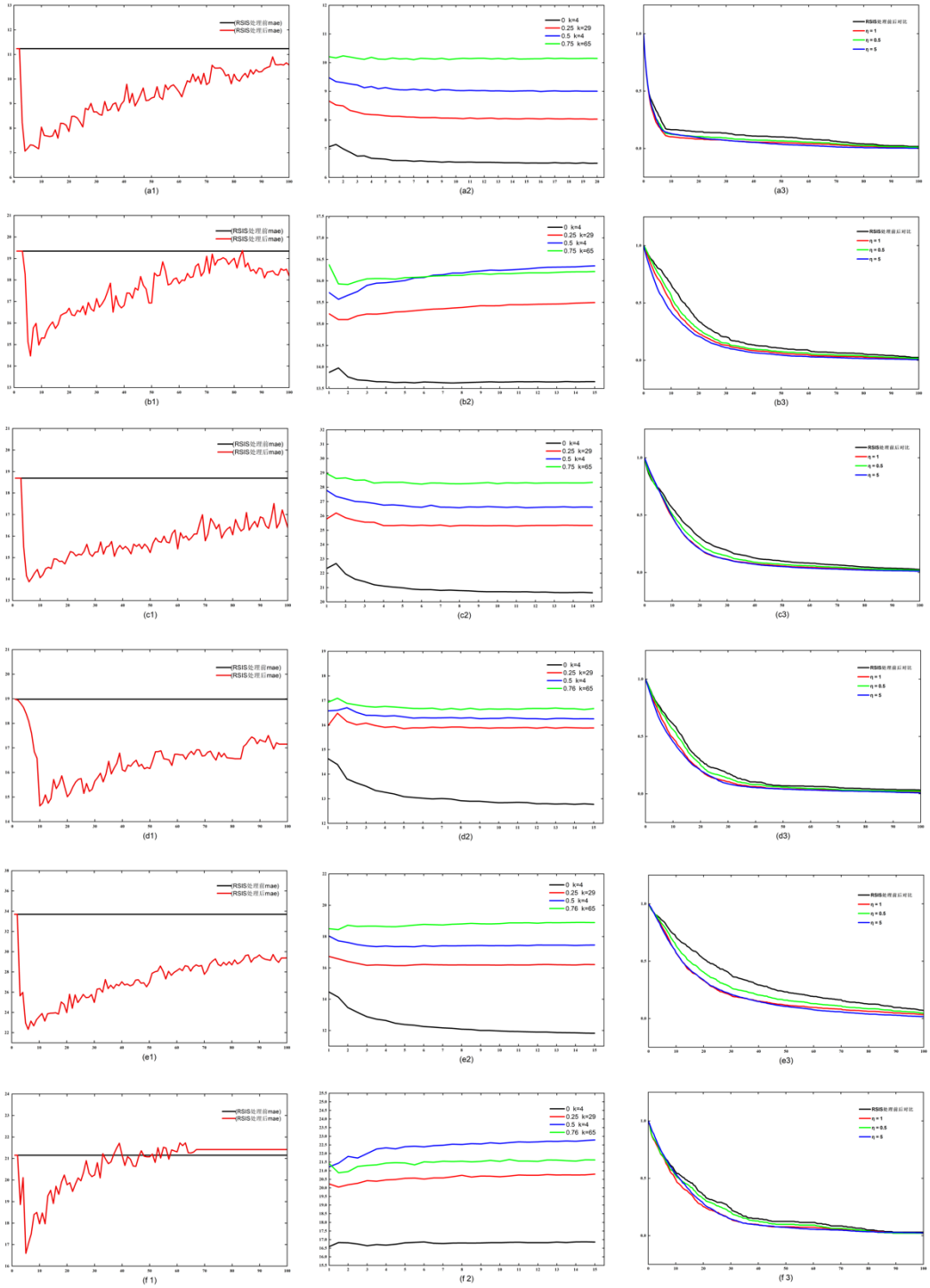

图 4.1 不同超参数下模拟数据集优化效果图

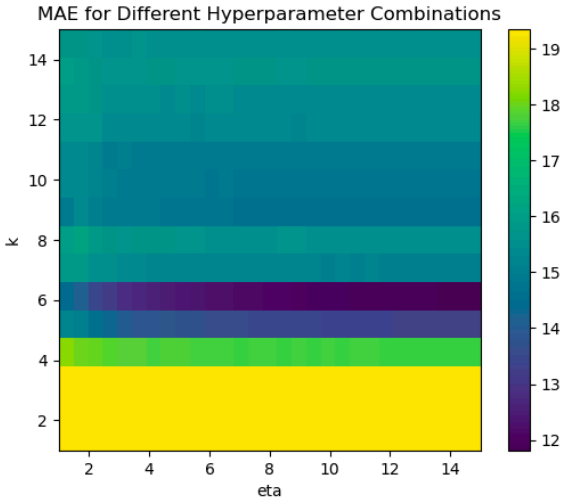

图 4.2 数据集D_1超参数热力图

在图 4.1 中，(a1),(b1),…,(f1)为六组模拟数据集在不同的超参数$k = 1,2,\ldots,100, \eta = 1$时样本的 MAE 变化情况；根据 MAE 的优化效果，得到多个分位数超参数k的取值，随后令$\eta = 1,1.5,\ldots,15$, 计算 MAE 的变化情况，具体可见图 (a2),(b2),…,(f2)；图(a3),(b3),…,(f3)为超参数k的取值为最优情况下，根据公式 4.1，探究不同η情况下不同程度的数据误差占总比的情况。在图 4.2 中，计算多种超参数组合下对模拟数据集D_1的优化效果，绘制热力图。

通过图 4.1 可以直观的看出：

1. 大多数情况下，样本的误差起初都会随着超参数k的增加而快速下降，随后波动上升；

2. 即使面对噪音更加复杂的情况，RSIS 方法依旧表现很好的优化效果，说明了 RSIS 方法具有不错的鲁棒性；

3. 在超参数k取值合适的情况下，超参数η的增加会使得优化效果进一步提升，部分情况下超参数η的增加不会显著影响优化效果，很少部分情况会恶化优化效果；

4. 超参数η的增加会明显使得误差较大的样本占比下降。

5. 相比于超参数η, 样本的优化效果对于超参数k的敏感性更强。

4.2 实例分析

使用 Bike Sharing Demand、Air Quality、三个实例数据[19-20]，按照 7:3 的比

例划分训练集与测试集，并进行极大极小归一化处理。对训练集数据使用 RSIS 方法进行数据合成，机器学习预测模型选择 K 近邻(K-Nearest Neighbor, KNN)，随机森林(Random Forest, RF)，梯度提升决策树(Gradient Boosting Decision Tree, GBDT)，多层感知机(Multilayer Perceptron, MLP)以及支持向量回归机(Support Vector Regression, SVR)。评价指标选择均匀绝对误差(Mean Absolute Error, MAE)。预测效果详见表 4.3：

表 4.3 实例数据预测效果

数据集	处理	超参数	样本量	测试集 MAE				
				KNN	RF	MLP	SVR	GBDT
Bike Sharing Demand	-	-	7620	0.022	**0.0012**	0.0054	0.0426	0.0023
	RSIS	K=150, η=10	40380	0.020	**0.0006**	0.0025	0.0412	0.0019
Air Quality	-	-	6549	0.0299	**0.0257**	0.0480	0.0387	0.0258
	RSIS	K=20, η=100	101984	0.0289	0.0255	**0.0208**	0.0384	0.0277
Forest Fires	-	-	361	**0.0406**	0.0493	0.1058	0.0736	0.0449
	RSIS	K=10, η=100	21131	**0.0389**	0.0432	0.048	0.1005	0.0414

Data sets	Optimizing	Training (Testing) set size	Models			
			KNN	MLP	SVR	GBDT
Bike Sharing Demand	-	500(1000)	5.99 (0.16)	2.31 (0.14)	64.33 (2.34)	5.25 (0.05)
		1000(1000)	4.44 (0.12)	1.65 (0.12)	42.98 (1.44)	4.45 （0.04）
		2000(1000)	3.66 (0.11)	0.47 (0.02)	26.85 (0.99)	3.37 (0.04)
	RSIS	500(1000)	3.66 (0.14)	**0.29** **(0.02)**	12.27 (1.44)	2.92 (0.05)
		1000(1000)	2.94 (0.12)	**0.18** **(0.01)**	8.32 (0.83)	2.94 (0.05)
		2000(1000)	2.66 (0.09)	**0.15** **(0.006)**	4.74 (0.34)	2.45 (0.04)
Air Quality	-	500(1000)	105.38 (0.12)	104.23 (0.12)	327.52 （0.52）	95.38 (0.11)
		1000(1000)	101.86 (0.11)	98.97 (0.11)	306.72 (0.49)	89.29 (0.10)
		2000(1000)	95.76 (0.11)	92.80 (0.10)	268 (0.44)	89.24 (0.10)
	RSIS	500(1000)	101.43 (0.11)	**89.85** **(0.10)**	218.93 (0.31)	89.92 (0.10)
		1000(1000)	96.52 (0.11)	88.77 (0.10)	174.02 (0.30)	**87.52** **(0.10)**
		2000(1000)	90.24 (0.10)	86.18 (0.10)	144.60 (0.20)	**84.18** **(0.09)**
Facebook Metrics (Normalized)	-	100(200)	0.079 (0.189)	0.072 (0.162)	0.101 (0.187)	0.0040 (0.0125)
		200(200)	0.065 (0.152)	0.070 (0.112)	0.084 (0.148)	0.0017 (0.0058)
		300(200)	0.051 (0.124)	0.055 (0.098)	0.077 (0.141)	0.0009 (0.0042)
	RSIS	100(200)	0.078 (0.181)	0.061 (0.102)	0.105 (0.198)	**0.0023** **(0.0069)**
		200(200)	0.067 (0.151)	0.048 (0.093)	0.086 (0.155)	**0.0012** **(0.0040)**
		300(200)	0.056 (0.125)	0.030 (0.053)	0.076 (0.136)	**0.0007** **(0.0029)**
Forest Fires (Normalized)	-	50(200)	0.064 (0.54)	0.177 (0.169)	0.093 (0.083)	0.063 (0.053)
		300(200)	0.060 (0.055)	0.161 (0.158)	0.080 (0.076)	0.058 (0.054)
		300(200)	0.050 (0.039)	0.093 (0.081)	0.076 (0.067)	0.055 (0.044)
	RSIS	50(200)	**0.062** **(0.052)**	0.144 (0.104)	0.124 (0.114)	0.077 (0.066)
		150(200)	0.059 (0.054)	0.058 (0.054)	0.082 (0.078)	**0.056** **(0.052)**
		300(200)	**0.048** **(0.037)**	0.068 (0.057)	0.073 (0.063)	0.054 (0.043)

通过表 2 可知，每个数据集经过 RSIS 方法处理后,都可以有效提升模型的泛化能力。并且，以上数据集包含许多非连续变量，这并不满足 RSIS 方法的假设，

以及对于样本量较少的数据集(如 Forest Fires)，在子空间之间插值时，很难保证线性拟合误差$\epsilon_i^{'} \to 0$。这表明，即使在实际应用中存在违反 RSIS 假设的情况，该方法仍可能取得较好的优化结果。

五、结合卷积神经网络框架的进一步研究

随着深度学习技术的发展，自从 CNN 在图像识别和分类方面取得重大突破以来，从最初的 LeNet 到更复杂的架构如 AlexNet、VGG、ResNet 等，CNN 已成为处理图像数据的标准工具，被广泛应用于各种视觉任务，包括对象检测、图像分割、实时视频处理等。这些应用不仅展示了 CNN 在处理复杂视觉信息方面的强大能力，也推动了从自动驾驶到医疗诊断等多个领域的技术发展。

上文的研究主要针对矢量数据使用 RSIS 进行数据合成，本章节重点探究 RSIS 方法结合 CNN 框架在计算机视觉领域的应用研究。

5.1 方法概述

RSIS 方法针对向量数据进行数据合成，但是无法直接用于图像数据，因此 RSIS 方法存在一定的局限性。CNN 作为优秀的特征提取模型，本文将 RSIS 方法与传统的深度学习 CNN 框架相结合，提出了一种创新的 CNN 框架，旨在增强图像数据的特征表示和模型泛化能力。该框架首先利用 CNN 的卷积和池化层对输入图像进行深度特征提取，然后通过全局池化层将这些特征转换为一维向量表示。在此基础上，应用 RSIS 方法来生成新的向量表示。见图 6.1。

具体来说，对于一组 RGB 图像数据集合$D = I_i, L_i{}_{i=1}^{n}$，其中I_i为图像数值化的张量数据。设 CNN 的特征提取函数$f: \mathbb{R}^{H\times W\times C} \to \mathbb{R}^{d}$，其中$H \times W \times C$表示图像的尺寸以及通道数，$d$表示提取特征的维度。通过 CNN 的特征提取操作，图像I_i被转换为一维的特征向量$v_i = f(I_i)$. 将集合$v_i, L_i{}_{i=1}^{n}$使用 RSIS 方法进行数据合成得到$v_i, L_i{}_{i=1}^{n'}$，其中n'为添加了新的合成数据后的特征向量集的样本量。最后将所有样本放入全连接网络中进行迭代训练。训练好的模型可以用于多种应用，如图像分类、对象识别或其他相关的计算机视觉任务。该方法的关键优势在于通过数据合成增强了数据集的代表性和多样性，从而提高了模型在处理新场景和变化时的泛化能力和鲁棒性。

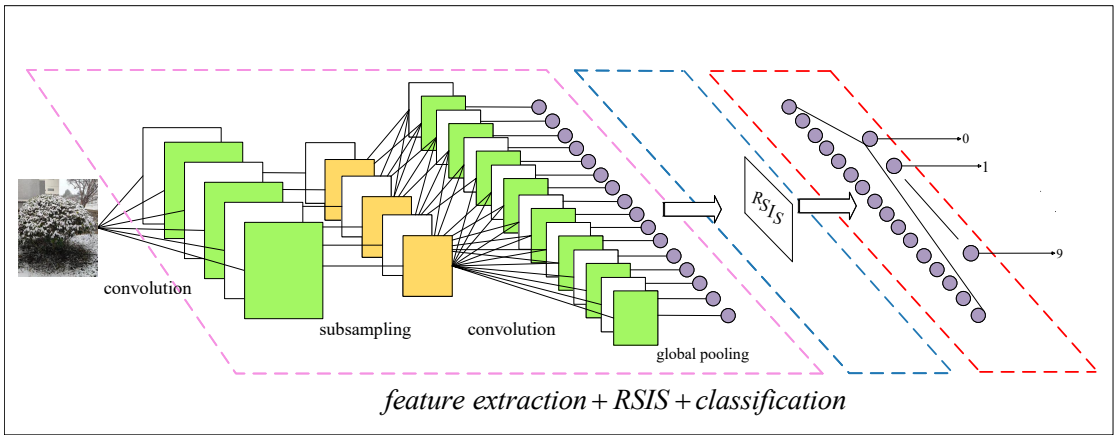

图 6.1 框架结构图

5.2 训练策略的设计

将 RSIS 方法融入 CNN 框架中，传统的模型训练策略并不适用，本文设计了一种契合该框架的模型训练策略，分为两个阶段：

在第一个阶段中，与传统的训练方式一样，依旧将图像进行数据增强处理，随后放入模型中进行前向传播计算损失函数。最后对损失函数求梯度进行反向传播操作，更新模型参数，直至模型收敛为止。

在第二阶段中，将原始图像放入第一阶段训练好的特征提取器中得到每个图像的特征向量数据集，对特征向量数据集使用 RSIS 方法合成新的数据，并将新的数据再次放入全连接层中对其进行训练直到收敛。需要注意的是，这一阶段仅仅只针对全连接层的参数进行更新，特征提取层并不参与训练过程。

5.3 优化效果分析

本实验旨在评估四种不同的深度学习模型（ResNet50、VGG16、DenseNet121、MobileNet V2）在四个公开数据集（MNIST、FashionMNIST、CIFAR-10、SVHN）上的性能。通过比较这些模型在各个数据集上在使用 RIRS 方法前后的测试误差，我们旨在确定 RIRS 方法是否对于传统模型与图像数据集有显著的提升效果。

5.3.1 数据集描述

MNIST：手写数字数据集，包含 60,000 个训练样本和 10,000 个测试样本，图像为 28x28 的灰度图。

FashionMNIST：服装图像数据集，结构与 MNIST 相同，旨在替代传统的手写数字识别任务。

CIFAR-10：含有 10 个类别的小图像数据集，每个类别有 6,000 个图像，总共有 50,000 个训练图像和 10,000 个测试图像，图像尺寸为 32x32 彩色图。

SVHN：街景房屋数字数据集，包含超过 600,000 个数字图像，用于数字识别任务。

5.3.2 模型网络描述

ResNet50：ResNet50 是残差网络（ResNet）系列中的一个变种，具有 50 层深的架构。它通过引入“残差块”来解决深层网络训练中的梯度消失问题，允许网络学习恒等映射，从而使得更深层次的网络训练成为可能。其准确率高，深度较大，参数数量较多，适用于复杂的图像识别任务。

VGG16：VGG16 是视觉几何组（Visual Geometry Group）开发的卷积神经网络模型之一，特点是其均匀的架构，包含 16 层网络，主要由卷积层和全连接层构成。结构简单，层数较深，参数量大，计算成本较高，但模型解释性好，适用于图像识别和图像分类任务。

DenseNet121：DenseNet121 是密集连接网络（DenseNet）系列中的一个，具有 121 层。它的特点是每一层都与前面的所有层相连接，极大地提高了网络的参数效率。其参数效率高，具有很强的特征提取能力，通过特征重用减少了模型的复杂度和计算量。

MobileNet V2：MobileNet V2 是专为移动和嵌入式视觉应用设计的轻量级深度学习模型。它引入了倒置残差结构，其中包括线性瓶颈层，以改进模型效率。其模型小巧，计算效率高，非常适合在计算资源有限的设备上运行，如智能手机和其他移动设备。

5.3.2 前后迭代对比

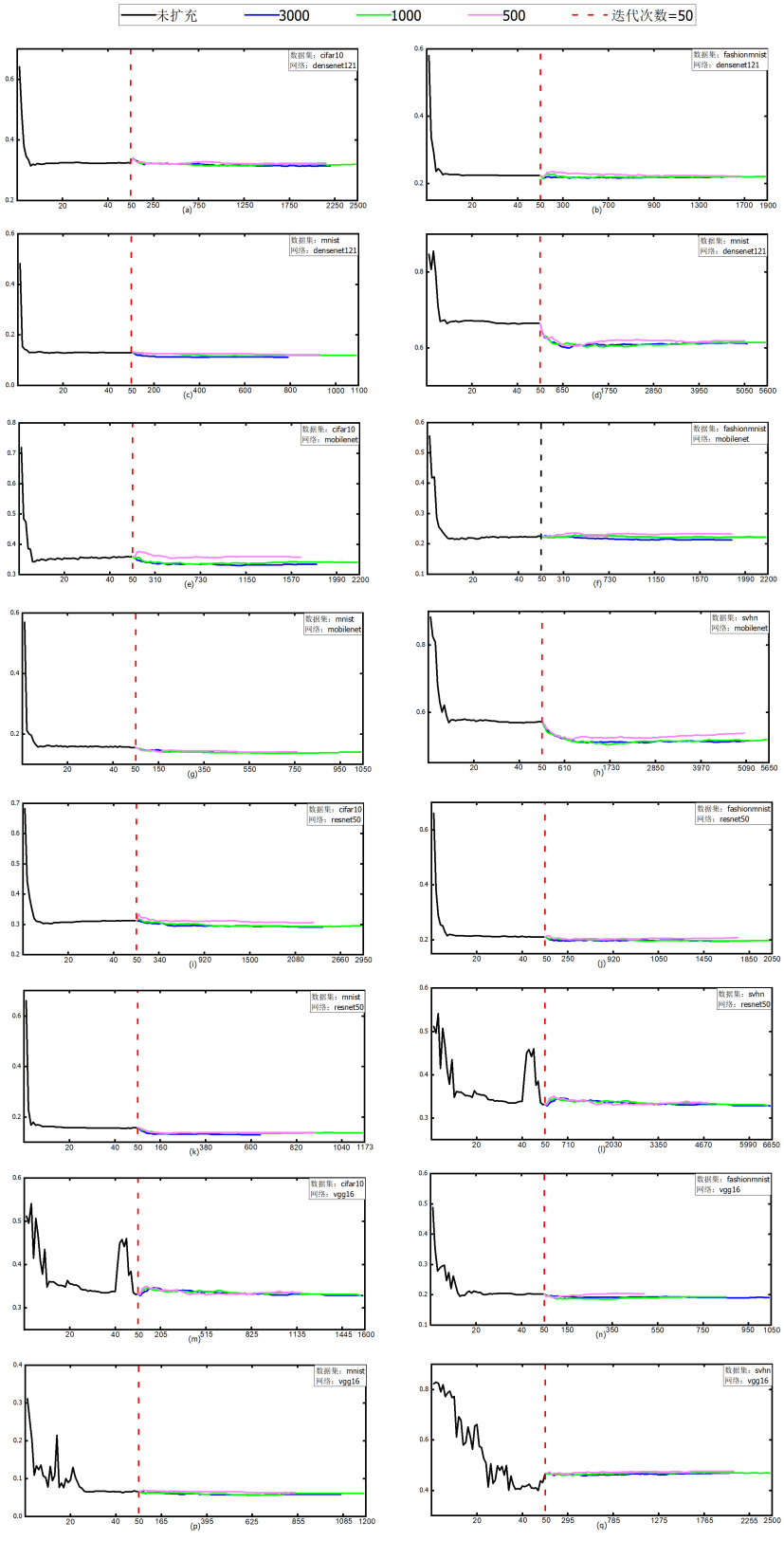
未扩充
3000
1000
500
迭代次数=50
数据集：cifar10
网络：densenet121
(a)
数据集：fashionmnist
网络：densenet121
(b)
数据集：mnist
网络：densenet121
(c)
数据集：mnist
网络：densenet121
(d)
数据集：cifar10
网络：mobilenet
(e)
数据集：fashionmnist
网络：mobilenet
(f)
数据集：mnist
网络：mobilenet
(g)
数据集：svhn
网络：mobilenet
(h)
数据集：cifar10
网络：resnet50
(i)
数据集：fashionmnist
网络：resnet50
(j)
数据集：mnist
网络：resnet50
(k)
数据集：svhn
网络：resnet50
(l)
数据集：cifar10
网络：vgg16
(m)
数据集：fashionmnist
网络：vgg16
(n)
数据集：mnist
网络：vgg16
(p)
数据集：svhn
网络：vgg16
(q)

图 6.2 预测误差迭代对比

由图 6.2 可以看出，在原本模型收敛的情况下，预测误差趋于稳定并难以再下降，而采用本文提出的方法扩充之后在全连接层继续训练，并设置早停机制，最终预测误差进一步得到下降。

表 6.1 4 种网络架构在 4 个数据集的扩充前后效果对比

Model	After RIRS	Dataset			
		MNIST	FashionMNIST	CIFAR-10	SVHN
Resnet50	500	1.70% -	0.30% -	0.50% -	3.10% -
	1000	1.90% -	1.30% -	1.60% -	3.60% -
	3000	2.70% -	1.50% -	2.10% -	3.60% -
Vgg16	500	0.20% -	0.10% +	0.30% +	0.80% +
	1000	0.50% -	0.90% -	0.40% -	0.00% ≈
	3000	0.70% -	1.10% -	0.40% -	0.00% ≈
Densenet121	500	0.70% -	0.20% -	0.00% ≈	4.70% -
	1000	1.00% -	0.30% -	0.50% -	5.10% -
	3000	1.70% -	0.60% -	1.00% -	5.40% -
Mobilenet v2	500	1.30% -	0.70% +	0.20% -	3.30% -
	1000	1.50% -	0.40% -	1.80% -	5.20% -
	3000	1.70% -	1.20% -	2.50% -	5.70% -
-		12	10	10	9
+		0	2	1	1
≈		0	0	1	2

表 6.1 显示了使用不同的深度学习模型在四个不同数据集（MNIST, FashionMNIST, CIFAR-10, 和 SVHN）上的测试误差变化幅度，并通过 Wilcoxon 检验来判断这些变化是否显著。每行表示不同的深度学习模型（Resnet50, Vgg16, Densenet121, 和 Mobilenet v2），列表示不同的数据集。百分数表示当训练集从 200 张图片扩充到 500、1000、3000 张图片时（测试集固定为 1000 张图片），测试误差的变化幅度。这里的 “-” 表示显著性检验的结果为负，即扩充后的测试误差减少；“+” 表示扩充后的测试误差显著增加；“≈” 表示变化不

显著。

样本扩充通常伴随着测试误差的显著下降，在大多数情况下显著提高了模型的泛化能力。特别是在相对简单的数据集，如 MNIST 上，所有网络架构在经过样本扩充后均显示出显著的测试误差下降，验证了样本扩充策略的有效性。进一步观察发现，随着训练集规模的增加，测试误差的下降趋势在所有数据集和模型中都更加明显，表明更大规模的训练集能够提供更丰富的信息，有助于模型挖掘数据的潜在特征。

然而，也有一些例外情况，尤其是 VGG16 在 FashionMNIST 和 SVHN 数据集上的性能，提示我们在实施样本扩充时需要考虑到特定模型与数据集的兼容性。此外，不同的网络模型对样本扩充的响应存在差异。例如，ResNet50 和 DenseNet121 在 MNIST 数据集上的表现较好，可能因为它们更适合处理手写数字识别任务。特别是 DenseNet121 在 SVHN 数据集上表现出较高的误差下降，这可能与其网络结构的深度和复杂性有关。

对于 CIFAR-10 和 SVHN 这样更复杂的数据集，样本扩充能够更显著地降低测试误差。这表明复杂数据集能从更多样化的训练样本中获益，进而改善模型的学习和泛化能力。不过，不同数据集对样本扩充的响应差异显著，尤其是当将变化较小的 MNIST 与多样性更高的 CIFAR-10 和 SVHN 进行比较时。这可能是因为对于变化不大的简单数据集，样本扩充未必能引入足够的新信息来显著改善模型性能。

此外，我们还观察到随着模型复杂度的增加，样本扩充对测试误差减少的影响不一定呈现出一致的增强趋势，暗示模型复杂度与样本扩充效果之间可能不存在直接的正相关关系。在小规模数据集上，即便是相对简单的模型如 VGG16，也能在如 FashionMNIST 和 CIFAR-10 上表现出显著的测试误差下降，表明在数据较少的情况下，样本扩充可以是提升模型性能的一种有效手段。

特定模型对某些数据集的响应特别积极，例如 DenseNet121 在 SVHN 数据集上的出色表现可能表明其架构特别适合捕捉该数据集中的图像特征，强调了选择与数据集特性相匹配的模型架构的重要性。

本文提出的 RIRS 方法，在大多数情况下是一个有效的策略，特别是对于复杂的数据集和需要更大规模训练数据的深度学习模型。在多数情况下对模型性能有积极影响，但这种影响受到多种因素的影响，包括数据集的特性、模型的结构和复杂度，以及训练样本量的大小。因此，对于不同的应用背景和要求，本文提出的 RIRS 方法应该细致规划并仔细实施。

六、总结与展望

本文提出了一种数据合成方法 RSIS，它可以自适应地调整数据集的大小，且扩展后的数据通常包含最小的实际错误。此外，它还可以调整样本的结构，这可以显著降低大误差样本的比例。对模拟数据集的实验结果表明，RSIS 可以优化样本，相比于其他方法，使用这种方法生成的数据具有更小的误差，并且能更好地处理未知噪声，具有良好的鲁棒性。在实例数据集上的实验结果显示，该方法可以适用于多种机器学习模型，并且在大多数情况下，可以提高模型的泛化能力。

本文还将 RSIS 方法结合卷机神经网络应用到计算机视觉领域，通过实验结果可知，RSIS 方法同样可以有效提升图像分类模型的泛化能力。值得注意的是，RSIS 方法在减少数据噪声影响的同时，还增强了数据的多样性以及代表性，这一点对于那些依赖于数据多样性的机器学习模型尤为重要。通过提供更准确、更干净且更具有多样性的训练数据，RSIS 方法有助于提高机器学习模型的预测性能，从而在实际应用中发挥更大的价值。在未来的研究中，可以考虑将该方法应用到进化算法、自然语言处理等领域中，进一步挖掘 RSIS 方法的潜在价值。

参考文献

[1] ALIZADEHSANI R, ROSHANZAMIR M, HUSSAIN S, 等. Handling of uncertainty in medical data using machine learning and probability theory techniques: a review of 30 years (1991–2020)[J/OL]. Annals of Operations Research, 2020. DOI:10.1007/s10479-021-04006-2.

[2] GUO Y, WANG W, WANG X. A Robust Linear Regression Feature Selection Method for Data Sets With Unknown Noise[J/OL]. IEEE Transactions on Knowledge and Data Engineering, 2023, 35(1): 31-44. DOI:10.1109/TKDE.2021.3076891.

[3] JESMEEN M, HOSSEN J, SAYEED S, 等. A survey on cleaning dirty data using machine learning paradigm for big data analytics[J]. Indonesian Journal of Electrical Engineering and Computer Science, 2018, 10(3): 1234-1243.

[4] MA T, LU S, JIANG C. A membership-based resampling and cleaning algorithm for multi-class imbalanced overlapping data[J/OL]. Expert Systems with Applications, 2024, 240: 122565. DOI:10.1016/j.eswa.2023.122565.

[5] ASHRAF H, SHAH B, SOOMRO A M, 等. Ambient-noise Free Generation of Clean Underwater Ship Engine Audios from Hydrophones using Generative Adversarial Networks[J/OL]. Computers and Electrical Engineering, 2022, 100: 107970. DOI:10.1016/j.compeleceng.2022.107970.

[6] JIANG Z, ZHOU W, SHAN C, 等. Design of robust Gaussian approximate filter and smoother with latency probability identification[J/OL]. ISA Transactions, 2023, 137: 405-418. DOI:10.1016/j.isatra.2023.01.033.

[7] CHENG M, SHI D, CHEN T. Event-triggered risk-sensitive smoothing for linear Gaussian systems[J/OL]. Automatica, 2023, 158: 111301. DOI:10.1016/j.automatica.2023.111301.

[8] WANG J, PEI Z K, WANG Y, 等. An investigation of income inequality through autoregressive integrated moving average and regression analysis[J/OL]. Healthcare Analytics, 2024, 5: 100287. DOI:10.1016/j.health.2023.100287.

[9] JU H, HONDA N, YOSHIMURA S H, 等. Multidimensional fractal scaling

analysis using higher order moving average polynomials and its fast algorithm[J/OL]. Signal Processing, 2023, 208: 108997. DOI:10.1016/j.sigpro.2023.108997.

[10] YUNLONG W, HUI L, ZHENGBO Z, 等. Outlier detection algorithm for satellite gravity gradiometry data using wavelet shrinkage de-noising[J/OL]. Geodesy and Geodynamics, 2012, 3(2): 47-52. DOI:10.3724/SP.J.1246.2012.00047.

[11] TAKAHASHI K, OOKA R, KUROSAKI A. Seasonal threshold to reduce false positives for prediction-based outlier detection in building energy data[J/OL]. Journal of Building Engineering, 2024, 84: 108539. DOI:10.1016/j.jobe.2024.108539.

[12] LONG Y, PAN J, XI Y, 等. Full Image-Index Remainder Based Single Low-Dose DR/CT Self-supervised Denoising[C]//International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, 2023: 466-475.

[13] GOEL N, SINGARAVELU M, GUPTA S, 等. Parameterized Clustering Cleaning Approach for High-Dimensional Datasets with Class Overlap and Imbalance[J]. SN Computer Science, 2023, 4(5): 464.

[14] AL-FAHDAWI S, AL-WAISY A S, ZEEBAREE D Q, 等. Fundus-DeepNet: Multi-label deep learning classification system for enhanced detection of multiple ocular diseases through data fusion of fundus images[J/OL]. Information Fusion, 2024, 102: 102059. DOI:10.1016/j.inffus.2023.102059.

[15] RIX T, DREHER K K, NÖLKE J H, 等. Efficient Photoacoustic Image Synthesis with Deep Learning[J/OL]. Sensors, 2023, 23(16): 7085. DOI:10.3390/s23167085.

[16] YAPI D, NOUBOUKPO A, ALLILI M S. Mixture of multivariate generalized Gaussians for multi-band texture modeling and representation[J/OL]. Signal Processing, 2023, 209: 109011. DOI:10.1016/j.sigpro.2023.109011.

[17] CHEN B, FAN Y, LAN W, 等. Fuzzy support vector machine with graph for classifying imbalanced datasets[J/OL]. Neurocomputing, 2022, 514: 296-312. DOI:10.1016/j.neucom.2022.09.139.

[18] AKKEM Y, BISWAS S K, VARANASI A. A comprehensive review of synthetic data generation in smart farming by using variational autoencoder and generative

adversarial network[J/OL]. Engineering Applications of Artificial Intelligence, 2024, 131: 107881. DOI:10.1016/j.engappai.2024.107881.

[19]AZIM S, ARORA M, RANGANATHA N E, 等. A Survey of Autoregressive Models for Image and Video Generation[J].

[20]TURHAN C G, BILGE H S. Recent Trends in Deep Generative Models: a Review[C/OL]//2018 3rd International Conference on Computer Science and Engineering (UBMK). Sarajevo: IEEE, 2018: 574-579[2024-01-27]. https://ieeexplore.ieee.org/document/8566353/. DOI:10.1109/UBMK.2018.8566353.

[21]SAEZ J A, LUENGO J, HERRERA F. Dealing with noise problem in machine learning data-sets: A systematic review[J]. Procedia Computer Science, 2019, 159: 244-253.

[22]FRÉNAY B, KABÁN A. Classification in the presence of label noise: a survey[J]. IEEE Transactions on Neural Networks and Learning Systems, 2013, 25(5): 845-869.

[23]YI K, WU J. Probabilistic end-to-end noise correction for learning with noisy labels[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019: 7017-7025.

[24]EVANS R, GREFENSTETTE E. Learning explanatory rules from noisy data[J]. Journal of Artificial Intelligence Research, 2018, 61: 1-64.

[25]WU Z, RINCON D, LUO J, 等. Improving Machine Learning Modeling of Nonlinear Processes Under Noisy Data Via Co-teaching Method[C]//2021 American Control Conference (ACC). IEEE, 2021: 4660-4666.

[26]WU Z, RINCON D, LUO J, 等. Handling noisy data in machine learning modeling and predictive control of nonlinear processes[C]//2021 American Control Conference (ACC). IEEE, 2021: 3345-3351.

[27]KIM Y, YIM J, YUN J Y, 等. NLNL: Negative Learning for Noisy Labels[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019: 101-110.

[28]OLEGHE O. A predictive noise correction methodology for manufacturing process

datasets[J]. Journal of Big Data, 2020, 7(1): 1-27.

[29]HAN J, LUO P, WANG X. Deep self-learning from noisy labels[C]//Proceedings of the IEEE/CVF international conference on computer vision. 2019: 5138-5147.

[30]BHATIA A, CHUG A, SINGH A P, 等. A hybrid approach for noise reduction-based optimal classifier using genetic algorithm: A case study in plant disease prediction[J]. Intelligent Data Analysis, 2022, 26(4): 1023-1049.

[31]QAISAR S M, DALLET D. ECG noise removal and efficient arrhythmia identification based on effective signal-piloted processing and machine learning[C]//2021 IEEE International Instrumentation and Measurement Technology Conference (I2MTC). IEEE, 2021: 1-6.

[32]LAPINS S, BUTCHER A, KENDALL J, 等. DAS-N2N: Machine learning Distributed Acoustic Sensing (DAS) signal denoising without clean data[J]. arXiv preprint arXiv:2304.08120, 2023.

[33]PARIZAD A, HATZIADONIU C J. Cyber-attack detection using principal component analysis and noisy clustering algorithms: A collaborative machine learning-based framework[J]. IEEE Transactions on Smart Grid, 2022, 13(6): 4848-4861.

[34]MUNDRA S, VIJAY S, MUNDRA A, 等. Classification of imbalanced medical data: An empirical study of machine learning approaches[J]. Journal of Intelligent & Fuzzy Systems, 2022, 43(2): 1933-1946.

[35]BREUNIG M M, KRIEGEL H P, NG R T, 等. LOF: identifying density-based local outliers[C]//Proceedings of the 2000 ACM SIGMOD international conference on Management of data. 2000: 93-104.

[36]DÜNDAR H, SOYSAL M, ÖMÜRGÖNÜLŞEN M, 等. A green dynamic TSP with detailed road gradient dependent fuel consumption estimation[J/OL]. Computers & Industrial Engineering, 2022, 168: 108024. DOI:10.1016/j.cie.2022.108024.

[37]GHEYSARI K, KHOEI A, MASHOUFI B. High speed ant colony optimization CMOS chip[J/OL]. Expert Systems with Applications, 2011, 38(4): 3632-3639. DOI:10.1016/j.eswa.2010.09.017.

[38]YU Y, LIAN F, YANG Z. Pricing of parcel locker service in urban logistics by a TSP model of last-mile delivery[J/OL]. Transport Policy, 2021, 114: 206-214. DOI:10.1016/j.tranpol.2021.10.002.

[39]AKHAND M A H, AYON S I, SHAHRIYAR S A, 等. Discrete Spider Monkey Optimization for Travelling Salesman Problem[J/OL]. Applied Soft Computing, 2020, 86: 105887. DOI:10.1016/j.asoc.2019.105887.

[40]WANG C, MA B, SUN J. A co-evolutionary genetic algorithm with knowledge transfer for multi-objective capacitated vehicle routing problems[J/OL]. Applied Soft Computing, 2023, 148: 110913. DOI:10.1016/j.asoc.2023.110913.

[41]TOAZA B, ESZTERGÁR-KISS D. A review of metaheuristic algorithms for solving TSP-based scheduling optimization problems[J/OL]. Applied Soft Computing, 2023, 148: 110908. DOI:10.1016/j.asoc.2023.110908.

[42]MICHALAK K. Feasibility-Preserving Genetic Operators for Hybrid Algorithms using TSP solvers for the Inventory Routing Problem[J/OL]. Procedia Computer Science, 2021, 192: 1451-1460. DOI:10.1016/j.procs.2021.08.149.

[43]LIU C, LI B, VOROBEYCHIK Y, 等. Robust linear regression against training data poisoning[C]//Proceedings of the 10th ACM workshop on artificial intelligence and security. 2017: 91-102.

[44]CLEVELAND W S. Robust locally weighted regression and smoothing scatterplots[J]. Journal of the American statistical association, 1979, 74(368): 829-836.

[45]LEE H B, YANG E, HWANG S J. Deep asymmetric multi-task feature learning[C]//International Conference on Machine Learning. PMLR, 2018: 2956-2964.

[46]MRINI K, DERNONCOURT F, YOON S, 等. A gradually soft multi-task and data-augmented approach to medical question understanding[C]//Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). 2021: 1505-1515.

[47]DUONG L, COHN T, BIRD S, 等. Low resource dependency parsing: Cross-

lingual parameter sharing in a neural network parser[C]//Proceedings of the 53rd annual meeting of the Association for Computational Linguistics and the 7th international joint conference on natural language processing (volume 2: short papers). 2015: 845-850.
[48] MISRA I, SHRIVASTAVA A, GUPTA A, 等. Cross-stitch networks for multi-task learning[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 3994-4003.
[49] Lan 等 - 2019 - A hybrid SCA–VNS meta-heuristic based on Iterated .pdf[Z].
[50] Matchings_in_Groups_and_Vector.pdf[Z].
[51] SharkID A Framework for Automated Individual Shark Identification.pdf[Z].
[52] ZUKHRUF F, BALIJEPALLI C, FRAZILA R B, 等. Algorithms for restoring disaster-struck seaport operations considering interdependencies between infrastructure availability and repair team assignments[J/OL]. Computers & Industrial Engineering, 2023, 175: 108894. DOI:10.1016/j.cie.2022.108894.
[53] DRENT C, KEIZER M O, HOUTUM G J V. Dynamic dispatching and repositioning policies for fast-response service networks[J/OL]. European Journal of Operational Research, 2020, 285(2): 583-598. DOI:10.1016/j.ejor.2020.02.014.
[54] ENAYATI S, MAYORGA M E, RAJAGOPALAN H K, 等. Real-time ambulance redeployment approach to improve service coverage with fair and restricted workload for EMS providers[J/OL]. Omega, 2018, 79: 67-80. DOI:10.1016/j.omega.2017.08.001.
[55] AKBARZADEH B, MAENHOUT B. A decomposition-based heuristic procedure for the Medical Student Scheduling problem[J/OL]. European Journal of Operational Research, 2021, 288(1): 63-79. DOI:10.1016/j.ejor.2020.05.042.
[56] DUAN R, PETTIE S. Linear-Time Approximation for Maximum Weight Matching[J/OL]. Journal of the ACM, 2014, 61(1): 1-23. DOI:10.1145/2529989.

附录 相关证明补充

proof of lemma 1:

$$
\begin{aligned}
f(\omega\beta_1+(1-\omega)\beta_2)-\omega f(\beta_1)-(1-\omega)f(\beta_2)&\\
&=\|y-X(\omega\beta_1+(1-\omega)\beta_2)\|_2^2-\omega\|y-X\beta_1\|_2^2-(1-\omega)\|y-X\beta_2\|_2^2\\
&=\|\omega y+(1-\omega)y-X(\omega\beta_1+(1-\omega)\beta_2)\|_2^2-\omega\|y-X\beta_1\|_2^2-(1-\omega)\|y-X\beta_2\|_2^2\\
&=\|\omega(y-X\beta_1)+(1-\omega)(y-X\beta_2)\|_2^2-\omega\|y-X\beta_1\|_2^2-(1-\omega)\|y-X\beta_2\|_2^2\\
&=\omega^2\|y-X\beta_1\|_2^2+(1-\omega)^2\|y-X\beta_2\|_2^2+2\omega(1-\omega)\langle y-X\beta_1,y-X\beta_2\rangle\\
&\quad-\omega\|y-X\beta_1\|_2^2-(1-\omega)\|y-X\beta_2\|_2^2\\
&=\omega^2\|y-X\beta_1\|_2^2+(1-2\omega+\omega^2)\|y-X\beta_2\|_2^2+(2\omega-2\omega^2)\langle y-X\beta_1,y-X\beta_2\rangle\\
&\quad-\omega\|y-X\beta_1\|_2^2-(1-\omega)\|y-X\beta_2\|_2^2\\
&=\omega^2(\|y-X\beta_1\|_2^2-2\langle y-X\beta_1,y-X\beta_2\rangle+\|y-X\beta_2\|_2^2)\\
&\quad+(1-2\omega)\|y-X\beta_2\|_2^2+2\omega\langle y-X\beta_1,y-X\beta_2\rangle\\
&\quad-\omega\|y-X\beta_1\|_2^2-(1-\omega)\|y-X\beta_2\|_2^2\\
&=\omega^2\|(y-X\beta_1)-(y-X\beta_2)\|_2^2+2\omega\langle y-X\beta_1,y-X\beta_2\rangle\\
&\quad-\omega\|y-X\beta_1\|_2^2-\omega\|y-X\beta_2\|_2^2\\
&=\omega^2\|(y-X\beta_1)-(y-X\beta_2)\|_2^2-\omega\|(y-X\beta_1)-(y-X\beta_2)\|_2^2\\
&=\omega(\omega-1)\|(y-X\beta_1)-(y-X\beta_2)\|_2^2\\
&\leq 0
\end{aligned}
$$

proof of lemma 2:

$$
\begin{aligned}
f(\omega\beta_1+(1-\omega)\beta_2)&=\|\omega\beta_1^1+(1-\omega)\beta_2^1-(\omega\beta_1^2+(1-\omega)\beta_2^2)\|_1\\
&=\|\omega(\beta_1^1-\beta_1^2)+(1-\omega)(\beta_2^1-\beta_2^2)\|_1\\
&\leq\omega\|\beta_1^1-\beta_1^2\|_1+(1-\omega)\|\beta_2^1-\beta_2^2\|_1\\
&=\omega f(\beta_1)+(1-\omega)f(\beta_2)
\end{aligned}
$$