分类号：TP39　　学校代码：10363
密　级：公开　　学　　号：2210910104

安徽工程大学
Anhui Polytechnic University

硕士学位论文

题目　特征演化数据流的分类方法研究

论 文 作 者　陈燕菲
指 导 教 师　刘三民 教授 王精明 教授
学科（专业）　软件工程
研 究 方 向　领域软件工程

论文提交日期：2024 年 5 月 23 日

分类号：TP39　　单位代码：10363
密　级：公开　　学　　号：2210910104

题　　目 特征演化数据流的分类方法研究

英文并列

题　　目 **Research on Classification Method for Feature Evolvable Data Streams**

学生姓名：陈燕菲
指导教师：刘三民 教授、王精明 教授
专　　业：软件工程
研究方向：领域软件工程

论文答辩日期：2024 年 5 月 24 日

安徽工程大学学位论文原创性声明

本人郑重声明：我恪守学术道德，崇尚严谨学风。所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已明确注明和引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的作品及成果的内容。论文为本人亲自撰写，我对所写的内容负责，并完全意识到本声明的法律后果由本人承担。

学位论文作者签名：陈燕菲

日期：2024 年 5 月 23 日

安徽工程大学学位论文版权使用授权书

学位论文作者完全了解学校有关保留、使用学位论文的规定，同意学校保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅或借阅。本人授权安徽工程大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

保密 □，在___年解密后适用本版权书。

本学位论文属于

不保密☑。

学位论文作者签名：陈燕菲　指导教师签名：刘三民

日期：2024年5月23日　日期：2024年5月23日

特征演化数据流的分类方法研究

摘 要

在大数据时代，海量数据往往以流的形式出现并积累，如何从动态实时产生的数据流中挖掘潜在重要信息成为目前研究热点之一。由于数据流学习环境的动态可变性，导致数据的特征空间随时间推移而发生变化，这种现象被称为特征演化。传统数据流分类算法囿于固定特征空间的假设，直接应用于特征演化数据流的学习将严重损害模型分类性能，甚至导致学习进程停止。因此，特征演化数据流的分类任务存在如下问题：（1）学习过程中不断抛弃已训练好的分类器而重新训练，这将造成巨大的资源浪费，如何在不同特征空间下重新利用历史分类器，从而对新特征数据进行适应性学习；（2）新特征空间下训练样本过少，不足以支持充分训练新分类器，如何构建并维护性能稳定的分类模型以提高整体分类精度；（3）数据流中往往伴随噪声干扰、类不平衡以及概念漂移等现象，如何提高分类模型的泛化能力，以降低这些因素对模型性能的负面影响。

为此，本文主要研究内容在于构建特征演化数据流的分类模型，同时对数据流中噪声干扰、类不平衡等问题进行处理，具体研究工作如下：

（1）针对特征演化的自适应问题，提出基于映射策略的特征演化数据流分类方法。该方法通过利用新旧特征共存阶段的数据，学习不同特征空间之间的映射关系矩阵，从而将训练良好的分类模型投影至新特征空间下继续训练，避免由于新特征数据量少而导致重新训练的分类器性能不佳；同时，引入模糊隶属函数对训练样本中的噪声点加以区分，减少异常值对决策面的干扰。最后，通过大量

对比实验验证了所提方法的有效性。

（2）针对分类模型的稳定性问题，提出基于集成学习的特征演化数据流分类方法。该方法基于增量孪生支持向量机对模型进行训练与更新；在出现新特征时重新训练新分类器，同时将已训练好的旧分类器投影至新特征空间继续学习；最后，为充分利用各阶段训练的分类模型，提出两种不同的集成策略以合并新旧分类器实现共同预测。实验结果表明，所提方法能够构建高效稳定的分类模型，同时具有良好的抗噪声能力。

（3）针对数据流中的类不平衡问题，提出具有类不平衡的特征演化数据流在线学习方法，适用于更现实的特征演化场景下数据流挖掘工作。首先，利用特征空间划分与分类器投影，自适应训练特征空间时刻变化的数据样本；然后，将代价敏感指标最小化问题融入模型在线优化目标函数中，根据不平衡率定义新的代价敏感因子，动态调整类别权重以解决类不平衡问题；最后，利用变异系数筛选出重要特征，从而对分类器稀疏截断处理。实验结果表明，该方法能够对特征演化数据流自适应学习，并能有效处理数据流中的类不平衡问题。

综上所述，本文通过研究数据流中的特征演化问题，并结合特征映射、集成学习和类不平衡处理等方法，为特征演化数据流的分类任务提出一系列解决方案，具有良好的学术价值与应用前景。

关键词：数据流分类，特征演化，类不平衡，集成学习，在线学习

RESEARCH ON CLASSIFICATION METHOD FOR FEATURE EVOLVABLE DATA STREAMS

ABSTRACT

In the era of big data, lots of data frequently emerge and accumulate in the form of streams. Mining valuable information from dynamically generated data streams has become one of the research hot. Due to the dynamic variability of data stream, the feature space undergoes temporal changes which are referred to as feature evolution. Traditional data stream classification algorithms are constrained by the assumption of a fixed feature space. Applying them to learn from feature evolvable data streams directly may significantly impair the classification performance and potentially hinder the learning process. Therefore, the classification task for feature evolvable data streams presents several problems: (1) During the course of learning, discarding and retraining the well-trained classifiers often results in substantial resource loss. Thus, how to reuse the historical classifier under different feature spaces is important to learn new feature data fast; (2) There are insufficient training samples in the new feature space to support sufficient training of new classifiers. Therefore, how to construct and maintain an efficient classification model to improve overall classification accuracy is critical; (3) Data streams often come with noise pollution, class imbalance and concept drift phenomena. How to enhance the generalization ability of classification models to reduce the negative impact of these factors is one of the key problems.

Therefore, the main research content of this dissertation is to construct an adaptive classification model for feature evolvable data

streams and address issues such as noise interference and class imbalance in data streams. The specific research works are as follows:

(1) To enhance the adaptability to feature evolution, a feature evolvable data streams classification method based on mapping strategy is proposed. This method uses the coexistence stage of new and old features to learn the mapping matrix between different feature spaces. As a result, the well-trained model can be projected into the new feature space, and it avoids the poor performance of the classifier due to limited amount of new feature data. Additionally, the fuzzy membership function is introduced to distinguish the noise points in the training samples and reduce the interference of outliers on the decision boundaries. Finally, a large number of comparative experiments are conducted to verify the effectiveness of the proposed method.

(2) To ensure the stability of the classification model, a feature evolvable data stream classification method based on ensemble learning is proposed. This method employs incremental twin support vector machines for training and updating the models. When new features emerge, the new classifier is retrained and the old classifier is projected to the new feature space for continued learning. Finally, to make full use of the classification models trained in each stage, two different ensemble strategies are proposed to combine the old classifiers and new classifiers for prediction. The experimental results show that the proposed method can construct an efficient and stable classification model which also has good anti-noise ability.

(3) To tackle the class imbalance problem within data streams, an online learning method for feature evolvable data stream with class imbalance is provided in this dissertation, which is suitable for mining data stream in more realistic feature evolution scenarios. Firstly, the

feature space division and classifier projection were used to train the data samples of the feature space changing at any time. Then, the cost-sensitive minimization problem is integrated into the online optimization objective function of the model. In order to solve the class imbalance problem, we define a new cost-sensitive factor based on the imbalance rate to adjust the class weight dynamically. Furthermore, the coefficient of variation is used to identify the important features, so that the classifier can be sparse and truncated. A large number of experimental results show that the proposed method can learn the feature evolvable data stream adaptively, and can deal with the class imbalance problem in the data stream effectively.

In summary, this dissertation investigates the feature evolution problem in data streams and integrates methods such as feature mapping, ensemble learning and class imbalance learning. It proposes a series of solutions for the classification task of feature evolvable data streams, which has good academic value and promising application prospects.

KEY WORDS: data stream classification, feature evolution, class imbalance, ensemble learning, online learning

目录

第1章 绪论

1.1 研究背景与意义

随着大数据、云计算以及物联网等技术的飞速发展，越来越多的应用平台以及物联网终端生成的数据以流的形式快速聚集，呈现出大数据特征，例如传感器网络、推荐系统、视频监控网络等[1]。数据流中蕴含大量有价值的信息，如何提取数据流中这些有价值的信息，引起学术界和工业界的关注，相关研究方兴未艾[2]。传统的数据流挖掘算法在封闭静态环境下运行，然而在现实场景中数据流的特征空间、数据分布和类别标签等都可能随时间推移发生不可预测的变化[3]，这给面向数据流的分类挖掘模型泛化能力提出新的挑战。基于此，Zhou 等[4]对开放环境中的机器学习（Open-environment Machine Learning，Open ML）问题范畴进行归纳，强调未来在开放环境中的机器学习技术应能够成功应对这些重要因素的变化。曾任国际人工智能顶级会议 AAAI 大会主席的 ThomasG。Dietterich 教授同样提出需要考虑开放动态的学习环境，创建鲁棒的机器学习模型是解决相关问题的有效途径之一[5]。

现有的面向开放动态环境数据流分类方法大多可以满足内存受限、单遍扫描、实时响应以及概念漂移检测等约束[3,6-8]。尽管如此，这些动态数据流挖掘模型建立在同构特征空间的基础上，而并未考虑数据流的特征空间动态可变的场景。在本课题中，将数据流的特征空间动态可变的现象称为特征演化[9]，这种现象经常出现在环境监控[10]、智慧医疗[11]、异常检测[12]、热点话题[13,14]等领域中。例如，在智慧医疗领域，通过各种传感设备收集应诊者身体机能特征用以表征健康状况。为使预测结果更全面精准或受到传感器寿命限制，会在传感器寿命结束之前在旧传感器附近部署新的传感器，由于数量和位置发生变化，传感器收集的数据流的特征空间将随之改变。数据流中特征空间的变化，将导致训练好的模型性能显著下降甚至完全失效，因此特征演化的数据流分类工作引起广泛关注。

此外，数据流中的类不平衡问题同样亟待解决。由于少数类的误分类代价远高于多数类的误分类代价，模型训练时应更关注少数类的分类情况，例如在癌症诊断中，将癌症患者误诊为健康人群将导致严重后果[15]。目前研究人员主

要从特征选择、数据分布调整和训练算法改进三个方面展开类不平衡数据流的问题研究。然而，传统的类不平衡学习方法，都需要在完整的特征空间下进行，不适用于特征演化数据流的分类任务。

综上所述，针对数据流中特征演化问题以及更为复杂的动态变化场景，本文设计一种高效稳定的学习框架，以提高特征演化数据流的分类模型的准确度与泛化性能，具有良好的研究价值与应用前景。

1.2 国内外研究现状

1.2.1 数据流挖掘

近年来，国内外学者在数据流挖掘领域进行了广泛而深入的研究，根据处理数据的方式可以将分类算法分为增量学习算法、批处理学习算法和在线学习算法。增量学习要求学习模型接收新数据时，提炼与整合新知识的同时保证不影响或覆盖已有知识。批处理学习通过短期离线学习，将数据流划分为一系列数据块，在这些数据块上分批次进行模型训练。在线学习的核心优势在于其增量式的学习特性：每当接收到新样本，则迅速对模型进行增量更新，并不断利用当前模型对未知样本进行预测，一旦对样本处理完毕则及时释放内存，而无需再次访问。在线学习不仅可以处理数据流带来的样本无限量和实时分析的挑战，还不依赖于样本独立同分布的假设，从而使其在处理大规模流式数据时表现出色[16]。正因如此，在线学习领域得到广泛的发展，并相继涌现了许多经典算法。其中，在线梯度下降（Online Gradient Descent，OGD）[17]作为简单高效的一阶在线学习方法，其核心思想是在每轮学习中根据瞬时损失函数的负梯度方向对模型进行修正，以保证损失最小化，然后将修正后的模型投影到可行域内以确保其满足约束条件。被动-主动算法（Passive-Aggressive learning，PA）[18]借鉴 OGD 算法的核心思想的基础上，充分考虑预测的置信水平，即样本到当前决策边界的距离，这使得算法在更新模型时能够针对每个样本的置信水平进行学习步长的差异化调整。该方法不仅提高了算法的学习效率，还增强了其对于不同置信水平样本的处理能力，从而提升了算法的整体性能。置信度加权学习（Confidence-Weighted，CW）[19]是一种二阶在线学习方法，作为对 PA 算法的一种拓展，不仅使用瞬时损失函数的函数值和次梯度信息，还进一步引入瞬时损失函数的 Hessian 矩阵信息，从而能够更精确地刻画损失函数的局部性

质。自适应权重正则化方法（Adaptive Regularization Of Weights，AROW）[20]则是对 CW 算法的一种改进，通过对 CW 算法中的约束条件进行放松，并将其转化为正则化项，进而将其整合到 KL 散度函数中。这使得算法在每次学习中仅需解决一个无约束的优化问题，不仅简化了计算过程，还提高了算法的鲁棒性，使其在面对复杂多变的数据环境时能够保持稳定的性能。

数据流分类方法根据分类器个数可分为：单分类器算法和集成学习算法。单分类器算法以灵活简便的优势广泛应用于数据挖掘任务中，主要包括决策树[21]、贝叶斯[22]、支持向量机[23]和 K 近邻[24]等方法，其中支持向量机（Support Vector Machines，SVM）是一种精确的数据分类算法，具有完善的数学基础，但也存在计算速度慢、模型鲁棒性差的缺点。孪生支持向量机（Twin Support Vector Machines，TSVM）[25]为每个类构造一个超平面，将大的分类问题分成两个小的分类问题，可保证较高的分类精度和较高的计算速度。模糊支持向量机（Fuzzy Support Vector Machine，FSVM）[26]旨在处理噪声点或离群点的分类问题，该算法通过为每个实例样本分配模糊隶属度值，实现对不同样本点的差异化处理，从而在构建最优分类超平面的过程中，能够根据不同样本点的贡献度进行权衡。FSVM[26]能够有效降低噪声或离群点对最优决策面的不良影响，并在复杂数据环境下保持较高的分类准确性和鲁棒性。

数据流集成分类方法[2,27]使用不同的训练子集来构造多个基分类器，通过多个基分类器投票来预测未知样本。例如 Online Bagging & Boosting 算法[28]、Accuracy-Weighted Ensemble 算法[29]以及 Meta-knowledge Ensemble 算法[30]等。通过基分类器的选择和组合，Grossi 等[31]提出一种基于自适应行为的选择性集成方法来处理数据流分类问题，该方法使用一个动态更新的阈值来控制哪些分类器可以被激活，只有与权重最高的最佳分类器之间的差值不超过阈值的分类器才会被激活，但该方法容易受到阈值选择的影响。如果选择的阈值过高，则可能会遗漏某些有效的分类器，导致性能下降；而如果选择的阈值过低，则可能会激活太多的分类器，导致分类器之间的冲突和干扰，进而影响性能。AUE 和 AUE2[32]用均方误差来评估基分类器的性能，并使用均方误差来表示基分类器的权重，使用加权投票进行预测，但是该方法是基于块的集成分类方法，容易受到块大小的影响。虽然目前已经有关于选择性集成分类的方法，但是大多

数方法没有同时考虑基分类器的多样性和准确性，且不能同时适用于多类不平衡、概念漂移等问题的数据流。因此，为提高基分类器的多样性，通过随机采样训练不同的 Hoeffding 树作为基分类器，并根据所提的评价方法选择性能优异的基分类器进行组合。

1.2.2 类不平衡学习

类不平衡问题是数据挖掘领域重要挑战之一。当数据流中的不同类别样本数量极不均衡时，即出现类不平衡问题。这种不平衡可能导致传统的机器学习模型偏向于多数类，从而忽视少数类，使得模型在少数类上的预测性能不佳。现有类不平衡数据流学习算法主要分为特征选择、数据分布调整和训练算法改进三个方面。

（1）特征选择。特征选择的目的是过滤冗余特征，通过仅保留更能体现类不平衡的特征的策略，使分类器在类不平衡的前提下达到较好的性能[33]。Fu 等[34]基于海灵格距离和稀疏正则化技术提出 sssHD 算法，该算法实现稳定的稀疏特征选择，并将其应用于复杂的类不平衡数据。文献[35]中提出基于距离的特征选择方法，首先识别数据中存在的不同概念，然后根据类之间以及子概念之间的距离计算有效距离度量以解决类间和类内同时发生的不平衡。

（2）数据分布调整。主要通过数据欠采样、过采样以及混合采样等手段使类别在一定程度上达到平衡[36]。CHAWLA 等[37]提出的基于人工合成样本的过采样方法（Synthetic Minority Over-sampling Technique，SMOTE）是一种经典的过采样方法，该算法通过线性插值的形式，在少数类样本之间生成人工样本，从而增加少数类样本的数量，保持数据流中的类别平衡。SMOTE 方法能够有效提高分类器的整体性能，但也同时面临对噪声样本过拟合与增加类重叠问题。为此，Soltanzadeh 等[38]提出基于范围控制的过采样方法（Range-Controlled SMOTE，RCSMOTE），通过采用样本分类方案来识别适合过采样的小样本，并根据数据特征计算安全范围以提供安全的过采样区域。文献[39]结合 K 近邻算法对多数类与少数类样本数量进行计算，然后结合 ADASYN 和 SMOTE 算法对少数类样本进行合成，降低类不平衡带来的影响。SMOTEBoost[40]和 RAMOBoost[41]将过采样或欠采样的抽样方法与集成算法相结合，改变训练数据的分布以保持类别平衡，不过这类方法往往不适用于不平衡度过高的情况。

（3）训练算法改进。在算法层面加入类别惩罚因子或权重稀疏策略[42]。其中，代价敏感学习可较好地处理数据中类不平衡问题[43]。主要思路是通过设计代价矩阵，根据不同的误分类代价给分类器分配不同的惩罚力度[44]。文献[45]提出基于在线梯度下降技术的代价敏感在线分类框架（Cost-Sensitive Online Classification，CSOGD），将在线学习和成本敏感分类结合，直接优化预定义的代价敏感指标。虽然该方法能够处理成本敏感的在线分类任务，但由于只利用梯度的加权平均值这类一阶信息而无法显著提高分类性能。为此，Zhao 等[46]提出基于置信加权策略的自适应正则化成本敏感在线梯度下降算法，综合考虑特征之间的相关性等二阶信息，并进一步引入草图技术，在性能和效率之间取得更好的平衡。

1.2.3 特征演化学习

近年来，针对数据流中特征演化（feature evolution）问题的研究工作相继展开，并取得一系列研究成果。Hou 等[47]提出一种新的特征演化流学习范式（Feature Evolvable Streaming Learning，FESL），通过特征映射恢复消失特征空间数据并利用在线梯度下降算法对模型进行更新，结合集成策略实现特征演化流的学习。但该学习范式仅适用于特定的特征演化流，即先后出现两个完全不同的特征空间，且二者需有共存期。基于 FESL 算法的思想，Liu 等[48]提出一种基于被动攻击更新策略的特征演化学习方法，该方法采用一阶在线学习算法加快模型收敛速度和提升分类性能。随后，他们在文献[49]中提出特征加权的置信-加权学习算法，进一步挖掘数据特征相关性和二阶信息提升分类准确性，但同时带来较高的时空复杂度。文献[50]提出具有增量和递减特征的一遍学习方法，通过压缩与拓展两个阶段实现特征信息的保留与传递，以辅助对当前数据的分类预测精度。面对特征部分消失与出现的特征演化场景，文献[51]提出面向特征继承性增减的在线分类算法，通过结合在线被动-主动算法和结构风险最小化原则进行模型训练，并运用在线矩阵补全方法恢复原特征数据实现模型重用，但缺失特征补全的策略可能使特征冗余甚至引入噪声。

上述方法大多聚焦于对特征演化流的适应性学习问题研究上，随着对特征演化数据流研究的深入，Hou 等[52]意识到更实际的情况是原特征会不可预测地消失，从而导致一个不完整的重叠周期，因此提出具有不可预测特征演化的预

测，运用矩阵补全的方法来解决重叠时期的特征不完全的问题。由于特征演化的同时也可能导致数据分布变化，这将使特征映射或补全方法不再可靠，因此文献[53]提出特征空间和分布可进化流学习，为充分利用旧特征空间中的数据，关键在于弥合新旧特征空间与新特征空间之间的差距，针对这一问题提出一种新的差异度量方法——进化差异，利用横跨两个不同特征空间的差异度量来提高模型泛化能力。考虑到大多时间步后不提供标签，虽然流形正则化技术可以很好地解决标签缺失问题，但是需要一个缓冲区来存储部分以往数据来辅助学习，于是 Hou 等[47]提出存储适合特征演化流学习，利用流形正则化技术将 FESL 算法中的损失函数转换为“风险函数”，从而重新使用在线梯度下降进行计算。

在对存在特征演化的高维数据流进行分类时，则需权衡收敛速度和模型深度。为此，Lian 等[54]提出新的在线深度学习范式，其核心思想是用变分逼近发现新旧特征之间的共享潜在子空间，由此实现在线分类遗憾快速最小化和特征级关系的实施逼近。针对特征数量随时间递增的数据流，Zhang 等[55]提出梯形数据流的在线学习方法，该方法利用在线被动-主动算法增量训练分类器，仅考虑到不断有新特征加入而忽略特征会消失的可能。为进一步放松限制，He 等[56]提出生成式学习任意数据流学习方法，假设每一维特征都有对应的分类器，每当有新的特征时，借助图的相关技术存储之前的信息来把当前样本映射到通用特征。在处理高维数据时，该方法会带来较高的计算和存储开销。文献[57]中基于多样化特征空间的在线学习方法，将特征分为存留特征、共享特征和新特征。存留特征指的是仅上一轮分类器拥有的特征，共享特征指当前样本和上一轮分类器共同拥有的特征，而新特征指只有当前样本拥有的特征。根据结构风险最小化原则进行模型更新，预测时则需要将当前模型和当前样本映射到相同维度的子空间做预测。

1.3 论文研究内容

研究现状和发展趋势表明，面向特征演化数据流的分类任务仍存在以下重要挑战：

（1）特征演化数据流的自适应学习问题：传统数据流分类算法只能处理固定特征空间的数据流，而动态变化的特征空间会使数据与模型特征维度不一

致，这将造成分类性能下降甚至失效。因此，如何在动态特征空间下挖掘与表示学习到的特征权重信息，在不同特征空间传递与继承这些信息，使模型仍能对新特征数据进行预测，是特征演化数据流分类工作的重点研究内容之一。

（2）提高分类模型的稳定性问题：在特征演化过程中数据的特征空间不断变化，为保证分类过程中需构建不同维度的分类模型，如何采取有效的集成策略，通过选择与组合这些分类模型，在避免资源浪费的同时辅助提升新分类器的性能亟待解决。

（3）特征演化数据流中的类不平衡问题：传统的基于采样策略或基于代价敏感方法的不平衡学习方法都需要完整的特征空间，而不适用于特征演化数据流的在线分类任务。因此，如何改进代价敏感学习方法，根据当前实例类别标签与不平衡率重新定义代价敏感因子，不断调整实例的类别权重以降低对类不平衡数据流的误分类代价，提高分类模型的泛化能力同样是特征演化数据流学习面临的重要问题之一。

针对上述问题，本课题致力于特征演化数据流的分类方法研究，结合特征映射与划分、集成学习、代价敏感等技术，设计高效稳定的动态分类算法，并进一步在具有类不平衡和特征演化的场景中进行适应性学习。本文具体工作总结如下：

（1）针对特征演化自适应问题，提出基于映射策略的特征演化数据流分类方法。该方法首先采用模糊隶属度来提高数据的准确性和可靠性；然后利用孪生支持向量机算法对已有特征的实例进行分类训练；随着新特征的出现，通过局部加权线性回归算法来拟合异构特征空间之间的映射矩阵，将已训练良好的模型映射至新特征空间，从而有效地适应新特征数据的学习。实验结果表明，该方法能够适应特征演化流的学习，同时具有良好的抗噪声能力。

（2）针对分类模型的稳定性问题，提出基于集成学习的特征演化数据流分类方法。该方法首先通过引入模糊隶属度函数并结合增量孪生支持向量机模型，鲁棒地训练与更新分类器；当出现新特征时重新训练新分类器，同时利用对应阶段收集的共现数据学习新旧特征之间的映射函数，将已训练好的旧分类器投影至新特征空间继续更新；最后，提出新的基分类器权重更新策略，合并新旧分类器进行集成预测。大量实验验证该方法提高对特征演化数据流的分类

准确度与稳定性。

（3）针对数据流中的类不平衡问题，提出具有类不平衡的特征演化数据流在线学习方法。该方法研究特征演化方式更具随机性的数据流，同时处理数据流中类不平衡和概念漂移问题。首先将实例特征空间划分为共享特征空间和新增特征空间，并将分类器分别投影至对应特征空间，结合在线被动-主动算法分别训练不同特征空间下的分类器；然后将代价敏感指标最小化问题融入模型在线优化目标函数中，根据不平衡率定义新的代价敏感因子，动态调整类别权重以解决类不平衡问题；最后为提高分类器泛化性能，利用变异系数筛选出重要特征，从而对分类器稀疏截断处理。大量仿真实验结果表明，所提方法能有效处理特征演化流中的类不平衡和概念漂移问题，具有良好的泛化能力。

1.4 论文组织结构

全文由六个章节构成，各章节具体内容如下：

第 1 章，主要从本文研究工作的背景与意义、国内外研究现状进行介绍，分析目前面临的重要研究问题，并针对性提出解决方案形成本文研究内容。

第 2 章，对数据流、类不平衡以及不同类型的特征演化问题进行形式化定义，针对增量型、渐变型、突变型和混合型特征演化数据流的分类方法进行综述。

第 3 章，提出基于映射策略的特征演化数据流分类方法。首先分析问题场景，研究解决方案；然后设计算法框架，并展开介绍模型训练更新与异构特征映射的具体方法；最后介绍实验设计方案与结果分析。

第 4 章，提出基于集成学习的特征演化数据流分类方法。所提方法在上一章研究工作的基础上加以改进，通过结合集成学习与增量学习策略提高分类模型的稳定性与准确率，最后通过对比实验验证所提方法的有效性。

第 5 章，提出类不平衡的特征演化数据流在线分类方法。首先对特征演化形式更复杂的数据流场景进行分析，同时考虑数据类别分布不平衡的问题，构建算法框架并对各个子模块进行详细介绍，最后通过对实验设计与结果进行分析和讨论。

第 6 章，对论文研究工作进行总结，分析讨论当前研究工作的优势与不足，并进一步探讨未来研究方向。

第2章 背景知识

本章主要阐述课题相关背景知识与方法，首先介绍数据流、类不平衡以及特征演化的形式化定义，并根据不同演化形式对特征演化问题进行细分，展开介绍增量型、渐变型、突变型以及混合型特征演化，并针对四类特征演化问题的对应数据流分类方法展开介绍。

2.1 数据流

数据流是基于时间维度顺序到达的样本序列，可定义为 $\mathrm{Ds}=\{(\boldsymbol{x}_t, y_t)\,|\,t=1,2,\ldots,\mathrm{T}\}$，其中 t 表示时间，$\boldsymbol{x}_t=\{f_1, f_2,\ldots, f_{d_t}\}\in\mathbb{R}^{d_t}$ 是一个 d_t 维的向量，代表样本的特征信息；类别标签 $y_t\in Y=\{y_1, y_2,\ldots, y_n\}$，标签空间 Y 是由 n 个类标签构成的集合。数据流在样本产生的过程中往往存在概念漂移现象[58]，这是由样本产生固有的联合概率分布函数发生变化所引起的，形式化定义如下。

定义 2.1（概念漂移）若数据流样本在产生过程中服从某种联合概率分布 $P(\boldsymbol{x}, y)$，存在两个不同的时间步 $t=i$ 和 $t=j$，有 $P_i(\boldsymbol{x}, y)\neq P_j(\boldsymbol{x}, y)$ 成立，则称数据流中存在概念漂移。

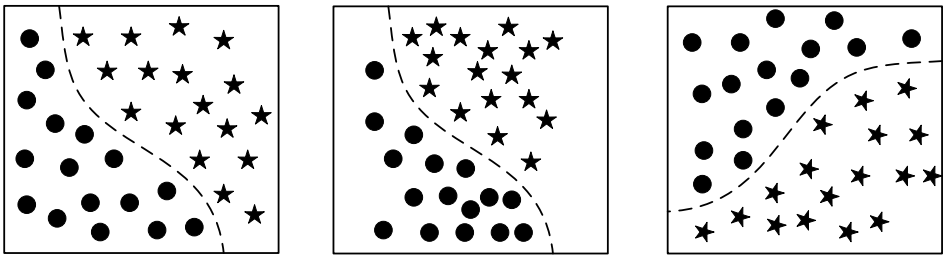
对联合概率分布进行分解可知，$P(\boldsymbol{x}, y)$ 由边缘概率 $P(\boldsymbol{x})$、后验概率 $\mathrm{P}(y\,|\,\boldsymbol{x})$ 两部分决定，即 $P(\boldsymbol{x}, y)=P(y\,|\,\boldsymbol{x})P(\boldsymbol{x})$。因此根据发生变化的原因，可将概念漂移分为三种类型[59]：

（1）若 $P(\boldsymbol{x}_{t-1})\neq P(\boldsymbol{x}_t)$，则称为虚拟概念漂移（Virtual Concept Drift），其数据分布如图 2-1（b）所示，其中虚线所表示的分界面不发生变化。该类概念漂移通常假设特征空间本身不变，而样本分布发生变化。

（2）若 $P(y\,|\,\boldsymbol{x}_{t-1})\neq P(y\,|\,\boldsymbol{x}_t)$，则称为真实概念漂移（Real Concept Drift），其数据分布如图 2-1（c）所示，可看出虚线所表示的分界面漂移前后产生明显改变。后验概率发生变化会对分类器决策边界造成影响，导致学习器分类性能下降。

（3）若 $P(\boldsymbol{x}_{t-1})\neq P(\boldsymbol{x}_t)$ 且 $P(y\,|\,\boldsymbol{x}_{t-1})\neq P(y\,|\,\boldsymbol{x}_t)$，则称混合型概念漂移，这要

求学习器需要从两方面同时处理分布的变化。



(a)漂移前数据分布　　(b)虚拟概念漂移　　(c)真实概念漂移

图 2-1 虚拟概念漂移与真实概念漂移

Fig. 2-1 Virtual concept drift and real concept drift

2.2 类不平衡

类不平衡问题广泛存在于数据流中，对于二分类任务，少数类（正类）实例的数量远小于多数类（负类）实例的数量，但两种类别实例的误分类代价是不相同的。传统数据流分类算法倾向于保证分类模型的整体准确率，而忽视少数类的分类精度。基于此，经典的分类器性能评价指标不再适用，需选取合适的评估指标更全面地衡量分类模型性能。

首先，需要引入如表 2-1 所示的混淆矩阵，便于介绍本文采用的相关评估指标。其中，TP 表示正确预测为正的样本的数量，FP 表示错误预测为正的样本的数量，TN 表示正确预测为负类样本的数量，FN 表示错误预测为负类样本的数量，所有实际为正类的样本数量合计为 $Num_p = TP + FN$ ，所有实际为负类样本数量合计为 $Num_n = FP + TN$ 。

表 2-1 混淆矩阵

Table 2-1 Confusion Matrix

实际类别	预测类别		合计数量
	正类	负类	
正类	TP（真正类）	FN（假负类）	Num_p
负类	FP（假正类）	TN（真负类）	Num_n

数据流分类任务中，通常使用准确率（Accuracy，ACC）评价模型性能，它能够反映模型在两个类别上整体预测精度，但面对类不平衡问题时，只使用准确率进行模型评价并不理想。几何均值 G-mean 是对召回率与特异度的几何平均值，同时考虑对不平衡数据流中正、负类样本分类正确率，是不平衡数据流分类模型的有效评价指标。马修斯相关系数（Matthews Correlation Coefficient，MCC）能够反映实际与预测结果之间的相关性，综合考虑混淆矩

阵中的 TP、TN、FP、FN，具有较高的均衡性，其取值范围是[-1,1]，越接近 1 则表明分类器分类结果越好，反之说明分类器性能越差，MCC=0 说明分类器在进行随机分类。上述评价指标具体定义如下：

（1）分类准确率 $ACC = \frac{TP + TN}{TP + FP + FN + TN}$

（2）召回率 $\mathrm{Re}call = \frac{TP}{TP + FN}$

（3）特异度 $Specificity = \frac{TN}{TN + FP}$

（4）几何均值 $G - mean = \sqrt{\mathrm{Re}call \times Specificity}$

（5）马修斯相关系数 $MCC = \frac{TP \times TN - FN \times FP}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$

2.3 特征演化

定义 2.2（特征演化）在数据流 Ds 中，存在两个相邻时间步的数据样本 $\boldsymbol{x}_{t-1}$ 和 $\boldsymbol{x}_t$，若其特征空间不同，即特征维度 $d_{t-1} \neq d_t$ 或特征分布 $P(\boldsymbol{x}_{t-1}) \neq P(\boldsymbol{x}_t)$ 成立，则称数据流中发生特征演化。

现实场景不同，特征演化的方式呈现多种类型，因此本章根据原有特征空间的保留情况、新特征空间的出现方式对特征演化场景分为四种：增量型特征演化、渐变型特征演化、突变型特征演化和混合型特征演化。

2.3.1 增量型特征演化

在增量型特征演化流中，随着数据流样本持续到达，新的特征出现在样本中导致特征维度持续增长。增量型特征演化如图 2-2 所示。图中实例 $\boldsymbol{x}_{t-1}$ 和 $\boldsymbol{x}_t$ 分别由特征集 $\{f_1, f_2, \ldots, f_{d_1}, \ldots, f_{d_{t-1}}\} \in \mathbb{R}^{d_{t-1}}$ 和 $\{f_1, f_2, \ldots, f_{d_1}, \ldots f_{d_{t-1}}, f_{d_t}\} \in \mathbb{R}^{d_t}$ 所描述，显然 $\boldsymbol{x}_{t-1}$ 的特征空间维度小于 $\boldsymbol{x}_t$，此时发生增量型特征演化。

因此对于任意时刻 t，当前数据特征维度 d_t 始终不小于上一时刻的数据特征维度 d_{t-1}，对增量型特征演化的定义如式(2-1)所示：

$$\forall t : d_{t-1} \leq d_t \tag{2-1}$$

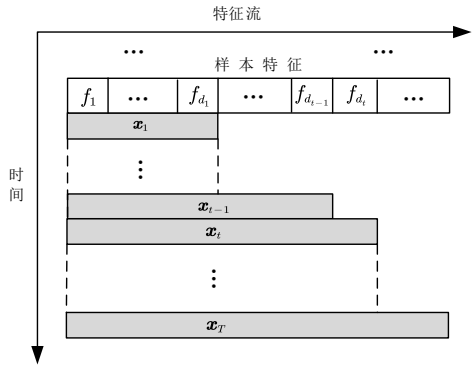


图 2-2 增量型特征演化

Fig. 2-2 Incremental feature evolution

2.3.2 渐变型特征演化

在渐变型特征演化流中，特征空间随着时间推移发生减少和增加变化，即原来的部分特征消失，另一部分原特征作为存活特征被保留下来，与新加入的特征共同描述新到达的数据，渐变型特征演化如图 2-3 所示。

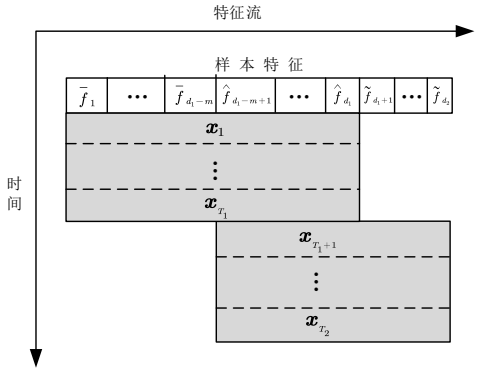


图 2-3 渐变型特征演化

Fig. 2-3 Gradual feature evolution

图 2-3 中，在一个演化周期内（t=1,2,…,T_2）存在三个特征子集：消失特征集 $S_v=\{\bar{f}_1,\bar{f}_2,\ldots,\bar{f}_{d_1-m}\}\in\mathbb{R}^{d_1-m}$、存活特征集 $S_e=\{\hat{f}_{d_1-m+1},\hat{f}_{d_1-m+2},\ldots,\hat{f}_{d_1}\}\in\mathbb{R}^{m}$ 和新增特征集 $S_n=\{\tilde{f}_{d_1+1},\tilde{f}_{d_1+2},\ldots,\tilde{f}_{d_2}\}\in\mathbb{R}^{d_2-d_1}$，实例 $\boldsymbol{x}_{T_1}$ 的特征空间为 $[S_v,S_e]$，实例 $\boldsymbol{x}_{T_1+1}$ 的特征空间为 $[S_e,S_n]$，先后到达两个实例的特征空间包含相同的特征子集 S_e，即旧特征空间包含 d_1-m 个消失特征和 m 个存活特征，这些存活特征与 d_2-d_1 个新增特征共存于新特征空间，渐变型特征演化中特征空间 S_t 与特征维

度 d_t 随时间变化特点如式(2-2)所示：

$$S_t=\begin{cases}[S_v,S_e]\in\mathbb{R}^{d_1} & ,t=1,2,\ldots,T_1\\ [S_e,S_n]\in\mathbb{R}^{d_2-d_1+m} & ,t=T_1+1,\ldots,T_2\end{cases} \tag{2-2}$$

2.3.3 突变型特征演化

在突变型特征演化流中，某一时刻新特征空间同时出现、旧特征空间同时消失。与渐变型特征演化区别在于，突变型特征演化前后的特征空间完全不同，在特征空间发生变化的过程中有少量连续实例同时被两个特征空间共同描述。

突变性特征演化如图 2-4 所示，例如，一个演化周期 t=1,2,…,T_2 内，实例 $\boldsymbol{x}_{T_1}$ 由旧特征集 $S_1=\{\bar{f}_1,\bar{f}_2,\ldots,\bar{f}_{d_1}\}\in\mathbb{R}^{d_1}$ 所描述，在短暂的重叠时期 $t=T_1+1,\ldots,T_1+n$ 内到达的实例 $\boldsymbol{x}_{T_1+1},\boldsymbol{x}_{T_1+2},\ldots,\boldsymbol{x}_{T_1+n}$ 均由旧特征集 S_1 和新特征集 $S_2=\{\tilde{f}_{d_1+1},\tilde{f}_{d_1+2},\ldots,\tilde{f}_{d_2}\}\in\mathbb{R}^{d_2-d_1}$ 共同描述，之后观察到的实例 $\boldsymbol{x}_{T_1+n+1}$ 的特征空间只包含 S_1。其特征空间 S_t 与特征维度 d_t 随时间变化特点如式(2-3)所示：

$$\begin{aligned}S_t&=\begin{cases}S_1\in\mathbb{R}^{d_1} & ,t=1,2,\ldots,T_1\\ [S_1,S_2]\in\mathbb{R}^{d_2} & ,t=T_1+1,\ldots,T_1+n\\ S_2\in\mathbb{R}^{d_2-d_1} & ,t=T_1+n+1,\ldots,T_2\end{cases}\\ d_t&=\begin{cases}d_1 & ,t=1,2,\ldots,T_1\\ d_2 & ,t=T_1+1,\ldots,T_1+n\\ d_2-d_1 & ,t=T_1+n+1,\ldots,T_2\end{cases}\end{aligned} \tag{2-3}$$

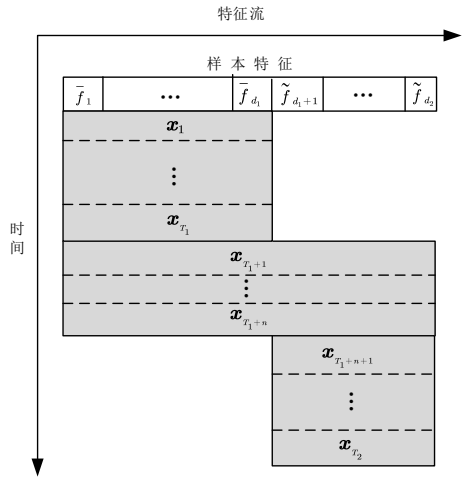


图 2-4 突变型特征演化

Fig. 2-4 Sudden feature evolution

2.3.4 混合型特征演化

在混合型特征演化流中，任意两个样本都可能包含不同的特征，其特征变化情况复杂多变。其典型变化如图 2-5 所示，例如实例 $\boldsymbol{x}_1$ 的特征空间为 $\{f_1, f_2, ..., f_{d_1}\} \in \mathbb{R}^{d_1}$，实例 $\boldsymbol{x}_2$ 的特征空间对应为 $\{f_1, \ldots, f_{d_1}, f_{d_{1-1}}, f_{d_t}\} \in \mathbb{R}^{d_1+2}$，两实例的特征空间可能包含相同特征，也可能包含不同特征，数据流的特征空间可能发生不可预测的演化。因此对于任意两个不同的时间步 $t=i$ 和 $t=j$，观察到的数据特征维度 d_i 与 d_j 不同，其定义如式(2-4)所示：

$$\forall i, j (i \neq j) : \exists d_i \neq d_j \tag{2-4}$$

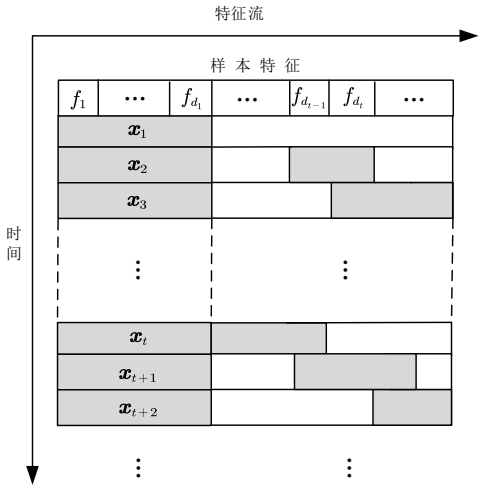


图 2-5 混合型特征演化

Fig. 2-5 Mixed feature evolution

值得说明的是，虽然图 2-5 中所有实例均包含相同特征子集 $\{f_1, f_2, ..., f_{d_1}\} \in \mathbb{R}^{d_1}$，但这只是诸多变化形式的一种，理论上对特征的演化方式不做任何假设。随着研究的逐步深入，特征演化的类型不局限于以上四种，日益复杂的特征演化问题可能同时包含上述两种及以上的演化形式。此外，上述特征演化问题，在不同的文献中描述有所不同，如在文献[55]中认为“增量型特征演化数据流”是新的双流数据（Doubly-Streaming）问题，并将这种双流数据称为梯形数据流（Trapezoidal Data Streams）；Hou 等[47]将此问题定义为特征可进化流（Feature Evolvable Streams），并在文献[9]中描述相关工作为多特征空间学习（Multiple Feature Sets Learning），文中将其定义为“突变型特征演化数据流”；He 等[60]将特征任意变化的情况称为反复无常的数据流（Capricious Data

Streams），文中将其定义为“混合型特征演化数据流”。

2.4 特征演化数据流分类方法

本节以四类特征演化问题的学习为主线，分别对面向增量型、渐变型、突变型和混合型特征演化流的分类方法进行总结与讨论。尽管这些方法都用来处理特征演化数据流，但是因为特征空间变化形式不同，数据流分类模型遇到的挑战以及对应的解决方案也不同，本文侧重于突变型与混合型特征演化数据流分类方法的研究。

2.4.1 增量型特征演化数据流分类

学习增量型特征演化流的关键是采取合适的模型更新策略，分类模型应具备良好的可扩展性，来适应递增的特征空间，同时为避免出现维度灾难，需要采取适当的剪枝或稀疏策略以限制模型维度的不断增加。

早期研究主要与增量学习相关，Guan 等[61]认为在现有的神经网络中必须考虑并添加一组新的输入属性的情况，提出输入属性增量学习算法（Incremental Learning in terms of Input Attributes，ILIA），该算法包含三个阶段：在阶段 1 中，现有的网络被保留为旧的子网络而不是丢弃浪费，其中呈现所有的训练、验证和测试模式，每个模式上每个输出单元的加权输入之和被计算并存储在数组中；在阶段 2 中，需要构建并训练新的子网络；在阶段 3 中，需要确定新的传入输入属性是否与输出相关并与现有的输入属性是否一致。如果新的输入属性与输出相关，且与已有的输入属性一致，则将新的子网络与旧的子网络合并，形成适应特征变化问题的新神经网络；否则，拒绝新的传入输入属性并保留旧的子网络。ILIA 算法考虑神经网络在动态环境中输入属性增量出现，采用集成新旧两个子分类器的方法来适应特征维度增加，但只假设一组新的输入属性被引入当前系统。

Zhang 等[55]将在线学习与数据流的特征选择相结合，提出一种新的基于流特征的在线学习算法（Online Learning with Streaming Features algorithm，OL_{SF} for short）及其两个变体，从而对增量型特征演化流进行持续学习，其中分类器按照在线学习中使用的 PA 算法（Passive-Aggressive）对数据流进行学习，对于接收到一个实例 $\boldsymbol{x}_t$，利用上一轮的分类器 $\boldsymbol{w}_t$ 进行预测 $\hat{y}_t = \mathrm{sign}\left(\boldsymbol{w}_t \cdot \Pi_{w_t}, \boldsymbol{x}_t\right)$，获

取真实标签 y_t 后，更新线性分类器 $\bar{\boldsymbol{w}}_{t+1}$ 如式(2-5)：

$$\bar{\boldsymbol{w}}_{t+1}=[\boldsymbol{w}_t+\tau_t y_t H_{\boldsymbol{w}_t}\boldsymbol{x}_t,\tau_t y_t\prod_{-\boldsymbol{w}_t}\boldsymbol{x}_t] \tag{2-5}$$

其中，τ_t 为学习率，$\neg\boldsymbol{w}_t$ 不包含在原分类器 $\boldsymbol{w}_t$ 中的向量。最后利用投影截断技术降低特征维度，但其不足在于每轮都需要获取单个实例的真实标签，只能进行二分类任务并且没有考虑到特征之间可能存在的非线性映射关系。

针对 OL_{SF} 算法的线性决策边界学习的局限性，Beyazit 等[62]提出一种具有快捷连接的单隐层前馈神经网络算法，该算法学习数据流是否需要使用非线性变换映射，以加快学习收敛速度，提出的模型中添加增长和修剪机制，来调整模型的复杂性以适应不断变化的特征空间，当一个特征维度 d_t 更高的训练示例 $\boldsymbol{x}_t$ 到来时，网络以完全连接的方式分配 $d_t-d_t^i$ 个输入和隐藏层节点并随机初始化权值，分配新的输入节点拓展网络以适应新增特征，新的隐藏层节点增强模型性能来满足不断增加的分类复杂度。该方法中的层被调整以处理增长的特征空间，并且应用基于权重的修剪过程以保持较低的运行时间，相比于 OL_{SF} 算法，SLFN-S 收敛速度更快且得到更为简单的决策边界。

文献[63]提出动态快速决策树方法（the Dynamic Fast Decision Tree，DFDT），通过重组和修建 Hoeffding 树来允许增加特征空间。算法首先构建 Hoeffding 树，然后检查到达的数据实例是否具有新特征，如果该实例的特征空间 $F_i\notin F$ 则将新特征添加至特征列表 F，若发现样本数量积累超过重构区间最小值 $m_{\min}$ 则进行重构树，随后检查是否需要删除不必要的节点来修剪树，从而提高模型的灵活度以更好的适应高动态的特征演化问题。

为处理标签稀少的增量型特征演化流，文献[64]提出一种新的标签稀缺的增量特征空间学习方法及其两个变体，每一轮中接收一个实例 $\boldsymbol{x}_t$ 并计算其预测置信度 f_t，利用基于边缘的在线主动学习来选择有价值的实例进行标记，从而在最小监督下构建性能良好的分类器，然后在共享特征空间与增强特征空间上，结合 OL_{SF} 算法中的在线 PA 更新策略和边缘值最大化原则，来动态更新分类器；最后利用投影截断技术来建立一个稀疏但有效的模型。

Liu 等[65]同样将主动查询策略与 PA 更新策略相结合，在 PA 主动算法 PAA（PA Active）的基础上提出一组新的在线主动算法：针对增量型特征演化流的

算法 PAA_{TS} 和多类 PAA_{TS}（multiclass PAA_{TS}，$MPAA_{TS}$）及其变体来解决在线二分类和多分类任务。每一轮中接收一个实例 $\boldsymbol{x}_t$ 并计算其预测置信度 f_t，与 FLLS 算法思路相似，通过实例调节的随机查询策略来考虑是否查询该实例的真值标签，若不查询则不更新学习器，否则根据查询到的真值标签计算瞬时损失后，利用 OL_{SF} 算法中的 PA 更新策略来更新分类模型。但 OL_{SF} 算法和 FLLS 算法不同的是，PAA_{TS} 算法有意省略处理特征稀疏性的过程，方便在考虑所有特征的情况下，探究主动查询比率对学习器性能的影响。

综上，对增量特征演化流的经典研究 OL_{SF} 算法[55]搭建合理的算法框架，SLFN-S[62]和 DFDT 算法[63]改进学习模型，进一步提高模型收敛能力与分类性能。PAA_{TS} 算法[65]和 FLLS 算法[64]在 OL_{SF} 算法的基础上结合在线主动学习技术，来解决增量特征演化流中标签稀缺的问题。

2.4.2 渐变型特征演化数据流分类

在特征空间不断变化的过程中，不仅存在特征维度不断增加的情况，还有可能部分特征消失，而另一部分特征被保留下来与新特征共存，面对这种渐变特征演化流的分类问题，关键在于如何计算特征之间的相似度，实现将消失特征包含的信息压缩至存活特征中。

Hou 等[50]具有增量和递减特征的一遍式学习方法（One-Pass Incremental and Decremental learning approach，OPID），假设旧特征部分消失时，仍有一个子集会存活下来，并且合并新特征集继续存在，该算法分为压缩阶段（C 阶段）和扩展阶段（E 阶段）。在 C 阶段，为将特征消失的数据所表征的判别信息压缩为特征存在的函数，针对具有不同特征的数据集训练的分类器上添加一致性约束，是对传统正则化最小二乘分类器的扩展，如式(2-6)所示：

$$\begin{aligned}&\min_{\tilde{\boldsymbol{w}},\boldsymbol{w}^{(s)}}\sum_{i=1}^{T_1}\left\|\left\langle\tilde{\boldsymbol{w}},\tilde{\boldsymbol{x}}_i\right\rangle-y_i\right\|^2+\sum_{i=1}^{T_1}\left\|\left\langle\boldsymbol{w}^{(s)},\boldsymbol{x}_i^{(s)}\right\rangle-y_i\right\|^2\\&+\lambda\sum_{i=1}^{T_1}\left\|\left\langle\tilde{\boldsymbol{w}},\tilde{\boldsymbol{x}}_i\right\rangle-\left\langle\boldsymbol{w}^{(s)},\boldsymbol{x}_i^{(s)}\right\rangle\right\|^2+\rho\left(\left\|\hat{\boldsymbol{w}}\right\|^2+\left\|\boldsymbol{w}^{(s)}\right\|^2\right)\end{aligned}\tag{2-6}$$

其中，$\tilde{\boldsymbol{w}}$ 和 $\boldsymbol{w}^{(s)}$ 分别是定义在所有特征和存活特征上的分类器系数，$\lambda>0$ 是确定一致性约束重要性的参数，$\rho>0$ 表示正则化的参数。在 E 阶段，利用集成方法统一两个分类器，如式(2-7)：

$$\min_{\tilde{h}^{(s)},\bar{h}} \boldsymbol{w}_1 \tilde{l}^{(s)}(\tilde{h}^{(s)}(\tilde{\boldsymbol{z}}_{T_1+1}^{(s)}), y_{T_1+1}) + \boldsymbol{w}_2 \bar{l}(\bar{h}^{(s)}(\bar{\boldsymbol{z}}_{T_1+1}), y_{T_1+1}) \tag{2-7}$$

其中，$\boldsymbol{w}_1 \geq 0, \boldsymbol{w}_2 \geq 0, \boldsymbol{w}_1 + \boldsymbol{w}_2 = 1$是平衡两个分类器的权重，$\tilde{l}^{(s)}$和$\bar{l}$是定义在$\tilde{\boldsymbol{z}}_{T_1+1}^{(s)}$和$\bar{\boldsymbol{z}}_{T_1+1}$上的两个代理损失函数。

Ye 等[66]提出通过异构预测器映射进行校正的鲁棒建模框架，实现在新特征空间中有效利用训练良好的旧模型，通过最优传输语义映射来弥合异构特征之间的不一致性。同时提出一种新的特征元信息编码（Encode Meta InformaTion of features，EMIT）策略，通过元编码对齐特征后，实现利用少量的新特征空间下的训练数据，从异构特征空间提炼新模型。因此，整个框架能够在动态环境中处理变化特征，但不足之处在于它使用的是批处理模式。

Dong 等[67]从度量学习出发研究计算特征相似度，提出一个新的在线进化度量学习方法，该方法包含转换阶段（T 阶段）和集成阶段（I 阶段）。在 T 阶段中，为从消失特征中提取重要信息，通过在存活特征和所有特征（包含消失和存活特征）上合并平滑的 Wasserstein 距离度量，来表征异构样本之间的相似性关系；I 阶段中，从 T 阶段继承存活特征的度量性能并拓展到新特征空间中。

Peng 等[68]考虑一个更为实际的数据流场景——同时具有概念漂移和特征演化的数据流，并提出一种挖掘具有演化特征的概念漂移数据流的通用框架 FEMC，并进一步提出 FEMC-kmeans 和 FEMC-knn 来解决特征演化流的聚类与分类问题。该算法的基本思想是动态维护一批具有不同特征空间的微簇：发生特征演化时，通过学习生存特征的权重向量来保存消失特征的信息；对于新出现的特征，通过新增维度将生存特征空间拓展到新的特征空间中；对于之后每轮传入的新实例，便可通过 kmeans 算法来进行聚类，或者采用 knn 进行分类。

上述研究工作致力于从消失特征中提取重要信息，并将其保存至存活特征中。为实现信息传递，OPID 算法[50]和 EML 算法[67]研究如何合理计算异构特征之间的相似度，以实现增量特征空间下的模型重用。REFORM[66]与 FEMC[68]分别使用特征元信息和微簇来感知特征的演化，同时 FEMC 算法可对特征演化流进行无监督聚类，拓展了该领域的研究成果。

2.4.3 突变型特征演化数据流分类

突变型特征演化流包含两个完全不同的特征空间，解决突变型特征演化问题首先要考虑如何合理构建新旧特征空间的映射来实现模型的重用与更新。

Hou 等[47]首次提出研究突变型特征演化流，将这种存在新特征出现、旧特征消失的数据流学习构建为 FESL 学习范式，该范式利用在线梯度下降算法对模型进行更新，如式(2-8)所示：

$$\boldsymbol{w}_{i,t+1}=\Pi_{\Omega_i}(\boldsymbol{w}_{i,t}-\tau_t\nabla\ell(\boldsymbol{w}_{i,t}^{\top}\boldsymbol{x}_t^{S_i},y_t)) \tag{2-8}$$

其中投影 $\Pi_{\Omega_i}(b)=\arg\min_{\alpha\in\Omega_i}\|a-b\|, i=1,2$，$\tau_t$ 为不断变化的步长。当特征发生演化时，在短暂的重叠期内，通过使用式(2-9)所示的最小二乘法来学习从新特征到旧特征的线性投影矩阵 $\boldsymbol{M}$，恢复消失的特征来重用原来训练良好的模型：

$$\min_{\boldsymbol{M}\in\mathbb{R}^{d_2\times d_1}}\sum_{t=T_1-B+1}^{T_1}\frac{1}{2}\|\boldsymbol{x}_t^{S_1}-\boldsymbol{M}^{\top}\boldsymbol{x}_t^{S_2}\|_2^2 \tag{2-9}$$

在此基础上设计两种集成方案：一种是将两个模型的预测结果结合起来，另一种是动态选择最佳单个预测，以此来提高分类性能。

随着对特征演化数据流研究的深入，Hou 等[52]意识到这种特征空间变化的假设通常不成立，而更实际的假设是不同的特征会不可预测地消失，那么将出现一个不完整的重叠周期，因此分析并提出新的模式——具有不可预测特征演化的预测（Prediction with Unpredictable Feature Evolution，PUFE），其中旧特征不可预测地消失而新特征会同时出现。为解决不可预测的特征缺失问题，算法利用名为 Matrix Chernoff[69]的技术进行对重叠期的不完整矩阵补全，得到补全后的完整重叠期后，可利用集成的方法来同时利用新旧两个模型以提高分类性能。

在进一步研究中，考虑到大多时间步后不提供标签，虽然流形正则化技术可以很好地解决标签缺失问题，但是需要一个缓冲区来存储部分以往数据来辅助学习，于是 Hou 等[70]提出存储适合特征演化流学习（Storage-Fit Feature-Evolvable streaming Learning，SF^{2}EL），针对缺少标签的数据流，利用流形正则化技术将 FESL 算法中的损失函数转换为“风险函数”，从而重新使用在线梯度下降进行计算。同时，算法采取一种名为储层采样[71]的缓冲策略，灵活设置固定缓冲区以接收历史样本。

由于特征演化的同时数据分布也可能变化，文献[37]对 FESL 进行拓展解决问题，文中将这种学习问题定义为特征空间和分布可进化流学习（Feature space and Distribution Evolvable Stream Learning，FDESL）。为充分利用旧特征空间中

的数据，关键在于弥合新旧特征空间与新特征空间之间的差距，针对这一问题提出一种新的差异度量方法——进化差异，并设计演化差异最小化 EDM 算法，利用横跨两个不同特征空间的差异度量来提高模型泛化能力。

使用在线梯度下降法进行模型更新时，仅会在分类错误时才会更新模型，并不能很好地提高分类性能，Liu 等[48]提出一种基于被动攻击更新策略的特征演化学习算法（Passive-Aggressive Learning with Feature Evolvable Streams，PAFE），其中引入在线 PA 算法更新分类模型，如式(2-10)：

$$\boldsymbol{w}_{i,t+1}=\boldsymbol{w}_{i,t}+\tau_{i,t}y_t\boldsymbol{x}_t^{S_i},i=1,2 \tag{2-10}$$

其中，$\tau_{i,t}=\min\{C,\ell(f_{i,t},y_t)/\|\boldsymbol{x}_i^{S_i}\|^2\}$。与 FESL[47]不同的是，PAFE 算法[48]学习新特征与旧特征之间的相互映射，不仅恢复旧特征空间以重用旧模型，而且利用旧模型对新模型进行初始化，随后对两个基模型进行组合预测或单个最优预测，来加快学习器的收敛速度并提高分类效果。然而 PAFE 算法仅仅使用一阶信息来进行预测将忽略参数的更新方向，为进一步提升分类器性能，Liu 等[49]进一步拓展研究，提出特征加权的置信-加权学习算法（Confidence-Weighted Learning for Feature Evolution，CWFE），该算法通过引入权重的自适应正则化 AROW 方法更新已学习的模型和置信矩阵，实现提高预测性能与加快收敛速度。

当进入大数据的在线深度学习环境下，对存在特征演化的高维数据流的分类则需要权衡收敛速度和模型深度，Lian 等[54]提出新的在线深度学习范式（Online Learning Deeply from Data of Double Streams，OLD^{3}S），其核心思想是用变分逼近发现新旧特征之间的共享潜在子空间，模型学习目标框架定义如式(2-11)所示：

$$\min\sum_{\mathrm{HBP}}^{T_1}\left[\sum_{t_1,t_b}(\mathcal{L}_{\mathrm{VI}}(\phi)+\mathcal{L}_{\mathrm{REC}}(\phi))+\sum_{t_b,t_2}\mathcal{L}_{\mathrm{CLF}}(f_t,\phi)\right] \tag{2-11}$$

其中，变分推理损失 $\mathcal{L}_{\mathrm{VI}}$ 与重构损失 $\mathcal{L}_{\mathrm{REC}}$ 共同决定共享潜在子空间的学习方式，分类损失 $\mathcal{L}_{\mathrm{CLF}}$ 表示如何集成新旧分类器以加快收敛速度。OLD^{3}S 作为一种完整的神经体系结构，其浅层类似一个简单的线性分类器，便于在初始轮中实现快速收敛；需要拟合更为复杂的特征映射关系时，则从浅层逐渐转入深层；获得重构映射后，模型可以很好地集成新旧分类器从而提高在线分类性能，由此实

现快速最小化在线分类遗憾和实时逼近特征级关系的需要。

针对突变型特征演化流的研究中，文献[47]明确问题解决的关键是学习不同维度特征之间的映射函数，随后 Hou 等[47]先后提出 PUFE 算法和 SF^{2}EL 算法，分别使用矩阵补全技术与流形正则化技术处理标签不足的情况，EDM 算法[72]用以解决特征空间与数据分布同时演化的情况，Liu 等[48]分别利用一阶与二阶信息提升在线学习模型性能，OLD^{3}S 算法[54]结合深度学习构建非线性分类器促进模型收敛。

2.4.4 混合型特征演化数据流分类

由于动态特征空间中存在新特征不断出现、旧特征无规律地消失，因此相关研究工作不对特征变化设置任何限制，其中大多算法选择构建通用特征空间的方法，从而被动适应特征空间任意变化的情况。

Beyazit 等[57]提出一种新的在线学习方法（Online Learning from Varying Features，OLVF），来应对特征在不同时间步上任意变化的情况。首先，对每轮输入的实例 $\boldsymbol{x}_t$，学习器将实例分类器和实例投影到当前共享特征空间上，利用特征空间分类器 $\bar{\boldsymbol{w}}_t$ 计算投影置信度，并自适应地根据实例分类器权重 $\boldsymbol{w}_t=[\boldsymbol{w}_t^e,\boldsymbol{w}_t^s,\boldsymbol{w}_t^n]$ 的各部分对目标函数的不同部分重新加权，其中 $\boldsymbol{w}_t^e$， $\boldsymbol{w}_t^s$， $\boldsymbol{w}_t^n$ 分别代表消失特征空间、存活特征空间以及新增特征空间；最后引入投影截断技术来裁减冗余或消失的特征。

He 等[60]提出生成式学习与随机数据流学习方法，每个到达的数据实例对应特征都不同，该算法在通过学习新旧特征空间之间的关系，用生成图模型构建通用特征空间，图中的每个顶点表示通用特征空间中的一个特征，每个顶点的邻接边表示该特征与其他特征之间的关系；最后在通用特征空间下训练更新学习器，该算法在增量型、突变型以及混合型特征演化流的学习中均有效可用。

为解决混合型特征演化流的不完全监督问题，文献[57]中提出一种新的学习范式（Online learning in Variable feature Spaces under Incomplete Supervision，OVSIS），并提出一种新的集成预测和稀疏截断的交替梯度下降算法（Alternating Gradient Descent with Ensemble prediction and Sparse truncation，AGDES），旨在特征任意变化下充分利用有标记实例更新分类器并加快收敛。首先，同样考虑构建通用特征空间，将输入的样本点投影到该特征空间上，实

现不同特征下样本点的距离测量；然后，利用无标记实例揭示底层数据的流形结构，并通过随机投影树（RP-Tree）来稀疏逼近其几何形状，以提升学习性能。

大多特征演化流的研究基于特征具有相同数据类型的假设，但是处理具有混合数据类型的特征演化流时则可能表现不佳。文献[57]从数据流类型的角度出发，拓展混合型特征演化流的学习，提出一种基于高斯关联的混合数据复杂联合分布模型，包含具有混合数据变量的可变特征空间在线学习算法（Online learning in Variable Feature spaces with Mixed data，OVFM），该算法没有使用线性模型来拟合映射关系，而是使用 copula 模型来拟合非线性映射，在连续变量编码的潜在正态空间中建模特征相关性，而不需要对任何特征的原始分布或数据族进行假设。但由于它将每个特征都视为关联组件，因此无法很好地拓展到高维空间。

此外，一些研究工作考虑在特征任意变化的混合型特征演化流中进行特征选择。Wu 等[73]提出 GF-CSF 算法框架，在特征选择之前采用潜在性因子分析（Latent Factor Analysis，LFA）[74,75]来实现特征预处理，实现从混合型特征演化流中进行选择特征。最近，You 等[76]提出的多源流特征选择方法（Multi-source Causal Feature Selection，MSFS），适用于各种多源流数据场景，包括多源流数据和多源固定数据、实例重叠且缺少特征的多源流数据、具有重叠实例和重叠特征的多源流数据场景。

综上可知，对于混合型特征演化流的研究工作中，OLVF 算法[57]、OVFM 算法[60]分别使用线性映射与非线性映射构建特征关系，GLSC 算法[60]、AGDES 算法[77]则对不同维度的特征进行泛化处理，通过构建通用特征空间实现模型的重用。这些算法最后需要添加投影截断技术来稀疏模型，而 GF-CSF 算法[73]和 MSFS 算法[76]则结合特征选择方法对不可预测的混合型特征演化流进行学习。

2.5 本章小结

面向特征演化的数据流分类挖掘研究正兴起，相关的理论知识和方法层出不穷。本章从三个方面对特征演化问题进行归纳总结：首先，给出相关概念的形式化定义，并重点阐述特征演化的特征及四种典型演化形式；然后，结合四种演化场景，综述当前现状和研究成果，为后续研究内容奠定理论基础。

第3章 基于映射策略的特征演化数据流分类方法

数据流分类任务中，研究人员通过挖掘数据流中蕴含的重要知识对分类模型进行迭代更新。而解决特征演化问题的关键是利用特征空间之间的相似性，最大程度地保留历史知识以辅助当前分类工作。

现有相关研究工作大多采用特征划分、特征映射以及矩阵补全等方法进行知识保留与传递。比如在学习增量型特征演化数据流时，Zhang 等[55]提出的基于流特征的在线学习 OL_{SF} 算法，该算法将数据特征划分为两部分：新增特征与历史特征，通过改进在线被动-主动算法，构建在两部分特征上分别更新的分类模型。OPID 算法[50]将数据特征划分为消失特征（Vanished Feature）、存活特征（Survived Feature）和增强特征（Augmented Feature），通过压缩-拓展的方式保留消失特征包含的判别信息以辅助分类器更新。基于特征划分的方法虽然能有效适应分类器与数据特征维度不一致的问题，但本质上对新特征下数据训练仍存在冷启动问题从而损害分类精度。FESL 算法[47]通过揭示新旧特征之间的关系矩阵，将新到样本映射回旧特征空间，利用现有分类器继续更新与预测。然而保证该方法有效性的关键在于学习到的关系矩阵需尽可能拟合两特征空间的映射关系，否则映射后数据存在偏差将导致预测性能下降。文献[51]中提出面向特征继承性增减的在线分类算法，通过运用在线矩阵补全方法恢复原特征数据实现模型重用，然而缺失特征的补全策略可能使特征冗余甚至引入噪声。同时考虑到学习环境的复杂性，数据采集过程中不可避免地容易将噪声数据引入特征演化数据流中，这将对数据挖掘模型的泛化能力产生负面影响[78]。因此，亟需构建鲁棒的特征演化数据流自适应分类模型。

基于此，本章提出一种基于映射策略的特征演化数据流分类方法。首先，引入模糊隶属度函数以提高给定数据的可靠性；然后基于局部权重的线性回归算法，利用共现数据构建新特征空间与已有特征空间之间的映射关系。最后，根据映射矩阵将历史模型投影至新特征空间继续学习，避免反复重新训练导致的资源浪费问题。

3.1 问题描述与分析

在特征演化数据流中，特征空间存在明显的阶段性变化，如图 3-1 所示：首先，在T_1阶段，收集大量具有现有特征的数据，其中在 B 阶段内存在部分共现数据，这些数据由现有特征和新特征共同描述；接着在T_2阶段，少量新到达的数据实例只具有新特性。值得注意的是，各阶段数据流特征保持相对稳定，各阶段特征变化规律总结如下：

①当$i=1\sim T_1-B$时，分类器观测到数据样本$\boldsymbol{x}_i^{(1)}\in\mathbb{R}^{d_1}$，所有样本均来自特征空间为$S_1=\left\{\bar{f}_1,\bar{f}_2,\ldots,\bar{f}_{d_1}\right\}\in\mathbb{R}^{d_1}$，其中$d_1$为特征空间$S_1$的维度。

②当 $i=T_1-B+1\sim T_1$时，分类器获取样本$\boldsymbol{x}_i^{(c)}=\left\{\left(\left[\boldsymbol{x}_i^{(1)},\boldsymbol{x}_i^{(2)}\right],y_i\right)\right\}_{T1-B+1}^{T1}$，其中$\boldsymbol{x}_i^{(1)}$和$\boldsymbol{x}_i^{(2)}$分别来自特征空间$S_1$和$S_2=\left\{\tilde{f}_1,\tilde{f}_2,\ldots,\tilde{f}_{d_2}\right\}\in\mathbb{R}^{d_2}$，其中$d_2$为特征空间$S_2$的维度。

③当 $i=T_1+1\sim T_2$时，分类器观测到数据样本$\boldsymbol{x}_i^{(2)}\in\mathbb{R}^{d_2}$，所有样本均来自特征空间$S_2$。

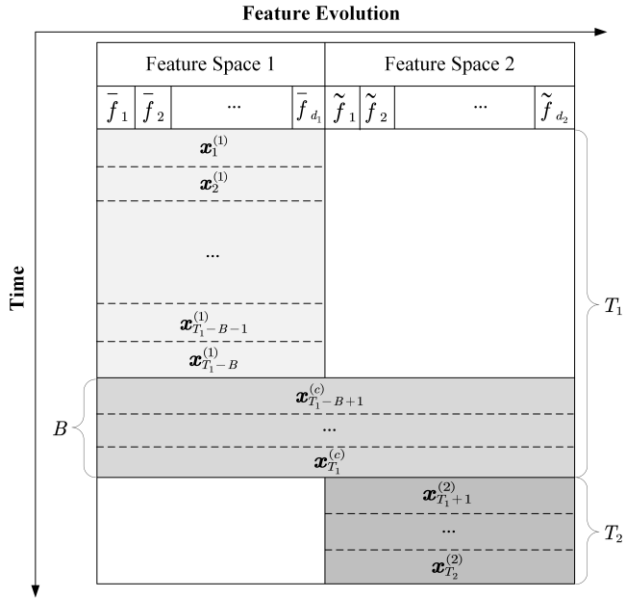


图 3-1 特征演化数据流

Fig. 3-1 Feature evolvable data stream

根据数据流特征演化规律可知，构建鲁棒的特征演化数据流分类模型亟需解决两个关键问题：（1）如何训练有效的分类模型并对新特征数据进行预测，并减轻噪声干扰；（2）如何跨异构特征空间重用历史模型，在保证分类精度的

同时避免资源浪费。

3.2 基于特征映射的数据流自适应分类模型

在上述讨论的基础上，提出一种新颖的学习特征演化数据流方法 FENSL，所提方法的框架和相关功能模块如图 3-2 所示。在功能模块 1 中，通过计算模糊隶属度值来区分支持向量和离群值，然后使用孪生支持向量机算法对 S_1 的数据进行模型训练。在功能模块 2 中，借助共现数据学习 S_1 和 S_2 之间的映射矩阵 M，使上一阶段训练的模型能够在新的特征空间中被重用。在功能模块 3 中，可利用该模型继续对 S_2 的新数据进行预测。

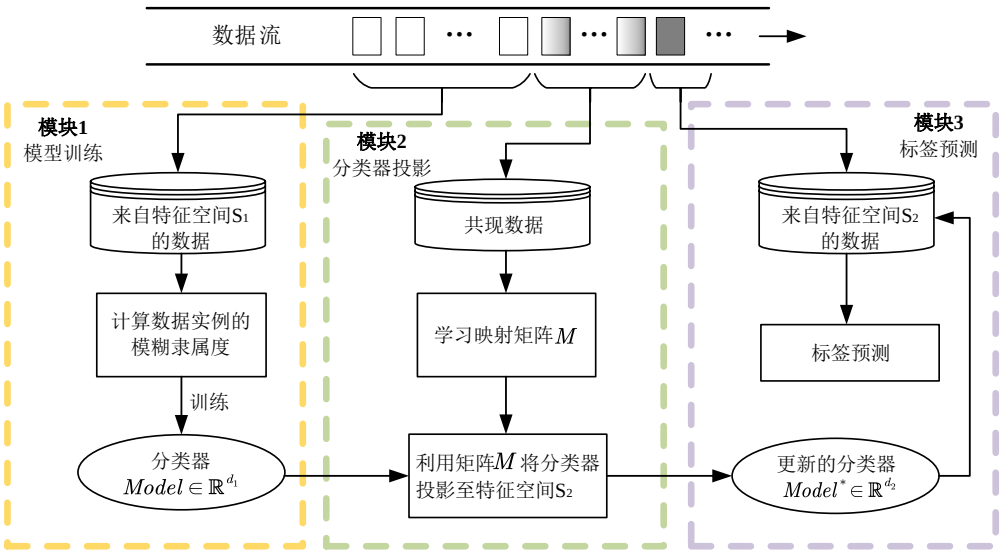


图 3-2 FENSL 的框架图

Fig. 3-2 The framework of FENSL

3.2.1 自适应模型训练

为解决问题（1），受模糊支持向量机[26]的启发，通过引入模糊隶属度值 $s_i \in [0,1]$ 来降低噪声影响，该值表示实例 $\boldsymbol{x}_i$ 属于某一类 y_i 的权值。模糊隶属度值用于确定不同实例对最终决策函数生成的影响，在鲁棒分类学习中起着重要作用。基于此，采用孪生支持向量机作为基分类器。该算法通过引入模糊隶属函数，既降低噪声的负面影响，又考虑支持向量的重要性。

首先，利用模糊隶属函数对噪声进行识别，选取正负类实例的均值点 $\bar{\boldsymbol{x}}_+$ 和 $\bar{\boldsymbol{x}}_-$ 之和作为类中心，如式(3-1)所示：

$$\begin{aligned}\bar{\boldsymbol{x}}_{+}&=\frac{1}{l_{+}}\sum_{i=1,y_i>0}^{l_{+}}\boldsymbol{x}_i,\\ \bar{\boldsymbol{x}}_{-}&=\frac{1}{l_{-}}\sum_{i=1,y_i<0}^{l_{-}}\boldsymbol{x}_i\end{aligned}\tag{3-1}$$

通过计算所有实例之间的距离，得到正类和负类的超球半径 r_{+} 和 r_{-}，如式(3-2)所示：

$$\begin{aligned}r_{+}&=\max\left\{\|\boldsymbol{x}_i-\bar{\boldsymbol{x}}_{+}\|\,|\,y_i>0\right\},\\ r_{-}&=\max\left\{\|\boldsymbol{x}_i-\bar{\boldsymbol{x}}_{-}\|\,|\,y_i<0\right\}\end{aligned}\tag{3-2}$$

以正实例的模糊隶属度 s_{i+} 为例，其计算公式如式(3-3)：

$$s_{i+}=\begin{cases}\lambda_1\left(1-\|\boldsymbol{x}_i-\bar{\boldsymbol{x}}_{+}\|/\left(r_{+}+\varepsilon\right)\right),\|\boldsymbol{x}_i-\bar{\boldsymbol{x}}_{+}\|\geq\|\boldsymbol{x}_i-\bar{\boldsymbol{x}}_{-}\|\\ \lambda_2\left(1-\|\boldsymbol{x}_i-\bar{\boldsymbol{x}}_{+}\|/\left(r_{+}+\varepsilon\right)\right),\|\boldsymbol{x}_i-\bar{\boldsymbol{x}}_{+}\|<\|\boldsymbol{x}_i-\bar{\boldsymbol{x}}_{-}\|\end{cases}\tag{3-3}$$

其中 λ_1，λ_2 用于平衡普通实例和噪声参数，$\lambda_1+\lambda_2=1$，$\varepsilon>0$ 是为了避免 $s_i=0$。例如，给定一个正类实例 x_i，若求出 $\|\boldsymbol{x}_i-\bar{\boldsymbol{x}}_{+}\|<\|\boldsymbol{x}_i-\bar{\boldsymbol{x}}_{-}\|$，则它的 s_i 值比较大，相反，如果该实例距离类中心较远，则通常会按比例为其分配较小的 s_i 值。类似的方法也可用于确定负类实例的模糊隶属度。

FENSL 的目标是构造两个非平行的超平面，这样构造的每个超平面都尽可能靠近类的实例，并且远离另一个类的实例，如式(3-4)所示：

$$\boldsymbol{w}_{+}^{\top}\boldsymbol{x}+b_{+}=0\ and\ \boldsymbol{w}_{-}^{\top}\boldsymbol{x}+b_{-}=0\tag{3-4}$$

其中 $\left(\boldsymbol{w}_{+},b_{+}\right)$ 和 $\left(\boldsymbol{w}_{-},b_{-}\right)$ 分别表示为由正负类实例构造的超平面系数。结合引入的模糊隶属函数，得到如式(3-5)和(3-6)所示的线性核权正则化模型：

$$\begin{aligned}&\min_{\boldsymbol{w}_{+},b_{+},\boldsymbol{\xi}_{-}}\ \eta_1c_1\|\boldsymbol{w}_{+}\|^2+\eta_2\|X_{+}\boldsymbol{w}_{+}+\boldsymbol{e}_1b_{+}\|^2+c_2s_{-}^{\top}\boldsymbol{\xi}_{-}\\ &\text{s.t.}\left(-1\right)*\left(\mathrm{X}_{-}\boldsymbol{w}_{+}+\boldsymbol{e}_2\mathrm{b}_{+}\right)+\boldsymbol{\xi}_{-}\geq\boldsymbol{e}_2,\boldsymbol{\xi}_{-}\geq0\end{aligned}\tag{3-5}$$

$$\begin{aligned}&\min_{\boldsymbol{w}_{+},b_{+},\boldsymbol{\xi}_{+}}\ \eta_1c_3\|\boldsymbol{w}_{-}\|^2+\eta_2\|X_{-}\boldsymbol{w}_{-}+\boldsymbol{e}_2b_{-}\|^2+c_4\boldsymbol{s}_{+}^{\top}\boldsymbol{\xi}_{+}\\ &\text{s.t.}\left(+1\right)*\left(\mathrm{X}_{+}\boldsymbol{w}_{-}+\boldsymbol{e}_1\mathrm{b}_{-}\right)+\boldsymbol{\xi}_{+}\geq\boldsymbol{e}_1,\boldsymbol{\xi}_{+}\geq0\end{aligned}\tag{3-6}$$

其中 $c_i\left(i=1,2,3,4\right)$ 表示惩罚因子，ξ_{-}，ξ_{+} 表示松弛变量，e_1 和 e_2 表示适当维数的单位列向量，s_{i-}，s_{i+} 表示模糊隶属度，η_1 和 η_2 表示平衡系数，这里设置 $\eta_1=\eta_2=\frac{1}{2}$。值得注意的是，$s_{i-}$，$s_{i+}$ 越小，参数 ξ_{-}，ξ_{+} 的影响越小。通过这种方式，它将反映训练实例的重要性。则对于一个测试实例 $\boldsymbol{x}$，可以给出预测函

数如式(3-7)：

$$f\left(\boldsymbol{x}\right)=\arg\min_{\pm}\frac{\left|\boldsymbol{w}_{\pm}^{\top}\boldsymbol{x}+b_{\pm}\right|}{\|\boldsymbol{w}_{\pm}^{\top}\|} \tag{3-7}$$

由此可知，决策超平面可以有效地检测噪声数据点并削弱它们的影响。算法 3-1 给出 FENSL 方法的整体流程。

算法 3-1 FENSL

输入：训练数据实例 $X_{\pm}^{(1)}$，测试数据实例 $X_{\pm}^{(2)}$，参数 $c_i\left(i=1,2,3,4\right)$.

输出：测试数据实例 $X_{\pm}^{(2)}$ 的预测标签 $\hat{Y}_{\pm}^{(2)}$.

1. 初始化分类模型 $\boldsymbol{w}_{\pm}=[0,0,\cdots,0]\in\mathbb{R}^{S_1}$；

训练分类模型：

2. 计算 $X_{\pm}^{(1)}$ 中每个实例的模糊隶属度 s_i；

3. 利用训练数据 $X_{\pm}^{(1)}$ 训练分类模型；

学习映射关系：

4. 根据算法 3-2 学习映射矩阵 M；

5. 将模型从 S_1 投影至 S_2 得到分类模型 $\boldsymbol{u}_{\pm}=\left[M^{\top}\boldsymbol{w}_{\pm}^{\top},b_{\pm}^{\top}\right]$；

进行标签预测：

6. 利用分类模型 $\boldsymbol{u}_{\pm}$ 进行预测标签 $\hat{Y}_{\pm}^{(2)}$；

3.2.2 异构特征映射矩阵

针对问题（2），FENSL 方法采用局部加权线性回归方法[79]来拟合两个特征空间之间存在的映射关系，这里用 M 来表示线性映射的系数矩阵。因此，利用共现数据可计算得到系数矩阵 M，如式(3-8)所示：

$$\min_{M\in\mathbb{R}^{d_2\times d_1}}\sum_{i=T_1-B+1}^{T_1}K^i\left(M^{\top}\boldsymbol{x}_i^{(2)}-\boldsymbol{x}_i^{(1)}\right)^2 \tag{3-8}$$

其中，高斯核权值 K 的定义如式(3-9)所示：

$$K\left(i,i\right)=e^{-\frac{\left(\boldsymbol{x}_i^{(2)}-\boldsymbol{x}^{(2)}\right)^2}{2k^2}} \tag{3-9}$$

式(3-8)中最小化问题的最优解如式(3-10)所示：

$$M_*=M_1^{-1}M_2 \tag{3-10}$$

其中， M_1 和 M_2 可根据式(3-11)进行计算：

$$M_1 = \sum_{i=T_1-B+1}^{T_1} \boldsymbol{x}_i^{(2)} K \boldsymbol{x}_i^{(2)\top}, \quad M_2 = \sum_{i=T_1-B+1}^{T_1} \boldsymbol{x}_i^{(2)} K \boldsymbol{x}_i^{(1)\top} \tag{3-11}$$

在得到线性映射关系后，训练模型的参数可以通过 $M^{\top}\boldsymbol{w}_{\pm}$ 映射到新的特征空间以继续预测新到达的样本，算法 3-2 给出学习映射矩阵的算法主要流程。

算法 3-2 学习映射矩阵

输入： 共现数据实例 $X_{\pm}^{(c)}$.

输出： 线性映射最优解 M_*.

1. 计算核矩阵 $K(i,i)$;
2. **for** $i = T_1 - B + 1, T_1 - B + 2, \ldots, T_1$ **do**
3. 计算 $M_1 = M_1 + \boldsymbol{x}_i^{(2)} K \boldsymbol{x}_i^{(2)\top}$ 和 $M_2 = M_2 + \boldsymbol{x}_i^{(2)} K \boldsymbol{x}_i^{(1)\top}$;
4. **end for**
5. 根据式(3-10)计算 M_*;

3.3 实验设计与结果分析

在本节中，首先介绍实验配置，包括数据集、对比方法和相关参数设置。为验证所提方法的有效性，分别进行四组实验并对实验结果进行分析与讨论。

3.3.1 数据集与实验设置

为验证本章方法有效性，分别在 6 个人工数据集上进行实验，数据集的详细信息如表 3-1 所示。其中 d_1 表示原始数据集的特征空间 S_1 的维度，d_2 表示人工生成的新特征空间 S_2 的维度。

表 3-1 数据集描述

Table 3-1 Description Of Datasets

数据集	样本总体数量	正类样本数量	负类样本数量	d_1	d_2
Credit-a	653	357	296	15	10
Credit-g	1000	300	700	20	14
DNA12	940	485	464	180	125
German	1000	300	700	59	41
Kr-vs-kp	3196	1527	1669	36	25
Splice	3175	1648	1527	60	42

首先通过随机高斯矩阵将原始数据人工映射到另一个特征空间，然后将数据集复制 5 次到一个更大的特征空间。实验设置两个阶段中的实例数量 $T_1:T_2=9:1$，共现实例数量 $B=T_2$。

为评价所提方法的分类性能，将提出的方法 FENSL 与方法 FR-TSVM[84]以及文献[47]中提出的方法 FESL-c、FESL-s、NOGD、ROGD-f 和 ROGD-u 进行比较。

• NOGD：从头开始调用 OGD 算法在新的特征空间中训练数据。

• ROGD-f：通过线性映射，将具有新特征的数据恢复到旧特征，然后使用学习到的模型进行预测。

• ROGD-u：通过线性映射，将具有新特征的数据恢复到旧特征，并使用 OGD 继续更新学习到的模型进行预测。

• FESL-s：结合 NOGD 和 ROGD-u 两个基本模型对新特征空间下的数据进行预测

• FESL-c：通过选择 NOGD 和 ROGD-u 两个基本模型的最优模型对新特征空间下的数据进行预测。

• FR-TSVM：从头开始训练数据并在新的特征空间中对数据进行预测。

• FENSL：将 S1 中历史模型映射至 S2 中，从新的特征空间中预测数据。

对于 NOGD、ROGD-f、ROGD-u、FESL-s 和 FESL-c 方法，为保证公平性，保留文献[47]中参数 $\tau_t=\frac{1}{c\sqrt{t}}$ 和 c 的设置，其中 c 在 $\{1,10,50,100,150\}$ 范围内搜索。研究参数 $c_i\,(i=1,2,3,4)$ 对分类精度的影响，其中 $c_1=c_3, c_2=c_4$，其取值区间为 $\left\{2^{(-2)},2^{(-1)},2^0,2^1,2^2,2^3,2^4\right\}$，为便于表达，这里定义 $C_0(=c_1=c_3)$ 和 $C_1(=c_2=c_4)$。所有方法的分类精度通过 10 次独立重复实验得到，取 10 次的平均值得到最终的预测结果。

本节共设置 4 组实验：在第一组实验中，在无噪声的人工数据集上进行对比实验并统计分类准确率；在第二组实验中，将上述七个对比算法部署在含噪声的人工数据集上进行实验并统计分类准确率；第三组实验是 FENSL 的抗噪声性能分析；第四组实验是参数灵敏度实验，验证参数 $c_i\,(i=1,2,3,4)$ 的不同取值对分类精度的影响。

3.3.2 无噪声数据集上对比实验

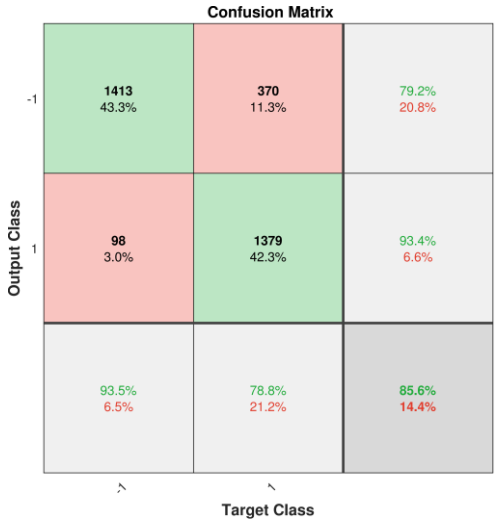
为评估 FENSL 的分类性能，本节在 6 个不同的无噪声人工数据集上进行实验，最终分类准确率如表 3-2 所示。观察发现在 6 个无噪声数据集中，FENSL 在 4 个数据集的分类准确率上优于其他方法。FR-TSVM 在另 2 个数据集上表现良好，说明充分利用少量共现数据来训练新模型具有一定的优势。在大多数情况下，ROGD-f 比 ROGD-u 表现得更好，说明在本章研究的特征演化场景中，通过将新到数据逆向映射回旧特征空间会降低数据质量，导致在大量数据上已训练好的固定分类器反而性能更加稳定。因此，历史分类模型具有很高的利用价值，不应随意丢弃。OGD 算法的分类性能与数据量呈正相关，NOGD 算法在处理新特征空间的少量数据时，分类性能在所有方法中较差。受到 NOGD 训练的基分类器的负面影响，FESL-c 和 FESL-s 算法同样表现不佳。

FENSL 在 6 个数据集上生成的混淆矩阵如图 3-3 所示。其中“Output class”作为真实标签，“Target class”是预测标签，由于数据中类别分布不平衡，在分类结果中对少数类的分类错误率较高。例如，German 数据集中正类样本数量远少于负类样本数量，对正类样本错误分类率为 16.6%，对负类样本错误分类率为 7.2%，说明由于分类器更倾向于对多数类样本进行预测。以上结果表明，FENSL 很好地适应这类特征演化问题，具有很高的实际应用价值。

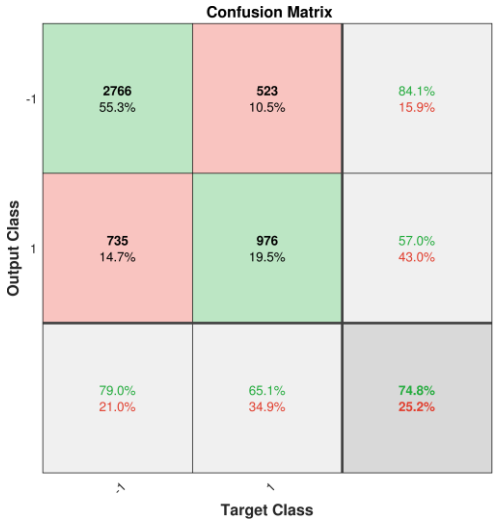
表 3-2 无噪声数据集上的分类准确率

Table 3-2 Classification accuracy on datasets without noise

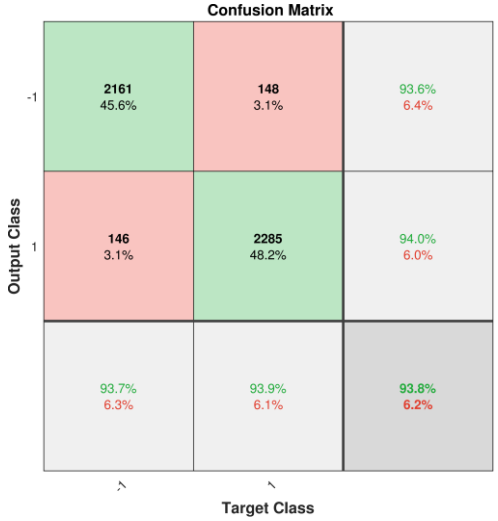
Dataset	Credit-a	Credit-g	DNA12	German	Kr-vs-kp	Splice
NOGD	.6773±.0324	.5974±.0258	.5091±.0279	.6026±.0301	.5218±.0150	.5120±.0088
FR-TSVM	.8549±.0093	**.7570±.0253**	.8996±.0150	.7156±.0181	**.8753±.0073**	.7548±.0066
ROGD-f	.8506±.0114	.7418±.0146	.9207±.0089	.7034±.0137	.7508±.0194	.7687±.0104
ROGD-u	.8439±.0178	.7392±.0151	.9179±.0158	.7036±.0143	.7807±.0227	.7611±.0117
FESL-s	.8396±.0164	.7386±.0155	.9137±.0159	.7018±.0143	.7780±.0217	.7607±.0115
FESL-c	.8408±.0173	.7392±.0151	.9160±.0159	.7020±.0141	.7792±.0228	.7607±.0116
FENSL	**.8564±.0102**	.7484±.0129	**.9386±.0119**	**.7622±.0086**	.8690±.0083	**.7766±.0101**



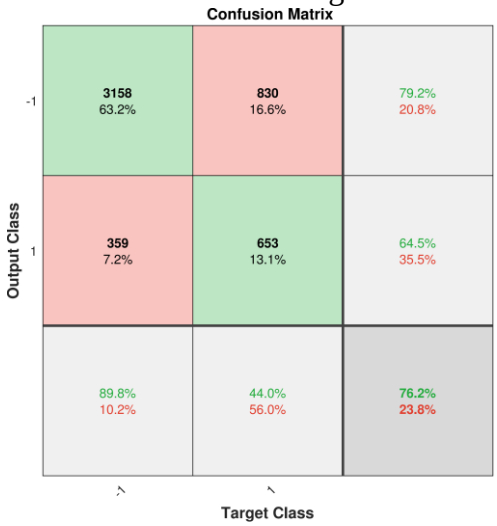
（a）Credit-a



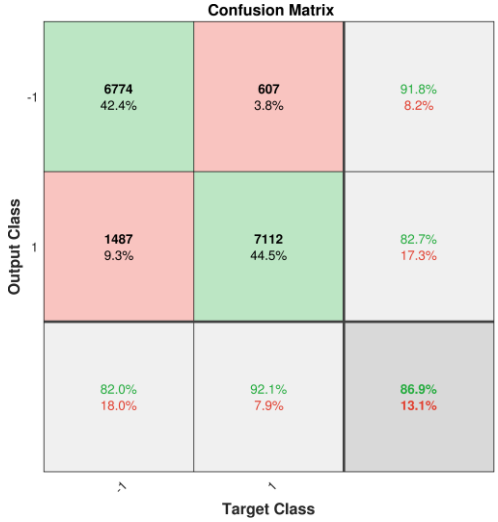
（b）Credit-g



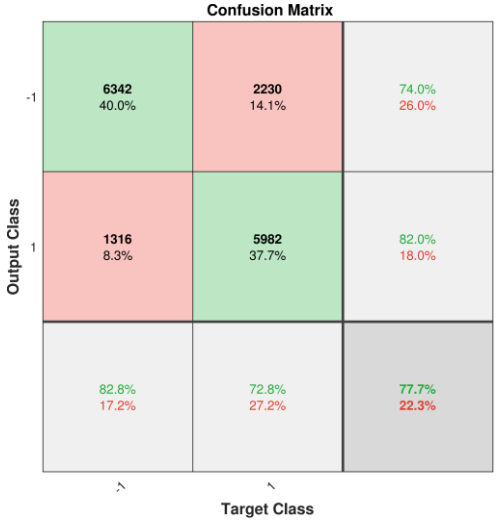
（c）DNA12



（d）German



（e）Kr-vs-kp



（f）Splice

图 3-3 混淆矩阵

Fig. 3-3 Confusion Matrix

3.3.3 含噪声数据集上对比实验

为验证 FENSL 的抗噪声能力，在 6 个不同的含噪声人工数据集上采用 7 种方法最终的分类准确率见表 3-3。

表 3-3 含噪声数据集上的分类准确率

Table 3-3 Classification accuracy on datasets with noise

Dataset	Credit-a	Credit-g	DNA12	German	Kr-vs-kp	Splice
NOGD	.6773±.0324	.5974±.0258	.5091±.0279	.6026±0.0301	.5218±.0150	.6372±.0194
FR-TSVM	.8543±.0116	.7146±.0144	.9019±.0148	.7196±0.0172	**.8791±.0072**	.7481±.0050
ROGD-f	.8402±.0180	.7102±.0270	.8941±.0146	.7034±0.0137	.7394±.0168	.6448±.0149
ROGD-u	.8420±.0171	.7258±.0141	.8892±.0160	.7034±0.0139	.7727±.0255	.6427±.0162
FESL-s	.8362±.0184	.7244±.0155	.8865±.0172	.7012±0.0139	.7697±.0253	.6354±.0170
FESL-c	.8396±.0174	.7258±.0141	.8871±.0160	.7016±.0138	.7715±.0255	.6453±.0169
FENSL	**.8558±.0119**	**.7264±.0138**	**.9072±.0164**	**.7500±.0121**	.8399±.0082	**.7551±.0104**

噪声样本由均值为 0，方差为 0.2 的高斯分布$\epsilon \in \mathcal{N}(0,0.2)$随机产生。实验其他设置与文献[34]保持一致。可以看出，FENSL 方法在 6 个人工数据集上分类性能均优于其他方法，这表明该方法在处理噪声数据流分类方面的有效性。NOGD 的性能表现最差，这是因为该方法具有新特征的数据数量过少不足以支持训练性能优异的分类模型。此外，在 6 个数据集中的 4 个中，ROGD-f 的性能优于 ROGD-u。这种现象突出了保留并重用旧模型的重要性，它们有助于保证分类模型稳定性和有效性。同时，通过分析表中 FENSL 的最佳结果发现，在噪声环境下，FENSL 的精度在 5 个数据集上仅下降约 0.01~0.03，在 2 个数据集上仅下降约 0.006。这也可以说明 FENSL 具有一定的抗噪声能力。基于以上分析，FENSL 在学习带有噪声的特征演化数据流的同时，获得良好的分类性能和鲁棒性。

表 3-4 展示在含噪声的数据流中各对比算法所需的训练时间。在其中 4 个数据集上 FENSL 算法的运行速度最快，表明该算法学习效率相对优于其他对比算法，特别是 FESL-s 和 FESL-c 算法。主要原因是 FENSL 通过使用旧模型在新特征数据上进行训练，而 FESL 增加新旧分类器的集成学习，导致训练时间较长。基于以上分析，FENSL 在学习带有噪声的特征演化数据流时取得良好的分类性能和鲁棒性。

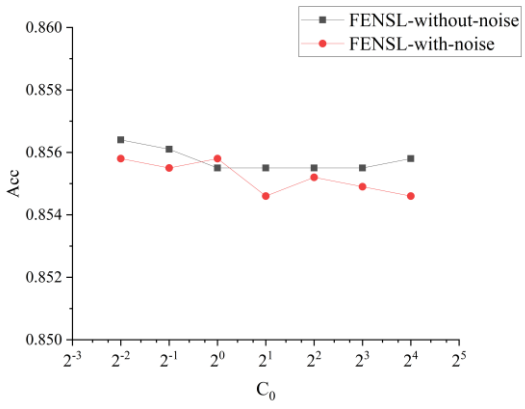
表 3-4 运行时间（秒）

Table 3-4 Running time （second）

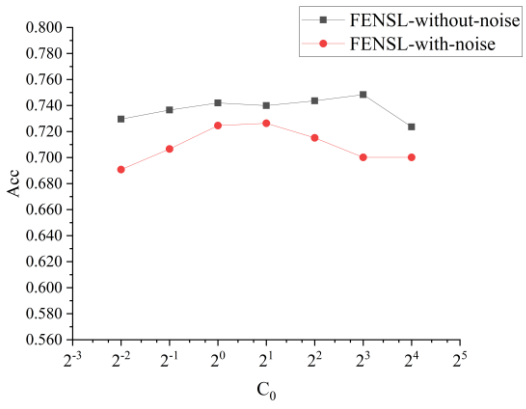
Dataset	Credit-a	Credit-g	DNA12	German	Kr-vs-kp	Splice
NOGD	0.0032	0.0077	0.0197	0.0087	0.0204	0.0273
FR-TSVM	0.0034	0.0036	**0.0039**	0.0029	0.0075	**0.0085**
ROGD-f	0.0030	0.0044	0.0240	0.0089	0.0208	0.0276
ROGD-u	0.0028	0.0051	0.0196	0.0101	0.0245	0.0256
FESL-s	0.0028	0.0050	0.0188	0.0079	0.0201	0.0278
FESL-c	0.0038	0.0050	0.0181	0.0081	0.0212	0.0287
FENSL	**0.0021**	**0.0029**	0.0040	**0.0029**	**0.0072**	0.0160

3.3.4 抗噪分析

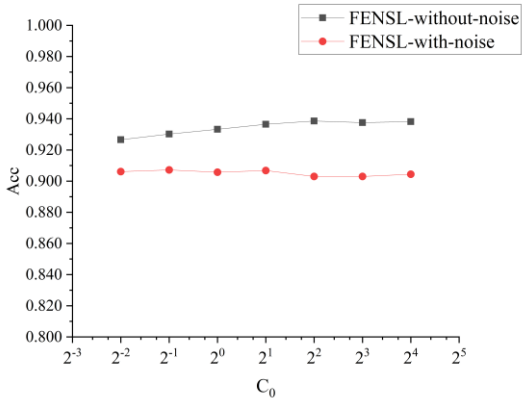
为进一步验证 FENSL 的鲁棒性，本节在参数c_1, c_3变化的含噪声/无噪声情况下测试所提方法。图 3-4 给出 6 个人工数据集的分类准确率变化趋势，这里将在无噪声数据集上的 FENSL 方法称为“FENSL-without-noise”，在含噪声数据集上测试 FENSL 方法称为“FENSL-with-noise”。可观察到在大多数情况下，该方法在无噪声数据集上表现更好。就 FENSL 在噪声/无噪声条件下的性能而言，最大偏差小于 0.08，性能下降幅度对于大多数实际应用来说是可以接受的。在一定的参数设置下，FENSL 在含噪声和无噪声情况下的性能非常接近。值得注意的是，在 credit-a 数据集的实验结果中观察到不寻常的结果：当$C_0 = 2^0$时，带噪声 FENSL 的精度甚至超过无噪声 FENSL。这可以归因于噪声可能被添加到不重要的实例中，因此不会显著影响分类器训练并导致类似的结果。综上所述，实验结果表明模糊隶属度确实发挥了重要作用，保证 FENSL 的抗噪声能力。



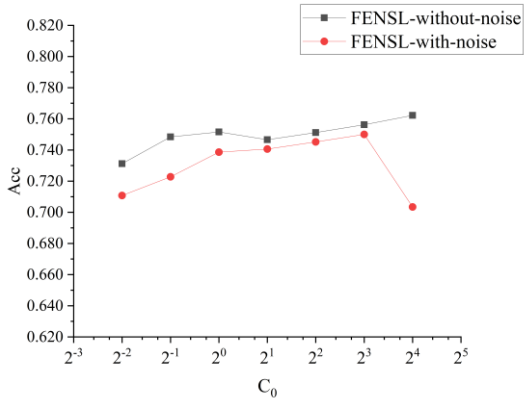
（a）Credit-a



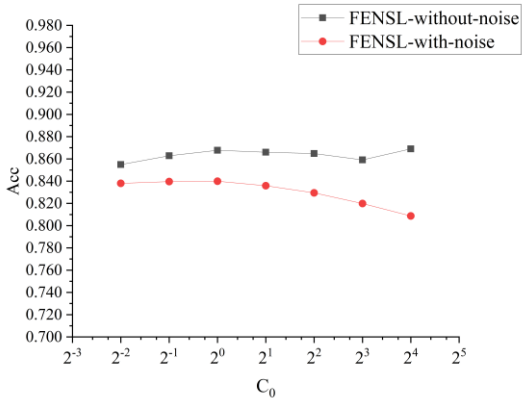
（b）Credit-g



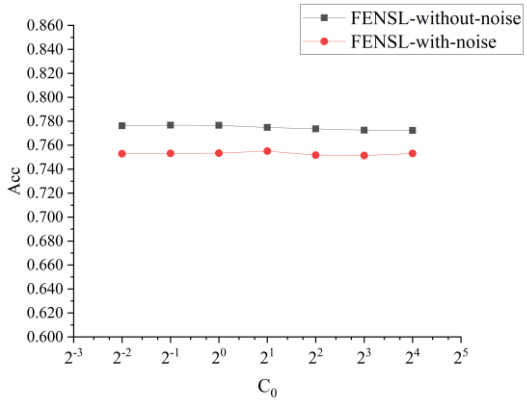
（c）DNA12



（d）German



（e）Kr-vs-kp



（f）Splice

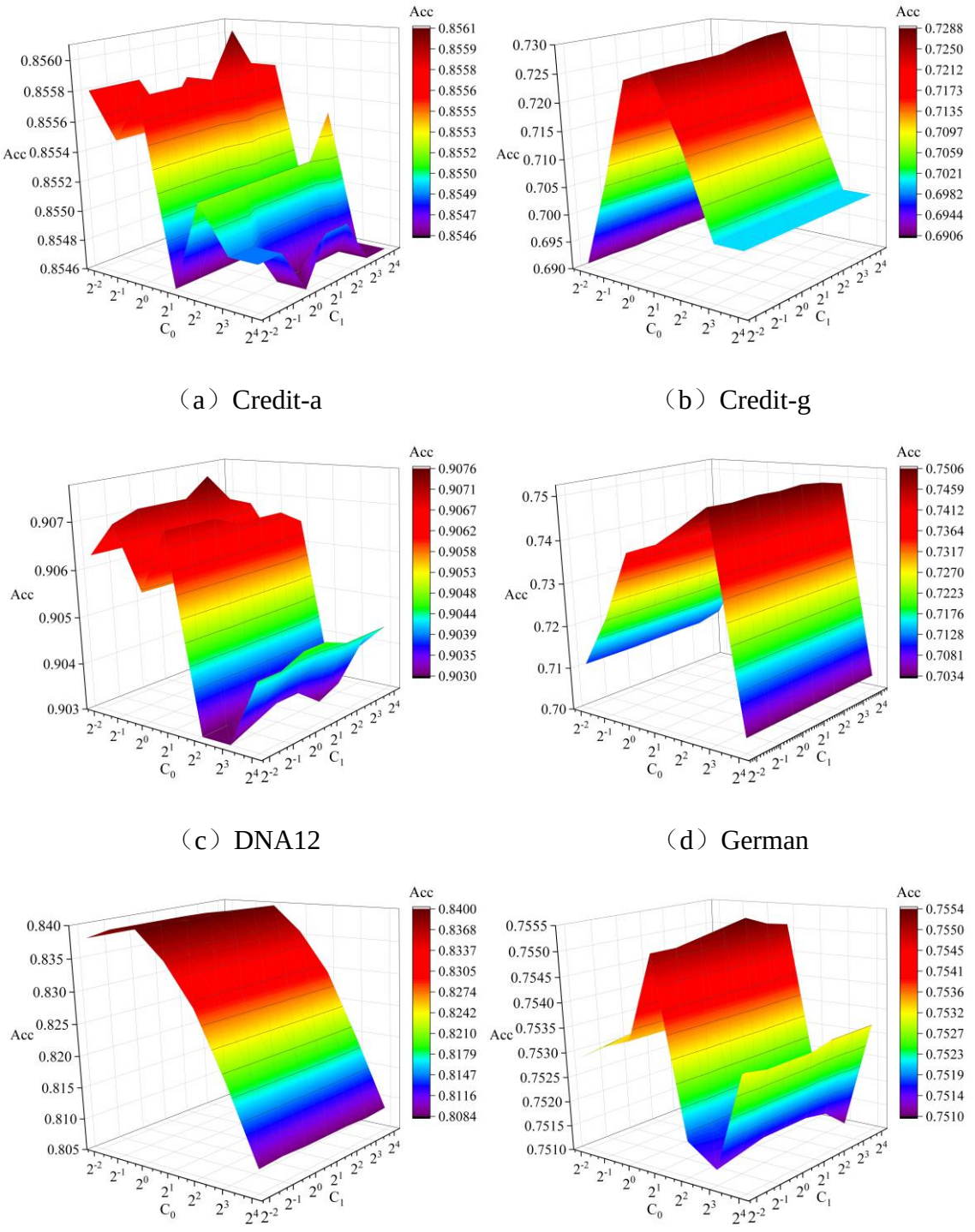
图 3-4 FENSL 的抗噪声性能实验结果

Fig. 3-4 Results of FENSL's anti-noise performance

3.3.5 参数敏感性分析

为进一步研究不同参数 $c_i\,(i=1,2,3,4)$ 对 FENSL 方法稳定性的影响，本节进行了含噪声数据流上的参数实验。

图 3-5 显示在含噪声的人工数据集上改变可调参数时的精度，测试参数 $c_i\,(i=1,2,3,4)$ 用于平衡自适应误差最小化和边际最大化的影响，其值在范围 $\{2^{(-2)},2^{(-1)},2^0,2^1,2^2,2^3,2^4\}$ 内，总共进行 49 次参数实验。分通过析不同参数 C_0 和 C_1 对实验结果的影响，发现 FENSL 的分类精度有一定波动幅度，而随着 C_1 的增加，在 6 个有噪声的数据集中，准确率曲线的波动相对稳定。当 C_0 约在 $(2^0,2^1)$ 范围内时，FENSL 大多能达到最佳效果。C_0 的变化对性能曲线的波动有显著影响。



（a）Credit-a　　（b）Credit-g

（c）DNA12　　（d）German

（e）Kr-vs-kp　　（f）Splice

图 3-5 参数实验

Fig. 3-5 Parameter experiment

通过以上分析，可以观察发现分类准确率在 3 个数据集上基本在 0.03~0.05 的范围内，在不同C_0值的 3 个数据集上基本在 0.001~0.004 的范围内，且在不同C_0值的范围内波动较大，因此每个参数对 FENSL 性能的影响是不同的，尤

其是参数c_1和c_3，在分类模型需要选择合适的参数。

3.4 本章小结

本章提出基于映射策略的 FENSL 方法来处理特征演化数据流。该方法首先计算从现有特征空间中收集的数据的模糊隶属度，并学习正类和负类的两个决策超平面；然后，使用共现数据的两部分来学习两个不同特征空间之间的映射矩阵。利用映射矩阵可以将训练好的模型从旧的特征空间投影到新的特征空间，从而构建一个高效的分类模型对新特征数据进行预测；最后，通过对比实验验证了该方法的有效性。

第4章 基于集成学习的特征演化数据流分类方法

集成学习[80,81]通过调整各个基分类器权重以适应变化的数据流环境，灵活的复合结构使集成学习模型具有高效稳健的性能优势。Bagging、Boosting 和 Stacking 等作为经典集成策略被广泛研究与应用[82]。数据流集成学习方法首先通过划分数据块或在线学习的方式训练多个基分类器，在对新到样本进行分类时，基分类器各自输出对应预测结果，然后按照加权集成、选择集成或弃权机制等决策规则对预测结果进行合并输出。

特征演化问题中，新特征出现的初期，重新训练的分类器性能不佳，历史分类器可有效辅助预测，但随新特征数据大量积累，新分类器性能上升而历史分类器将逐渐被淘汰。为此，Hou 等[47]提出两种集成学习方法：基于结合的方法和基于动态选择的方法，集成历史分类器与重新训练的分类器以实现共同预测。Liu 等[48]提出的基于被动-主动更新策略的特征演化学习算法，采用同样的两种集成方法合并各阶段训练的基分类器，以降低特征空间变化对分类性能的影响。因此，集成学习可用来平衡历史分类器与重新训练的分类器的预测性能，是处理特征演化问题的重要技术手段。

在学习特征演化数据流时，不仅需要采用有效的集成方法动态合并不同基分类器的预测结果，还需要利用新数据对分类模型进行增量更新以保留最新知识。同时，数据采集过程中不可避免的噪声干扰，将对分类性能产生负面影响。基于此，本章提出基于集成学习的特征演化数据流分类方法，首先引入增量模糊孪生支持向量机，在提升模型预测准确率的同时降低噪声干扰；然后，通过投影策略实现模型重用以适应特征演化；最后，改进基于损失函数的决策规则，形成两种新的集成学习策略以进行标签组合预测：加权组合和选择最优，从而提高分类模型的稳定性与鲁棒性。

4.1 问题描述与分析

本章所研究的问题场景如图 4-1：在$1\sim T_1-B$阶段，数据样本均来自特征空间S_1；当进入$T_1-B+1\sim T_1$阶段时，新的特征空间S_2出现，与S_1共同描述数

据。这样每个数据样本的特征空间都可被分为两个部分，因此将这些数据称之为“共现数据”；随后，进入 $T_1+1\sim T_2$ 阶段，新到达数据仅来自特征空间 S_2 下。该过程构成一个变化周期，可以循环往复，每个周期包含两个特征空间。值得注意的是，数据流以微批次（Mini-Batch）进行收集并进行增量训练，各阶段的数据特征保持相对稳定，但当前阶段无法预知下一阶段的特征空间及其变化，同时整个数据流上均含有噪声干扰。

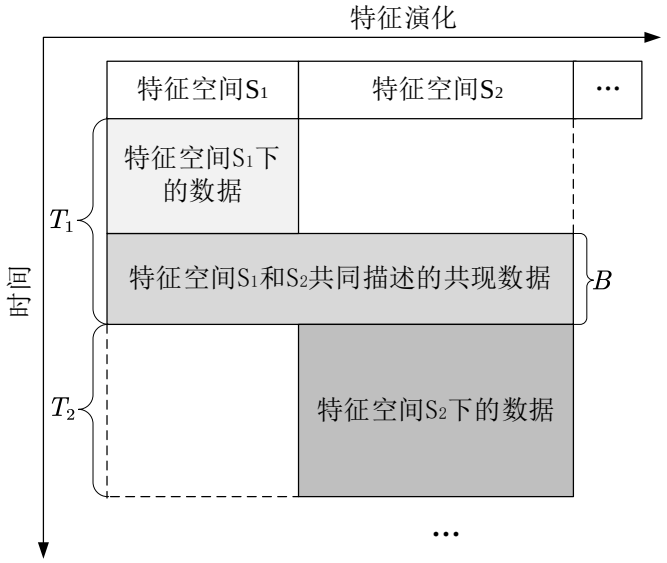


图 4-1 特征演化数据流

Fig. 4-1 Feature evolvable data stream

为解决上述问题研究场景，通过增量学习方法建立预测模型，需要解决如下三个关键问题：

（1）如何消除噪声干扰的，增量训练鲁棒的分类模型；

（2）如何在异构特征场景下实现模型拓展与重用，充分利用历史知识提升分类模型的准确性；

（3）如何利用集成策略，结合更新重用的旧模型和重新训练的新模型进行共同预测。

4.2 特征演化数据流的集成分类模型

本节针对上述归纳的关键问题设计 ILFE 方法框架，并从模型训练与更新与集成学习策略两个方面进行阐述。

4.2.1 模型增量更新

由于增量孪生支持向量机可以快速高效地训练分类模型，同时模糊隶属度在鲁棒分类学习中起着关键作用，考虑采取文献[83]中模糊有界孪生支持向量机算法（Fuzzy Bounded Twin Support Vector Machine, FBTWSVM）对数据流进行增量训练与更新模型。

在解决含噪声或异常点的数据流分类问题时，通过设计合适的模糊隶属度函数，为每个样本计算模糊隶属度值，以表征不同样本对决策函数的影响程度。ILFE 方法采用的模糊隶属度函数根据每个数据样本到类中心距离来计算样本的模糊隶属度值。首先选取正类实例和负类实例的均值点 $\bar{\boldsymbol{x}}_+$ 和 $\bar{\boldsymbol{x}}_-$ 作为类中心如式(4-1)所示：

$$\begin{aligned}\bar{\boldsymbol{x}}_+ &= \frac{1}{l_+}\sum_{i=1,y_i>0}^{l_+}\boldsymbol{x}_i\,,\\ \bar{\boldsymbol{x}}_- &= \frac{1}{l_-}\sum_{i=1,y_i<0}^{l_-}\boldsymbol{x}_i\end{aligned} \tag{4-1}$$

通过计算所有实例之间的距离，得到式(4-2)中正类和负类的超球半径：

$$\begin{aligned}r_+ &= \max\left\{\|\boldsymbol{x}_i-\bar{\boldsymbol{x}}_+\|\,|\,y_i>0\right\},\\ r_- &= \max\left\{\|\boldsymbol{x}_i-\bar{\boldsymbol{x}}_-\|\,|\,y_i<0\right\}\end{aligned} \tag{4-2}$$

那么，数据样本的模糊隶属度函数[84]如式(4-3)所示：

$$s_i=\begin{cases}\lambda_1(1-\|\boldsymbol{x}_i-\bar{\boldsymbol{x}}_+\|/(r_++\varepsilon)) & ,\|\boldsymbol{x}_i-\bar{\boldsymbol{x}}_+\|\geqslant\|\boldsymbol{x}_i-\bar{\boldsymbol{x}}_-\|\ \&y_i=+1\\ \lambda_2(1-\|\boldsymbol{x}_i-\bar{\boldsymbol{x}}_+\|/(r_++\varepsilon)) & ,\|\boldsymbol{x}_i-\bar{\boldsymbol{x}}_+\|<\|\boldsymbol{x}_i-\bar{\boldsymbol{x}}_-\|\ \&y_i=+1\\ \lambda_1(1-\|\boldsymbol{x}_i-\bar{\boldsymbol{x}}_-\|/(r_-+\varepsilon)) & ,\|\boldsymbol{x}_i-\bar{\boldsymbol{x}}_-\|\geqslant\|\boldsymbol{x}_i-\bar{\boldsymbol{x}}_+\|\ \&y_i=-1\\ \lambda_2(1-\|\boldsymbol{x}_i-\bar{\boldsymbol{x}}_-\|/(r_-+\varepsilon)) & ,\|\boldsymbol{x}_i-\bar{\boldsymbol{x}}_-\|<\|\boldsymbol{x}_i-\bar{\boldsymbol{x}}_+\|\ \&y_i=-1\end{cases} \tag{4-3}$$

其中 λ_1，λ_2 是用于平衡普通实例和噪声的参数且 $\lambda_1+\lambda_2=1$，$\varepsilon>0$ 是为避免 $s_i=0$。如果实例距离类中心较远，说明该实例重要程度越低，则通常会按比例赋予实例更小的 s_i 值。

为对数据流进行实时处理，可通过增量学习模糊孪生支持向量机，同时引入有效的模糊隶属度函数，既降低异常值所带来的噪声，又考虑支持向量的重要性。首先，本章所提 ILFE 算法采用孪生支持向量机作为基分类器，目标是计算式(4-4)所示的两个非平行超平面：

$$\boldsymbol{w}_+^{\top}\boldsymbol{x}+b_+=0\ and\ \boldsymbol{w}_-^{\top}\boldsymbol{x}+b_-=0 \tag{4-4}$$

其中$(\boldsymbol{w}_+, b_+)$和$(\boldsymbol{w}_-, b_-)$分别表示为由正负类实例构造的超平面系数。

通过引入模糊隶属度函数和边际参数，具有线性核的 ILFE 的原始问题定义如式(4-5)和式(4-6)所示：

$$\begin{aligned}&\min_{\boldsymbol{w}_+, b_+, \boldsymbol{\xi}_-} c_1\eta\left(\| \boldsymbol{w}_+\|^2 + b_+^2\right) + \eta\| X_+\boldsymbol{w}_+ + \boldsymbol{e}_+ b_+\|^2 + c_2\boldsymbol{s}_-^\top\boldsymbol{\xi}_- \\ &\text{s.t.} -(X_-\boldsymbol{w}_+ + \boldsymbol{e}_+ b_+) + \boldsymbol{\xi}_- \geqslant \boldsymbol{e}_-, \boldsymbol{\xi}_- \geqslant 0\end{aligned} \tag{4-5}$$

$$\begin{aligned}&\min_{\boldsymbol{w}_-, b_-, \boldsymbol{\xi}_+} c_3\eta\left(\| \boldsymbol{w}_-\|^2 + b_-^2\right) + \eta\| X_-\boldsymbol{w}_- + \boldsymbol{e}_- b_-\|^2 + c_4\boldsymbol{s}_+^\top\boldsymbol{\xi}_+ \\ &\text{s.t.}(X_+\boldsymbol{w}_- + \boldsymbol{e}_- b_-) + \boldsymbol{\xi}_+ \geqslant \boldsymbol{e}_+, \boldsymbol{\xi}_+ \geqslant 0\end{aligned} \tag{4-6}$$

其中c_1和c_3是间隔参数，c_2和c_4是惩罚参数，并加入b_+, b_-使结构风险最小化，$\boldsymbol{\xi}_+, \boldsymbol{\xi}_-$为松弛变量，$e_+, e_-$是单位列向量，平衡系数$\eta = 0.5$，同时$\boldsymbol{s}_+ \in \mathbb{R}^{l_+}$和$\boldsymbol{s}_- \in \mathbb{R}^{l_-}$分别是正类、负类实例相关的模糊隶属度值，它们为加权正则化模型的鲁棒性提供保障。

$\boldsymbol{w}_\pm$和$b_\pm$表示为正类和负类样本构造的超平面系数，那么可得出式(4-7)所示的预测函数：

$$f(\boldsymbol{x}) = \arg\min_{\pm} \frac{\left|\boldsymbol{w}_\pm^\top \boldsymbol{x} + b_\pm^*\right|}{\| \boldsymbol{w}_\pm^*\|} \tag{4-7}$$

为避免完全重构模型，需将新的数据点连续地集成到现有模型中从而增量学习数据流。该方法基于收缩启发式（Shrinking Heuristic）方法，结合新样本的模糊信息进行模型更新，从上一个训练步骤中推断出投影梯度的最小值$\min_{i-1}$和最大值$\max_{i-1}$，并通过式(4-8)计算新到样本点的投影梯度：

$$\nabla_i f(\boldsymbol{\alpha}) = -H_{i-}\boldsymbol{u}_+ - 1 \tag{4-8}$$

判断其数值大小是否在$(\min_{i-1}, \max_{i-1})$区间内来区分支持向量和影响分类边界的异常点。同时为避免增量过程导致模型维数的持续增长，不断去除对模型精度干扰较小或没有干扰的数据来控制模型维数。

为提升异构特征空间的分类效率，ILFE 方法采用局部加权线性回归算法，学习异构空间的线性映射关系对应的系数矩阵M，实现模型重用。利用共现数据可学习M，如式(4-9)所示：

$$\min_{M \in \mathbb{R}^{d_1 \times d_2}} \sum_{i=T_1-B+1}^{T_1} K^i \left(M^\top \boldsymbol{x}_i^{(2)} - \boldsymbol{x}_i^{(1)}\right)^2 \tag{4-9}$$

算法 4-1 ILFE

输入： 训练样本 $\boldsymbol{X}_{\pm}^{(1)}=\{(\boldsymbol{x}_{\pm}^{(1)},y_i)\}_1^{T_1}$ 以及共现数据样本 $\boldsymbol{X}_{\pm}^{(c)}=\{([\boldsymbol{x}_{\pm}^{(1)},\boldsymbol{x}_{\pm}^{(2)}],y_i)\}_{T1-B+1}^{T1}$ 和新特征样本 $\boldsymbol{X}_{\pm}^{(2)}=\{(\boldsymbol{x}_{\pm}^{(2)},y_i)\}_{T_1+1}^{T_2}$.

输出： 对 $\boldsymbol{X}_{\pm}^{(2)}$ 的标签预测结果.

1. **for** i=1,2,…,T_1 **do**
2. 接收样本 $X_{\pm\ i}^{(1)}$；
3. 选择合适的参数 $c_j\,(j=1,2,3,4)$；
4. 计算各样本点模糊隶属度 $\boldsymbol{s}_i$；
5. 更新超平面 $\bar{\boldsymbol{u}}_{\pm}=((\bar{\boldsymbol{w}}_{\pm})^{\top},\bar{b}_{\pm})$；
6. **if** i>T_1-B **then**
7. 接收共现数据样本 $\boldsymbol{X}_{\pm}^{(c)}$；
8. 初始化 $M_1=O^{d_2\times d_2}$ 和 $M_2=O^{d_2\times d_1}$；
9. **for** i=T_1-B+1,T_1-B+2,…,T_1 **do**
10. 接收样本 $\boldsymbol{X}_{\pm}^{(c)}=\{([\boldsymbol{x}_{\pm}^{(1)},\boldsymbol{x}_{\pm}^{(2)}],y_i)\}_{T1-B+1}^{T1}$；
11. 计算核矩阵 $K(i,i)$；
12. 交替计算系数矩阵 $M_1=M_1+\boldsymbol{x}_i^{(2)}K\boldsymbol{x}_i^{(2)\top}$ 和系数矩阵 $M_2=M_2+\boldsymbol{x}_i^{(2)}K\boldsymbol{x}_i^{(1)\top}$；
13. **end for**
14. 计算得到最优解 $M^*=M_1^{-1}M_2$；
15. 利用 $\tilde{\boldsymbol{w}}_{\pm}=(M^*)^{\top}\bar{\boldsymbol{w}}_{\pm}$ 将模型由 S_1 投影至 S_2；
16. 获取超平面 $\tilde{\boldsymbol{u}}_{\pm}=((\tilde{\boldsymbol{w}}_{1\pm})^{\mathsf{T}},\tilde{b}_{1\pm})$；
17. 同步骤 2-7，利用共现数据中新特征样本训练新的超平面 $\tilde{\boldsymbol{v}}_{\pm}=\left((\tilde{\boldsymbol{w}}_{2\pm})^{\mathsf{T}},\tilde{b}_{2\pm}\right)$；
18. **end if**
19. **end for**
20. **for** i=T_1+1,T_1+2,…,T_1+batch **do**
21. 接收样本 $X_{\pm\ i}^{(2)}$；
22. 同步骤 3-7，继续更新两个超平面 $\tilde{\boldsymbol{u}}_{\pm}^*$， $\tilde{\boldsymbol{v}}_{\pm}^*$；
23. **end for**
24. 分别调用算法 4-2 和 4-3，对样本 $X_{\pm}^{(2)}=\{(\boldsymbol{x}_{\pm}^{(2)},y_i)\}_{T_1+batch+1}^{T_2}$ 进行标签预测；

式(4-9)中高斯核权重 K 计算如式(4-10)所示：

$$K(i,i)=e^{-\frac{\left(\boldsymbol{x}_i^{(2)}-\boldsymbol{x}^{(2)}\right)^2}{2k^2}} \tag{4-10}$$

因此，线性映射关系的系数矩阵最优解 M^* 如式(4-11)所示：

$$M^*=\left(\sum_{i=T_i-B+1}^{T_i}\boldsymbol{x}_i^{(2)}K\boldsymbol{x}_i^{(2)\top}\right)^{-1}\left(\sum_{i=T_i-B+1}^{T_i}\boldsymbol{x}_i^{(2)}K\boldsymbol{x}_i^{(1)\top}\right) \tag{4-11}$$

计算得到该系数矩阵后，可以通过 $(M^*)^\top \boldsymbol{w}_\pm$ 将已训练好的模型参数投影至新特征空间下，从而对新到来样本继续预测。所提算法 ILFE 的伪代码具体如算法 4-1 所示。

4.2.2 两种集成策略

处理特征演化带来的新特征空间数据时，ILFE 算法通过拓展重用与重新训练得到两个基分类器，并对应产生两个预测结果。本节介绍两种新的集成方法：加权组合和选择最优，利用两个基分类器进行共同预测，保证模型整体分类性能高效稳定。

（1）加权组合预测

该方法基于当前逻辑损失计算分类器权重，通过加权组合的方法进行预测标签。两个基分类器预测值 $f_{1,i}$ 和 $f_{2,i}$ 如式(4-12)所示：

$$\begin{aligned} f_{1,i}&=\arg\min_{\pm}\frac{\left|(\boldsymbol{w}_{1\pm}^*)^\top\boldsymbol{x}+\tilde{b}_{1\pm}^*\right|}{\|\boldsymbol{w}_{1\pm}^*\|},\\ f_{2,i}&=\arg\min_{\pm}\frac{\left|(\boldsymbol{w}_{2\pm}^*)^\top\boldsymbol{x}+\tilde{b}_{2\pm}^*\right|}{\|\boldsymbol{w}_{2\pm}^*\|} \end{aligned} \tag{4-12}$$

第 i 轮的预测结果是对两个基分类器预测值 $f_{1,i}$ 和 $f_{2,i}$ 的加权平均值，如式(4-13)所示：

$$\hat{p}=\theta_{1,i}f_{1,i}+\theta_{2,i}f_{2,i} \tag{4-13}$$

其中，$\theta_{1,i}$ 和 $\theta_{2,i}$ 是两个基分类器预测值的权重。根据各个基分类器上一轮的损失值计算当前分类器权重，如式(4-14)所示：

$$\theta_{a,i+1}=\frac{\theta_{a,i}\left(1-\frac{1}{(1-\ell_i)+\varsigma}\right)}{\sum_{j=1}^{2}\theta_{j,i}\left(1-\frac{1}{(1-\ell_i)+\varsigma}\right)},a=1,2 \tag{4-14}$$

设置一个很小的常数项$\varsigma=0.001$，用来避免分母为零。本章将采用加权组合策略的算法称为 ILFE-c，如算法 4-2 所示。

算法 4-2 ILFE-c

输入: 测试样本 $\boldsymbol{X}_{\pm}^{(2)}=\{(\boldsymbol{x}_{\pm}^{(2)},y_i)\}_{T_1+batch+1}^{T_2}$，分类器 $\overline{\boldsymbol{u}}_{\pm}^{*}$ 和 $\overline{\boldsymbol{v}}_{\pm}^{*}$.

输出: 测试样本标签的预测结果.

1. **while** $i\le T_2$ **do**
2. 通过分类器 $\tilde{\boldsymbol{u}}_{\pm}^{*}=((\tilde{\boldsymbol{w}}_{1\pm}^{*})^{\mathsf{T}},\tilde{b}_{1\pm}^{*})$ 对 $X_{\pm}^{(2)}$ 进行标签预测得到 $f_{1,i}$；
3. 通过分类器 $\tilde{\boldsymbol{v}}_{\pm}^{*}=\left((\tilde{\boldsymbol{w}}_{2\pm}^{*})^{\top},\left(\tilde{b}_{2\pm}^{*}\right)^{\top}\right)$ 对 $\boldsymbol{X}_{\pm}^{(2)}$ 进行标签预测得到 $f_{2,i}$；
4. 计算预测结果 $\hat{p}$；
5. 揭示真实标签并计算当前损失 ℓ_i；
6. 更新分类器权重;
7. **end while**

（2） 选择最优预测

上述加权组合策略可通过集成多个基分类器预测结果以提升整体分类性能。该方法的前提假设是各个基分类器性能都不能低于某个阈值，这一假设在本章研究场景中并不总是成立，因为在特征演化初期分类器 $\tilde{\boldsymbol{v}}_{\pm}^{*}$ 性能较差，随着样本积累 $\tilde{\boldsymbol{u}}_{\pm}^{*}$ 可能逐渐不适合对当前样本分类。因此，提出另一种基于选择最优预测的策略，在本轮中挑选最优基分类器预测作为最终结果如式(4-15)：

$$\hat{p}=f_{a,i},a=1\text{或}2 \tag{4-15}$$

通过式(4-16)中基权重分布来决定挑选对应基分类器预测值：

$$\rho_{a,i+1}=\frac{\theta_{a,i}}{\sum_{j=1}^{2}\theta_{j,i}},a=1,2 \tag{4-16}$$

权重 $\nu_{1,i+1}$ 和 $\nu_{2,i+1}$ 的权重更新策略如式(4-17)所示：

$$\begin{aligned} \nu_{a,i+1} &= \theta_{a,i}\left(1-\frac{1}{(1-\ell_i)+\varsigma}\right), \\ \theta_{1,i+1} &= \mu\frac{V_i}{2}+(1-\mu)\theta_{1,i}, a=1,2 \end{aligned} \tag{4-17}$$

式(4-17)中V_i和μ的计算方法如式(4-18)所示：

$$\begin{aligned} V_i &= \nu_{1,i}+\nu_{2,i}, \\ \mu &= \frac{1}{T_2-batch-1} \end{aligned} \tag{4-18}$$

本章将采用挑选最优预测策略的算法称为 ILFE-s，如算法 4-3 所示。

算法 4-3 ILFE-s

输入: 测试样本 $\boldsymbol{X}_{\pm}^{(2)}=\{(\boldsymbol{x}_{\pm}^{(2)},y_i)\}_{T_1+batch+1}^{T_2}$, 分类器 $\tilde{\boldsymbol{u}}_{\pm}^{*}$ 和 $\tilde{\boldsymbol{v}}_{\pm}^{*}$.

输出: 测试样本标签的预测结果 $\hat{p}$.

1. **for** t=T_1+batch+1,T_1+batch+2,…,T_1+T_2 **do**
2. 通过分类器 $\tilde{\boldsymbol{u}}_{\pm}^{*}=((\tilde{\boldsymbol{w}}_{1\pm}^{*})^{\mathsf{T}},\tilde{b}_{1\pm}^{*})$ 对 $X_{\pm}^{(2)}$ 进行标签预测得到 $f_{1,i}$；
3. 通过分类器 $\tilde{\boldsymbol{v}}_{\pm}^{*}=\left((\tilde{\boldsymbol{w}}_{2\pm}^{*})^{\mathsf{T}},\left(\tilde{b}_{2\pm}^{*}\right)^{\mathsf{T}}\right)$ 对 $\boldsymbol{X}_{\pm}^{(2)}$ 进行标签预测得到 $f_{2,i}$；
4. 计算预测结果 $\hat{p}$；
5. 揭示真实标签并计算当前损失 ℓ_i；
6. 更新分类器权重;
7. **end for**

4.3 实验设计与结果分析

4.3.1 数据集与实验设置

实验选用来自不同领域的 10 个 UCI 数据集（所有数据集均可在 http://archive.ics.uci.edu 进行下载），通过采取和文献[47]相同处理方法，生成人工数据以模拟特征演化数据流。

具体来说，原始数据集最初仅有一个特征空间，借助随机高斯矩阵将原始数据集人工映射到新的特征空间下，并在学习过程中逐个读取使数据依次到来，那么将获得来自两个不同的特征空间 S_1 和 S_2 的数据流，即特征演化数据流。合成的人工数据集的详细信息如表 4-1 所示，其中 d_1 表示原始数据集的特征空间 S_1 的维度，d_2 表示人工生成的新特征空间 S_2 的维度。

表 4-1 数据集描述
Table 4-1 Detailed Description Of Datasets

数据集	实例数量	d_1	d_2
Australian	690	42	29
Credit-a	653	15	10
Credit-g	1000	20	14
Diabetes	768	8	5
DNA12	940	180	125
German	1000	59	41
Kr-vs-kp	3196	36	25
Svmguide3	1284	22	15
Magic04	19020	10	7
HTRU_2	17898	8	5

将所提算法 ILFE-c、ILFE-s 与文献[47]中所提算法 FESL-c、FESL-s 以及其过程中涉及的算法 NOGD、ROGD-f 和 ROGD-u，通过实验评估所提算法在测试数据上的分类性能。同时对比算法还包括算法 NX、RX-u 进行对比，其中 NX 是指从头调用 FBTWSVM 算法对共现数据进行训练，并对新特征空间下数据的预测；RX-u 是指通过线性映射，将新特征空间下的数据恢复至旧特征空间，利用 FBTWSVM 继续更新已学到的模型进行预测。

为保证实验对比的公平性，对比算法 NOGD、ROGD-u、FESL-s 以及 FESL-c 的参数设置中，对于 HTRU_2 数据集和 Magic04 数据集，经过调优后参数 C 均设置为 200，其他数据集的参数 C 的选取保留文献[47]中的设置。所有算法的分类精度是通过 10 次独立重复实验，每次重复随机打乱数据顺序，最后预测结果取 10 次结果平均值得到。参数敏感性实验中，研究参数 $c_i(i=1,2,3,4)$ 对分类准确率的影响，其中 $c_1=c_3\in\{2^1,2^2,2^3,2^4\}$， $c_2=c_4\in\{2^0,2^1,2^2,2^3\}$。

4.3.2 无噪声数据集上对比实验

为验证所提算法的可行性，采用上述 8 种算法在 10 个不同不含噪声的人工数据集上进行实验，实验结果如表 4-2 所示。其中，通过分组与整体对比不同算法性能，并用粗体来标注每个网格中较好的实验结果，用粗体加下划线的组合标记采取集成策略的算法中整体最优预测结果。

算法 ILFE-c 和 ILFE-s 在其中 9 个数据集上的准确率均优于其他对比算法。尤其在 Diabetes、Kr-vs-kp 和 Magic04 数据集上，算法 ILFE 比 FESL 分别高出

7.9%、9.1%和 21.7%；在除 Credit-g 外的其他数据集上，ILFE 算法实验结果高出约 0.3%~2.3%。这表明在不同数量级的数据集中 ILFE 算法都能取得不错的分类效果，说明 ILFE 算法较好地适应特征演化场景，具有现实应用价值。

实验结果表明，从各自对应网格中可以看出，NX 算法在 10 个数据集上整体表现都优于 NOGD 算法，在 Australian、Kr-vs-kp 数据集上准确率甚至分别高出 42.0%和 33.6%；而在 Credit-a、Credit-g、German 和 Svmguide3 数据集上，均高出 11.0%以上，RX-u 算法在 8 个数据集上相对于 ROGD-u 也表现出分类优势。再次体现 ILFE 对于特征演化数据流的学习效果整体有明显的提升。

表 4-2 无噪声人工数据集上各对比算法准确率

Table 4-2 Accuracy Of The Compared Algorithms On Synthetic Datasets Without Noise

Dataset	Australian	Credit-a	Credit-g	Diabetes	DNA12
NOGD	0.592±0.034	0.670±0.040	0.597±0.025	0.623±0.038	0.509±0.035
NX	**0.868±0.021**	**0.845±0.028**	**0.720±0.028**	**0.710±0.029**	**0.890±0.014**
ROGD-u	0.860±0.028	0.837±0.022	**0.732±0.025**	0.625±0.030	**0.890±0.014**
RX-u	**0.869±0.023**	**0.847±0.026**	0.718±0.030	**0.700±0.038**	0.890±0.026
FESL-c	0.850±0.030	0.824±0.025	0.732±0.025	0.633±0.032	0.885±0.017
FESL-s	0.842±0.033	0.811±0.022	**0.733±0.023**	0.623±0.035	0.888±0.013
ILFE-c	0.865±0.023	0.845±0.027	0.719±0.028	0.707±0.033	0.889±0.023
ILFE-s	**0.869±0.022**	**0.847±0.026**	0.721±0.029	**0.712±0.033**	**0.894±0.020**
Dataset	German	Kr-vs-kp	Svmguide3	HTRU_2	Magic04
NOGD	0.599±0.023	0.531±0.018	0.615±0.024	0.952±0.002	0.515±0.271
NX	**0.717±0.023**	**0.867±0.016**	**0.779±0.015**	**0.969±0.004**	**0.758±0.021**
ROGD-u	0.706±0.017	0.781±0.016	0.770±0.014	0.968±0.002	0.683±0.004
RX-u	**0.715±0.015**	**0.870±0.014**	**0.776±0.017**	**0.970±0.005**	**0.757±0.018**
FESL-c	0.706±0.017	0.779±0.015	0.768±0.013	0.968±0.002	0.547±0.289
FESL-s	0.701±0.020	0.769±0.022	0.770±0.014	0.966±0.005	0.547±0.288
ILFE-c	0.719±0.013	0.869±0.014	0.777±0.017	**0.971±0.004**	**0.764±0.015**
ILFE-s	**0.723±0.012**	**0.870±0.014**	**0.779±0.016**	**0.971±0.004**	0.763±0.015

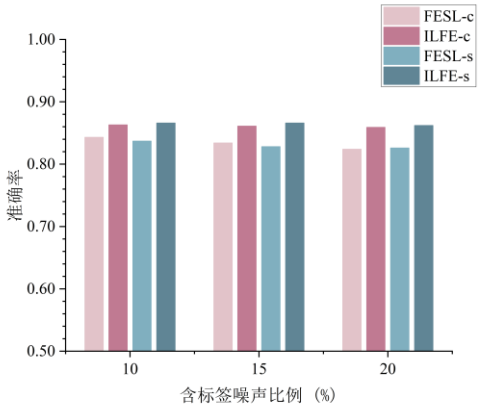
4.3.3 含噪声数据集上对比实验

为验证所提算法的抗噪声能力，将算法 ILFE-c、ILFE-s、FESL-c、FESL-s 部署在不同比例的标签噪声和属性噪声环境中，以分类准确率作为评价标准进行实验，实验结果分别如图 4-2 和图 4-3 所示。

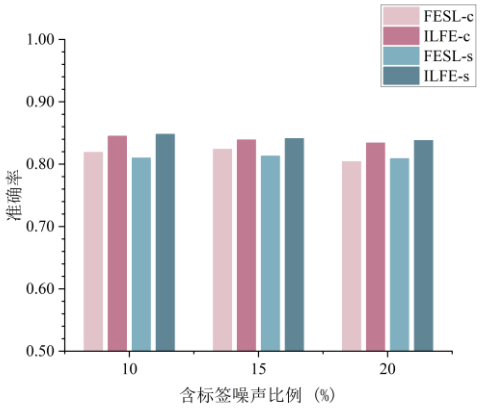
图 4-2 展示分别含 10%、15%和 20%的标签噪声数据集中，对比算法的分类结果。通过分析该柱状图可以看出，在 Australian、Credit-a、Diabetes 和

Svmguide3 这 4 个数据集上，整体 ILFE 系列算法平均准确率高于 FESL 系列算法。其中，在 Diabetes 数据集上这一优势表现更为明显。同时观察发现，随着标签噪声比例的增大，所有算法准确率都会有不同程度的下降，但 ILFE 算法相比于 FESL-c 和 FESL-s 下降相对缓慢，甚至在 Australian 数据集上基本呈现平稳趋势。进一步说明，对于给算法分类性能带来更大消极影响的标签噪声，ILFE 方法具有较好的健壮性。

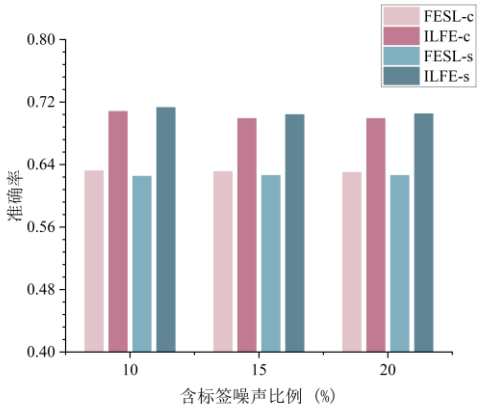
图 4-3 展示分别含 20%、40%和 60%的属性噪声数据集中，不同对比算法的分类准确率。比较在 Credit-g、DNA12、German 和 Kr-vs-kp 这 4 个数据集上的实验结果，发现随着噪声比例的增大，整体算法性能都会下降，但由于设置的属性噪声干扰较大，有时甚至会出现准确率先减后增、先增后减这样的情况。但是在有属性噪声干扰的环境中，所提算法的学习性能仍普遍高于 FESL-c 和 FESL-s，这表明 ILFE-c 和 ILFE-s 能够有效处理数据流中的属性噪声。



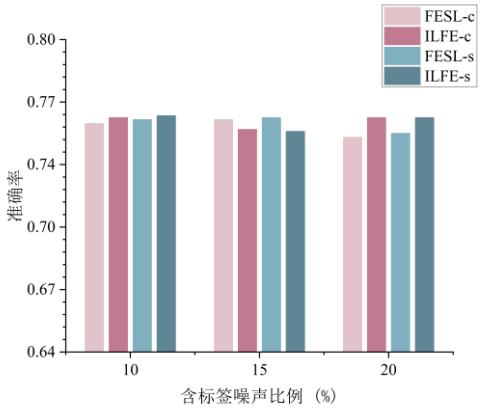
（a）Australian 数据集



（b）Credit-a 数据集



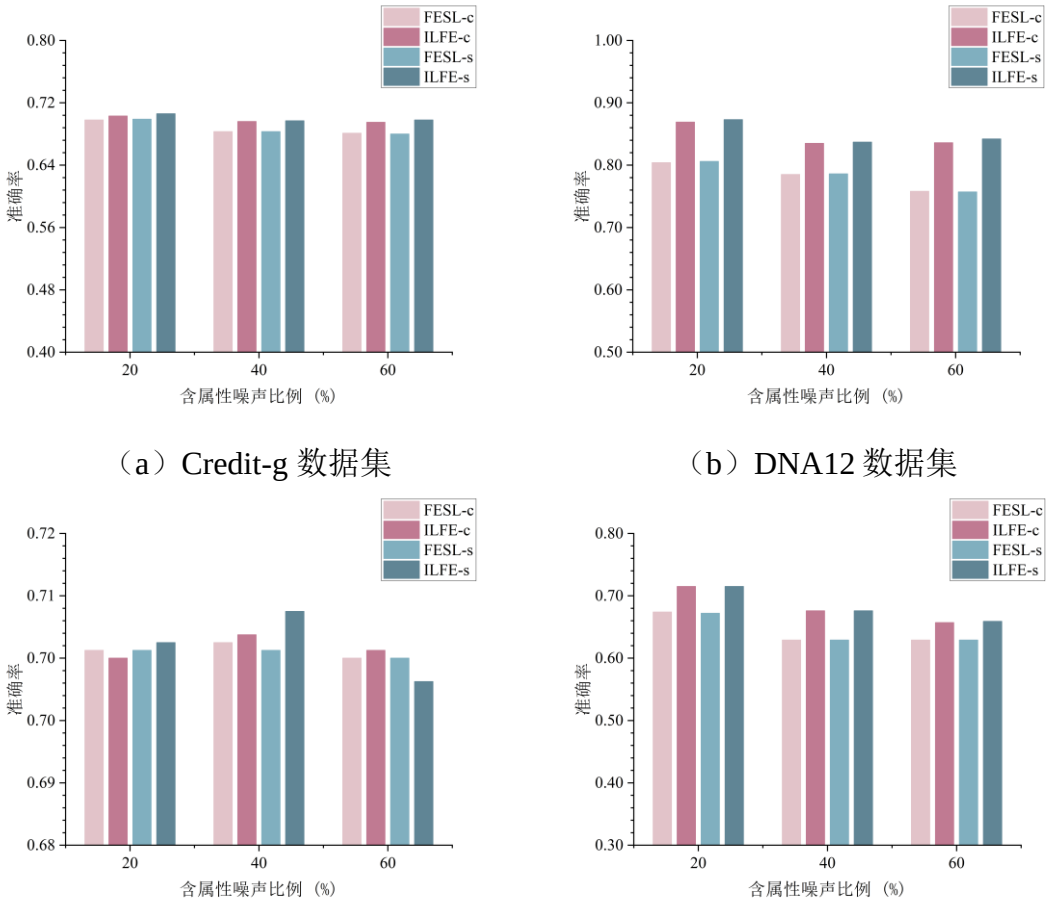
（c）Diabetes 数据集



（d）Svmguide3 数据集

图 4-2 不同比例标签噪声环境下的实验结果

Fig. 4-2 Experimental results under different proportions of label noise



（a）Credit-g 数据集　　（b）DNA12 数据集

（c）German 数据集　　（d）Kr-vs-kp 数据集

图 4-3 不同比例属性噪声环境下的实验结果

Fig. 4-3 Experimental results under different proportions of attribute noise

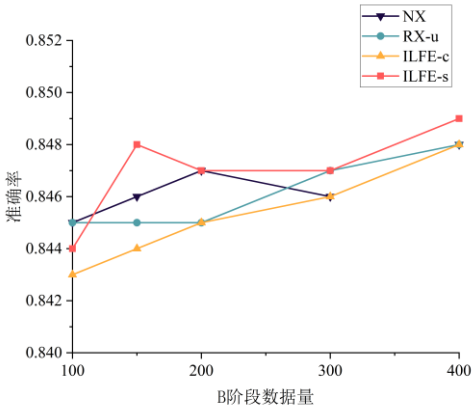
4.3.4　B 阶段不同数据量的对比实验

为观察不同的新旧特征共存期，即共现数据出现的阶段 B 长度，对算法分类性能的影响，在无噪声干扰的 4 个不同人工数据集中进行实验，分别设置在 B=100,200,300,400 中对比所提算法相关的 4 个算法，实验结果如图 4-4 所示。

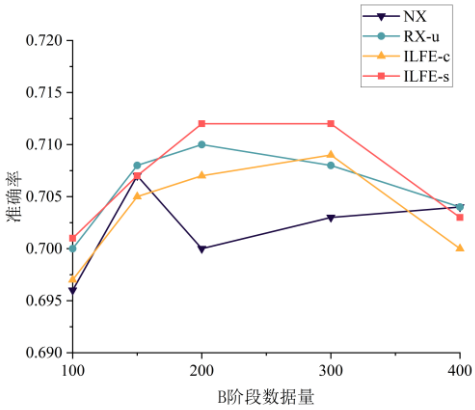
通过观察折线图，可以发现 ILFE-s 在不同数据量的 B 阶段数据集中平均分类效果较好。而 ILFE-c 在 B 值较小的情况下分类性能较差，这主要是受到该算法受到基分类算法 NX 的影响。对比两个基分类算法 NX 和 RX-u，B 阶段数据量较小时，RX-u 整体表现优于 NX。在现实应用平台中 B 值一般不大，即新旧特征重叠期短，恰恰说明将旧的分类模型投影至新特征下进行重用是有必要的。当然，随着 B 阶段数据量增大，投影的旧模型性能会逐渐下降，而 NX 重新训练的分类模型性能会得到提升，其原因还是用来训练的新特征数据量增多，这也是十分合理的。并且基本在 B 取 300 时，ILFE 算法能够稳中有升地表

现出良好的预测效果。

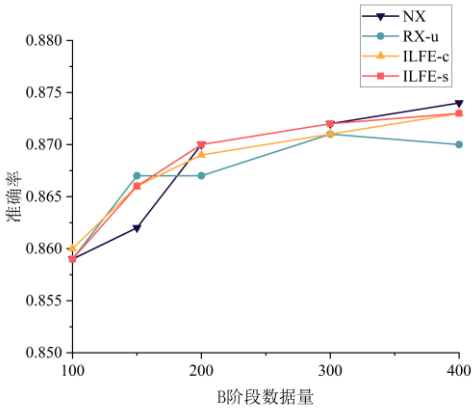
基于上述可知，B 阶段中数据量的多少会对分类性能带来影响，总体上两个基分类算法 NX 和 RX-u 的性能表现相对较差，而 ILFE-c 和 ILFE-s 能够得到更高的分类准确率，说明仅通过重新训练分类器或历史分类器重用的方法都无法保证分类模型的稳定性，从而验证了采用集成策略进行组合预测的有效性。



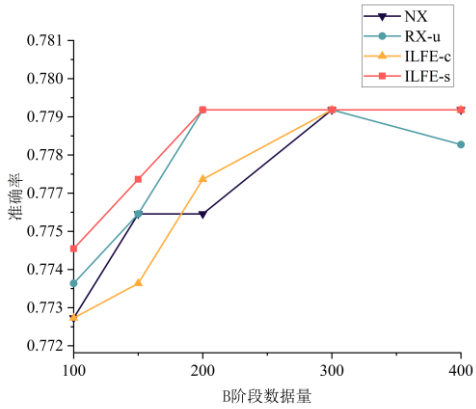
（a）Credit-a 数据集



（b）Diabetes 数据集



（c）Kr-vs-kp 数据集



（d）Svmguide3 数据集

图 4-4 不同 B 值的对比实验

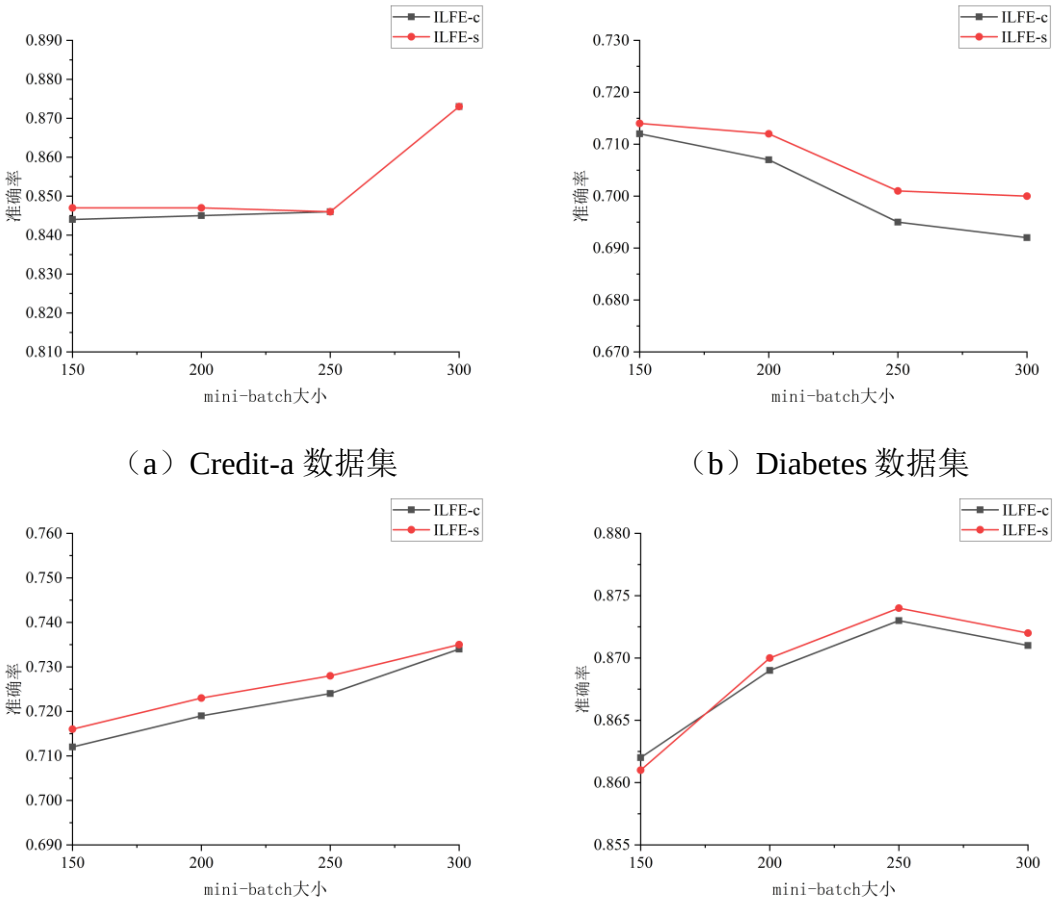
Fig. 4-4 Experimental results with different B values

4.3.5 不同的 mini-batch 大小的对比实验

为进一步分析 mini-batch 的大小对 ILFE-c 和 ILFE-s 算法的影响。本节设置 mini-batch 在 $\{150, 200, 250, 300\}$ 取值范围中，分别部署两个算法在 Credit-a、Diabetes、German 和 Kr-vs-kp 人工数据集上进行对比实验，结果如图 4-5 所示。

在 4 个数据集的实验中 ILFE-s 的平均准确率稍优于 ILFE-c。通过比较不同数据集上折线变化趋势，除 Diabetes 数据集的其他数据集上，两个算法的预测

准确度一般随着 mini-batch 的增大而增大，但也会出现先增后减甚至一直递减的现象，说明数据批量较小可能导致分类性能不稳定，批量较大则会影响模型收敛速度。因此，为平衡训练效率与分类准确率，针对不同数量的数据集仍需选取合适的 mini-batch 的大小。



（a）Credit-a 数据集　　（b）Diabetes 数据集

（c）German 数据集　　（d）Kr-vs-kp 数据集

图 4-5 不同 mini-batch 大小的对比实验

Fig. 4-5 Experimental results with different sizes of mini-batch

4.4 本章小结

本章提出面向特征演化流的集成学习方法 ILFE。首先，利用模糊隶属度函数区分噪声与支持向量，增量训练与更新孪生支持向量机分类模型；然后，新特征出现时，利用对应阶段收集的共现数据学习新旧特征之间的映射函数，同时重新训练一个新分类器；同时，在旧特征消失时，通过特征映射的方法实现模型投影至新特征空间，并提出两个新的集成方法：加权组合预测和选择最优预测。最后，大量实验验证所提方法具有对特征演化数据流自适应学习能力，同时能减少噪声对分类性能的干扰。

第5章 具有类不平衡的特征演化数据流在线分类方法

数据流挖掘任务面临诸多挑战：如数据实时处理、数据类别分布不平衡、分类模型与当前数据特征不匹配而失效等。传统的批处理学习方法在学习数据流时往往面临内存不足的问题，同时无法及时利用最新样本对模型增量更新。在线学习是进行数据流挖掘的有效工具之一[85]。在线学习方法对随时间依次产生的数据逐个学习，每轮训练获取损失信息用来优化模型，随后释放内存不再重复读取数据，因此适用于海量高速的数据流学习。

近年来，国内外对特征演化流的在线学习展开研究，针对特征空间动态变化的特点训练在线分类模型，并取得一系列研究成果。Hou 等[47]提出的 FESL 算法利用在线梯度下降算法对模型进行更新，结合集成策略实现特征演化流的学习，但该算法仅考虑瞬时损失函数的负梯度方向进行模型更新。为此，Liu 等[48]提出一种基于被动攻击更新策略的特征演化学习方法，该方法采用一阶在线学习算法加快模型收敛速度和提升分类性能。随后在文献[49]中提出特征加权的置信-加权学习算法，进一步挖掘数据特征相关性和二阶信息提升分类准确性，但也带来较高的时空复杂度。

数据流中的类不平衡问题同样亟待关注。如在疾病监测系统中，患者作为少数类样本更需被关注，但受到训练样本数量的影响，分类器更倾向于学习多数类样本，这导致少数类样本的分类错误率上升。传统类不平衡学习方法需在完整的特征空间下运行，而不适用于特征演化数据流的分类任务。目前鲜有在线学习框架同时处理数据流中特征演化和类不平衡问题，文献[86]提出面向不完备和不平衡数据流的在线学习（Online Learning for Incomplete and Imbalanced Data Streams，OLIDS），遵循经验风险最小化原则来识别不完备特征中信息量最大的特征，并通过将 F-measure 优化转化为加权代理损失最小化，开发动态成本策略来实时处理不平衡的类别分布。受此启发，本章提出具有类不平衡的特征演化数据流在线学习方法，首先通过构建全局特征空间与特征划分，进行特征演化自适应学习；为处理类不平衡问题，重新定义代价敏感因子以降低对类不平衡数据流的误分类代价，并通过截断稀疏技术保证模型的泛化能力。

5.1 问题描述与分析

本章主要研究数据流在线二分类任务，并对文献[56]中描述的特征演化场景进行拓展：所有出现过的已有特征随机消失或出现，同时未知的新增特征随时间不断到来。图 5-1 呈现的是类不平衡的特征演化流场景，具体描述如下：

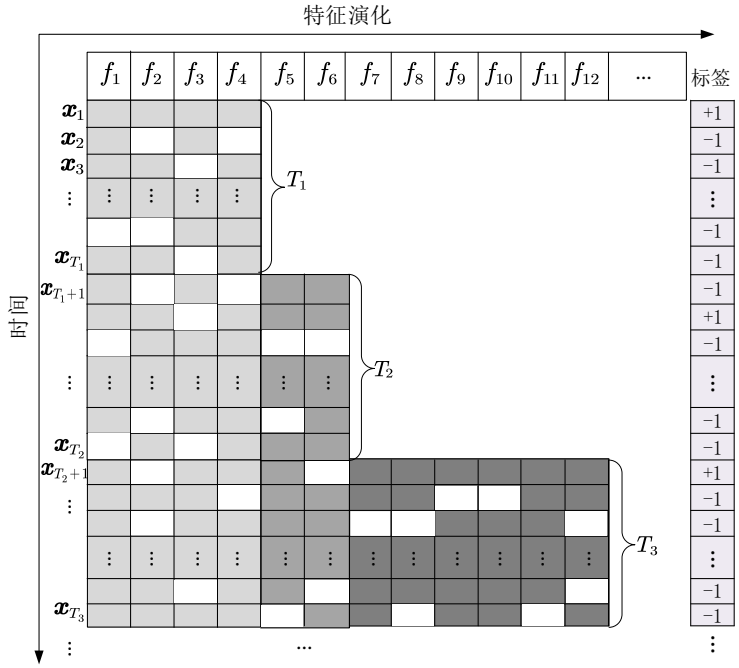


图 5-1 类不平衡的特征演化流

Fig. 5-1 Class imbalanced and feature evolvable streams

利用 $S=\{(\boldsymbol{x}_t,y_t)\,|\,t=1,2,\ldots,T\}$ 来表示数据流，$\boldsymbol{x}_t\in\mathbb{R}^{d_t}$ 表示 t 时刻接收的样本实例，d_t 为实例特征维度，$y_t\in\{-1,1\}$。在每一轮的学习中，分类器 $\boldsymbol{w}_t\in\mathbb{R}^{d_t}$ 会接收到一个样本实例并给出预测结果 $\hat{y}_t=\mathrm{sign}\left(\boldsymbol{x}_t\cdot\boldsymbol{w}_t\right)$，一旦给出预测结果，则揭示该样本的真实标签 $\boldsymbol{w}_t^m$，进而得到分类器的瞬时损失 $\ell_t=\max\{0,1-y_t(\boldsymbol{x}_t\cdot\boldsymbol{w}_t)\}$，该损失反映预测标签与真实标签之间的差异，根据该损失信息，算法在线更新分类模型，以便在下一轮中做出更好的预测。

（1）由目前所有已出现的特征构成样本的全局特征空间 $\mathcal{U}_t=\{\mathbb{R}^{d_1}\cup\mathbb{R}^{d_2}\cup\cdots\cup\mathbb{R}^{d_t}\}$，例如当 $t=\mathrm{T}_1$ 时，全局特征空间由 $\mathcal{U}_{\mathrm{T}_1}=\{f_1,f_2,f_3,f_4\}$ 构成，同理，当 $t=\mathrm{T}_2$ 时，样本的全局特征空间扩展为 $\mathcal{U}_{\mathrm{T}_2}=\{f_1,f_2,\cdots,f_6\}$，而当 $t=\mathrm{T}_3$ 时进一步扩展为 $\mathcal{U}_{\mathrm{T}_3}=\{f_1,f_2,\cdots,f_{12}\}$。特征演化过程中，全局特征空间逐渐加入新的特征而不断扩大；

（2）在所有阶段内特征均会在某一轮消失而在某一时刻重新出现，将目前从未出现在$\mathcal{U}_t$中的特征归为新增特征，而实例$\boldsymbol{x}_t$和$\mathcal{U}_t$同时具备的特征归为共享特征，依此原则对实例特征进行划分：以$\boldsymbol{x}_{\mathrm{T}_1+1}$为例，与$\mathrm{T}_1$时刻的全局特征空间$\mathcal{U}_{\mathrm{T}_1}$进行对比，可将实例$\boldsymbol{x}_{\mathrm{T}_1+1}$的特征空间划分为共享特征空间$\mathcal{F}^{\mathrm{s}}_{\mathrm{T}_1+1}=\{f_1, f_3\}$和新增特征空间$\mathcal{F}^{\mathrm{n}}_{\mathrm{T}_1+1}=\{f_5, f_6\}$；

（3）样本的类别存在不平衡，“+1”代表少数类，“-1”代表多数类，对应$\mathrm{Num_p}$与$\mathrm{Num_n}$分别表示少数类和多数类的实例数量，使用不平衡率$\mathrm{IR}=\mathrm{Num_n}/\mathrm{Num_p}$表示数据流中类别分不平衡程度。

通过研究特征演化特点，分析得到学习类不平衡的特征演化流时需解决的两个关键问题：（1）如何充分保留历史知识以辅助模型更新，仅根据当前实例特征对特征演化流进行学习；（2）如何将类别误分类代价最小化的目标和在线学习目标相结合，实时调整分类器权重以适应类不平衡的学习；

5.2 类不平衡的特征演化数据流在线分类模型

基于上述讨论，提出面向类不平衡的特征演化数据流在线学习方法 OIFE，总体框架如图 5-2 所示：

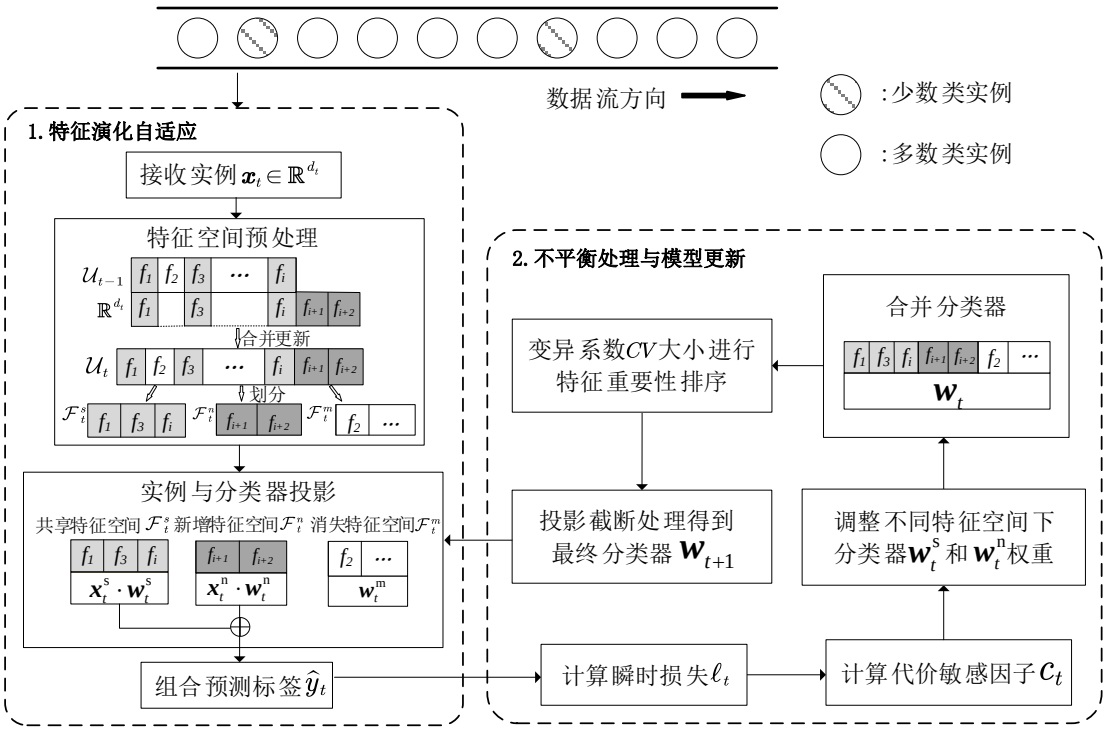


图 5-2 OIFE 总体框架图

Fig. 5-2 The framework of OIFE

OIFE 方法分为以下两个阶段，分别对应解决上述两个关键问题：

第 1 阶段为特征演化自适应阶段。首先接收实例 $\boldsymbol{x}_t \in \mathbb{R}^{d_t}$，将上一轮全局特征空间 $\mathcal{U}_{t-1}$ 与当前实例的特征空间进行对比，同时存在于 $\mathcal{U}_{t-1}$ 和 $\mathbb{R}^{d_t}$ 的特征划分至共享特征空间 $\mathcal{F}_t^{\mathrm{s}}$，只属于当前实例而不存在于全局特征空间的特征则划分至新增特征空间 $\mathcal{F}_t^{\mathrm{n}}$，其余部分特征则归属于消失特征空间 $\mathcal{F}_t^{\mathrm{m}}$，接着 $\mathcal{U}_{t-1}$ 和 $\mathbb{R}^{d_t}$ 合并后得到当前全局特征空间 $\mathcal{U}_t$。当前实例和分类器分别被投影至两个特征空间下，得到 $\boldsymbol{x}_t^{\mathrm{s}}$、$\boldsymbol{x}_t^{\mathrm{n}}$、$\boldsymbol{w}_t^{\mathrm{s}}$ 和 $\boldsymbol{w}_t^{\mathrm{n}}$，并将分类器剩余部分保存在 $\boldsymbol{w}_t^{\mathrm{m}}$ 中，便于组合预测实例标签以及合并分类器；

第 2 阶段为不平衡处理与模型更新阶段。揭示真实标签后 y_t 后，可计算瞬时损失以及多数类和少数类实例数量，由此计算代价敏感因子并对分类器 $\boldsymbol{w}_t^{\mathrm{s}}$ 和 $\boldsymbol{w}_t^{\mathrm{n}}$ 进行权重更新，以提高对不平衡数据流的分类准确率；最后，通过计算每个特征的变异系数 CV 值来对特征重要性进行排序，保留最重要的一定数量的特征使模型稀疏化处理，从而保障高效稳定的分类性能。

5.2.1 模型训练与更新

数据实例逐个到来时，每轮数据特征空间维度都可能发生改变，当前数据特征空间与上一轮更新学习到的模型特征维度不匹配，因此无法预测实例标签。考虑到在线分类模型只能利用与实例共享的特征进行更新，新增特征无法提供任何先验知识，而消失特征对当前实例预测没有帮助。

因此，为适应特征空间的动态改变，利用特征划分的方法。首先，接收到当前数据实例后，对比当前实例 $\boldsymbol{x}_t$ 所有特征和上一轮全局特征空间 $\mathcal{U}_{t-1}$，将新出现的特征并入当前全局特征空间 $\mathcal{U}_t$，并计算其中所有特征的平均值、方差和标准差等统计信息。根据比较结果可将当前实例 $\boldsymbol{x}_t$ 的特征空间划分为：共享特征空间 $\mathcal{F}_t^{\mathrm{s}} \in \mathbb{R}^{d_t^{\mathrm{s}}}$ 和新增特征空间 $\mathcal{F}_t^{\mathrm{n}} \in \mathbb{R}^{d_t^{\mathrm{n}}}$，其中 d_t^{s} 和 d_t^{n} 分别是 $\mathcal{F}_t^{\mathrm{s}}$ 和 $\mathcal{F}_t^{\mathrm{n}}$ 中的特征个数。然后，需要将当前实例 $\boldsymbol{x}_t$ 和分类器 $\boldsymbol{w}_t$ 各自投影至共享特征空间 $\mathcal{F}_t^{\mathrm{s}}$ 与新增特征空间 $\mathcal{F}_t^{\mathrm{n}}$ 进行训练与更新。OIFE 整体实现的伪代码如算法 5-1 所示。

算法 5-1 OIFE

输入：权衡参数 C、正则化参数 λ、选择特征比例 B 和动态代价伸缩因子 θ.

输出：更新后的分类器 $\boldsymbol{w}_{t+1}$.

初始化分类器权重 $\boldsymbol{w}_1 = (0,0,\ldots,0) \in \mathbb{R}^{d_i}$；
for t=1,2,…,T_1+T_2+T_3 **do**
  接收当前实例 $\boldsymbol{x}_t \in \mathbb{R}^{d_t}$；
  for $f_i \in \mathbb{R}^{d_i}, i = 1,2,\cdots,d_t$ **do** //遍历 $\boldsymbol{x}_t$ 的所有特征，与全局特征空间 $\mathcal{U}_{t-1}$ 进行对比
    $\mathcal{U}_t \leftarrow \mathcal{U}_{t-1} \cup \{f_i\}$；　　//将 $\boldsymbol{x}_t$ 对应特征并入当前全局特征空间 $\mathcal{U}_t$
  计算 $\mathcal{U}_t$ 中所有特征的平均值、方差和标准差等统计信息；
    if $f_i \in \mathcal{U}_{t-1}$ **then**　　//划分出共享特征空间 $\mathcal{F}_t^{\mathrm{s}}$ 和新增特征空间 $\mathcal{F}_t^{\mathrm{n}}$
      $\mathcal{F}_t^{\mathrm{s}} \leftarrow \mathcal{F}_t^{\mathrm{s}} \cup \{f_i\}$；
    else
      $\mathcal{F}_t^{\mathrm{n}} \leftarrow \mathcal{F}_t^{\mathrm{n}} \cup \{f_i\}$；
    end if
  $\boldsymbol{x}_t^{\mathrm{s}} = \boldsymbol{x}_t \cap \mathcal{F}_t^{\mathrm{s}}, \boldsymbol{x}_t^{\mathrm{n}} = \boldsymbol{x}_t \cap \mathcal{F}_t^{\mathrm{n}}$；
  $\boldsymbol{w}_t^{\mathrm{s}} = \boldsymbol{w}_t \cap \mathcal{F}_t^{\mathrm{s}}, \boldsymbol{w}_t^{\mathrm{n}} = \boldsymbol{w}_t \cap \mathcal{F}_t^{\mathrm{n}}$；
  $\boldsymbol{w}_t^{\mathrm{m}} = \boldsymbol{w}_t - (\boldsymbol{w}_t^{\mathrm{s}} \cup \boldsymbol{w}_t^{\mathrm{n}})$；　//将当前实例 $\boldsymbol{x}_t$、分类器 $\boldsymbol{w}_t$ 投影至共享特征空间与新增特征空间
  end for
  调用算法 5-2 进行类不平衡处理；
  调用算法 5-3 进行模型截断稀疏；
end for

5.2.2 类不平衡的处理

通过改进在线 PA 算法以构建适应动态特征演化数据流的线性分类模型。具体来说，OIFE 始终维护一个全局特征空间下的线性分类器，每一轮分别将当前实例 $\boldsymbol{x}_t$ 投影至共享特征空间 $\mathcal{F}_t^{\mathrm{s}}$ 和新增特征空间 $\mathcal{F}_t^{\mathrm{n}}$ 得到 $\boldsymbol{x}_t^{\mathrm{s}}$ 和 $\boldsymbol{x}_t^{\mathrm{n}}$，同样地，分类器 $\boldsymbol{w}_t$ 投影后得到 $\boldsymbol{w}_t^{\mathrm{s}}$ 和 $\boldsymbol{w}_t^{\mathrm{n}}$，分类器剩余部分保留为 $\boldsymbol{w}_t^{\mathrm{m}}$。然后，根据式(5-1)中特征演化流的在线学习目标函数进行模型局部在线更新：

$$\begin{aligned} \boldsymbol{w}_t = & \underset{\boldsymbol{w}_t=[\boldsymbol{w}^m,\boldsymbol{w}^s,\boldsymbol{w}^n]}{arg\min} \frac{1}{2}\| \boldsymbol{w}^{\mathrm{m}} - \boldsymbol{w}_t^{\mathrm{m}}\|^2 \\ & + \frac{1}{2}\| \boldsymbol{w}^{\mathrm{s}} - \boldsymbol{w}_t^{\mathrm{s}}\|^2 + \frac{1}{2}\| \boldsymbol{w}^{\mathrm{n}}\|^2 + C\xi \\ & s.t.\ \ell_t \leqslant \xi,\ \xi \geqslant 0 \end{aligned} \tag{5-1}$$

采用改进的铰链损失函数计算得到损失值 ℓ_t，如式(5-2)所示：

$$\ell_t = \max\left\{0, 1 - y_t\left(\boldsymbol{x}_t^{\mathrm{s}} \cdot \boldsymbol{w}_t^{\mathrm{s}} + \boldsymbol{x}_t^{\mathrm{n}} \cdot \boldsymbol{w}_t^{\mathrm{n}}\right)\right\} \tag{5-2}$$

考虑到数据流中还可能存在类别分布不平衡，可将最小化误分类代价的目标融入在线分类模型更新过程中。采用文献[45]中定义的误分类代价 cost，如式(5-3)：

$$\text{cost}=\text{c}_\text{p}\times\text{FP}+\text{c}_\text{n}\times\text{FN} \tag{5-3}$$

其中，c_p 和 c_n 分别为少数类和多数类的误分类代价权重且 $\text{c}_\text{p}>\text{c}_\text{n}$，FP 和 FN 分别代表将少数类和多数类错误分类的个数，代价敏感指标 cost 越低，分类性能越好。那么，最小化 cost 等同于最小化式(5-4)中目标函数：

$$\min \text{c}_\text{p}\sum_{y_t=+1} I\left(\hat{y}_t=\text{-1}\right)+\text{c}_\text{n}\sum_{y_t=-1} I\left(\hat{y}_t=+1\right) \tag{5-4}$$

式(5-4)给出明确的优化目标函数，但指标函数不是凸函数。为便于在线优化任务，将指示函数替换为其凸代理函数，即铰链损失函数，目标函数重新表述如式(5-5)所示：

$$\begin{aligned}&\min \text{c}_\text{p}\sum_{y_t=+1}\ell_t\left(\boldsymbol{w}_t;(\boldsymbol{x}_t,y_t)\right)+\text{c}_\text{n}\sum_{y_t=-1}\ell_t\left(\boldsymbol{w}_t;(\boldsymbol{x}_t,y_t)\right)\\&=\min \text{c}_t\sum_{y=\pm 1}\ell_t\left(\boldsymbol{w}_t;(\boldsymbol{x}_t,y_t)\right)\end{aligned} \tag{5-5}$$

由此，定义 c_t 为代价敏感因子，表示本轮训练实例的误分类代价。获得真实标签后可对 c_t 进行计算，如式(5-6)所示：

$$\text{c}_t=\left(\frac{\text{1-}\alpha}{\alpha}\times\text{IR}\right)^{I(y_t=+1)}+\left(\frac{\alpha}{\text{1-}\alpha}\times\frac{1}{\text{IR}}\right)^{(1-I(y_t=+1))} \tag{5-6}$$

其中，参数 $\alpha\in(0,0.5)$ 用以调节学习收敛速度，不平衡率 $\text{IR}=\text{Num}_\text{n}/\text{Num}_\text{p}$。

在线学习特征演化流时，将代价敏感 c_t 引入式（5-1）所示在线学习的约束条件中，计算并调整对当前实例 $\boldsymbol{x}_t$ 的误分类代价，从而实时处理数据流中类不平衡问题，得到新的目标函数如式(5-7)所示：

$$\begin{aligned}\boldsymbol{w}_t=&\underset{\boldsymbol{w}_t=[\boldsymbol{w}^m,\boldsymbol{w}^s,\boldsymbol{w}^n]}{arg\min}\frac{1}{2}\|\boldsymbol{w}^\text{m}-\boldsymbol{w}_t^\text{m}\|^2\\&+\frac{1}{2}\|\boldsymbol{w}^\text{s}-\boldsymbol{w}_t^\text{s}\|^2+\frac{1}{2}\|\boldsymbol{w}^\text{n}\|^2+C\xi\\&s.t.\ c_t\ell_t\leqslant\xi,\ \xi\geqslant 0\end{aligned} \tag{5-7}$$

为求解上式中含不等式约束的优化问题时，结合拉格朗日函数和 KKT 条件，设 τ 和 δ 为拉格朗日乘子，计算如式(5-8)所示：

$$L(\boldsymbol{w}_t,\xi,\tau,\delta)=\frac{1}{2}\|\boldsymbol{w}^{\mathrm{m}}-\boldsymbol{w}_t^{\mathrm{m}}\|^2+\frac{1}{2}\|\boldsymbol{w}^{\mathrm{s}}-\boldsymbol{w}_t^{\mathrm{s}}\|^2+\frac{1}{2}\|\boldsymbol{w}^{\mathrm{n}}\|^2+C\xi+\tau(\ell_t-\xi)-\delta\xi \tag{5-8}$$

对 $\boldsymbol{w}^m$、$\boldsymbol{w}^s$、$\boldsymbol{w}^n$ 和 δ 求偏导以求解如式(5-9)所示：

$$\begin{aligned}\boldsymbol{w}^{\mathrm{m}}&=\boldsymbol{w}_t^{\mathrm{m}}\\ \boldsymbol{w}^{\mathrm{s}}&=\boldsymbol{w}_t^{\mathrm{s}}+\tau c_t y_t \boldsymbol{x}_t^{\mathrm{s}}\\ \boldsymbol{w}^{\mathrm{n}}&=\tau c_t y_t \boldsymbol{x}_t^{\mathrm{n}}\\ \delta&=C-\tau\end{aligned} \tag{5-9}$$

将式(5-9)代入式(5-8)中可得：

$$L(\tau)=\frac{\tau^2}{2}\left(\|\boldsymbol{x}^{\mathrm{s}}\|^2+\|\boldsymbol{x}^{\mathrm{n}}\|^2\right)+\tau\ell_t \tag{5-10}$$

对式(5-10)中参数 τ 进行求导，计算得出其解如式(5-11)：

$$\tau=min\left\{C,\frac{\ell_t}{\|\boldsymbol{x}^{\mathrm{s}}\|^2+\|\boldsymbol{x}^{\mathrm{n}}\|^2}\right\} \tag{5-11}$$

从而推出线性分类模型更新规则如式(5-12)所示：

$$\begin{aligned}\boldsymbol{w}^{\mathrm{s}}&=\boldsymbol{w}_t^{\mathrm{s}}+\min\left\{Cc_t y_t \boldsymbol{x}_t^{\mathrm{s}},\frac{\ell_t c_t y_t \boldsymbol{x}_t^{\mathrm{s}}}{\|\boldsymbol{x}^{\mathrm{s}}\|^2+\|\boldsymbol{x}^{\mathrm{n}}\|^2}\right\}\\ \boldsymbol{w}^{\mathrm{n}}&=\min\left\{Cc_t y_t \boldsymbol{x}_t^{\mathrm{n}},\frac{\ell_t c_t y_t \boldsymbol{x}_t^{\mathrm{n}}}{\|\boldsymbol{x}^{\mathrm{s}}\|^2+\|\boldsymbol{x}^{\mathrm{n}}\|^2}\right\}\\ \boldsymbol{w}^{\mathrm{m}}&=\boldsymbol{w}_t^{\mathrm{m}}\end{aligned} \tag{5-12}$$

模型训练过程中的类不平衡处理算法伪代码如算法 5-2 所示。

算法 5-2 类不平衡处理

输入 投影得到的实例 $\boldsymbol{x}_t^{\mathrm{s}}$、$\boldsymbol{x}_t^{\mathrm{n}}$，以及不同特征空间下的线性分类器 $\boldsymbol{w}^{\mathrm{s}}$、$\boldsymbol{w}^{\mathrm{n}}$ 和 $\boldsymbol{w}^{\mathrm{m}}$.

输出 更新后的分类器 $\boldsymbol{w}_t$.

1. 预测当前训练实例的标签 $\hat{y}_t=\mathrm{sign}\left(\boldsymbol{x}_t^{\mathrm{s}}\cdot\boldsymbol{w}_t^{\mathrm{s}}+\boldsymbol{x}_t^{\mathrm{n}}\cdot\boldsymbol{w}_t^{\mathrm{n}}\right)$；
2. 揭示真实标签 y_t 利用式(5-2)计算瞬时损失 ℓ_t；
3. `if` $y_t>0$ `then`
4. $\mathrm{Num}_{\mathrm{p}}$ ++;
5. `else`
6. $\mathrm{Num}_{\mathrm{n}}$ ++;
7. `end if`
8. 计算误分类代价因子 c_t；
9. 更新分类器 $\boldsymbol{w}^{\mathrm{s}}$、$\boldsymbol{w}^{\mathrm{n}}$ 和 $\boldsymbol{w}^{\mathrm{m}}$；
10. 维护全局特征空间下的线性分类器 $\boldsymbol{w}_t=[\boldsymbol{w}^{\mathrm{s}},\boldsymbol{w}^{\mathrm{n}},\boldsymbol{w}^{\mathrm{m}}]$；

5.2.3 模型稀疏策略

由于特征演化流的特征空间动态变化，分类模型应具备良好的可扩展性以适应递增的特征空间，同时全局特征空间下分类器 $\boldsymbol{w}_t$ 的特征维度 d_t^w 也将不断增大。为控制最大维数以提高内存使用和运行效率，引入投影截断技术以过滤冗余特征，维护一个稀疏且有效的分类模型。

传统的分类器截断方法通常按一定比例保留分类器权重 $\boldsymbol{w}_t$ 中的最大值而将其余值置 0 的方法，对分类器进行截断。然而在学习动态特征演化流时，部分特征可能只出现在少量实例中，它们对应的分类器权重固然较小，但不能简单认为这部分特征不重要而截断。因此，每轮迭代时统计分类器中每个特征 $f_i(1\leqslant i\leqslant d_t^w)$ 的标准差 s_i 和平均值 μ_i，根据式(5-13)计算其变异系数 CV_i：

$$\mathrm{CV}_i=\frac{\mathrm{s}_i}{\mu_i} \tag{5-13}$$

利用 CV_i 衡量分类器特征的重要性，第 t 轮每个特征的权重 k_i 如式(5-14)：

$$\mathrm{k}_t^i=\frac{\mathrm{CV}_t^i}{\sum_{i=1}^{d_t^w}\mathrm{CV}_t^i} \tag{5-14}$$

可对分类器 $\boldsymbol{w}_t$ 进行投影稀疏，如式(5-15)所示：

$$\bar{\boldsymbol{w}}_t=\min\left\{1,\frac{\beta}{\langle \boldsymbol{w}_t,K_t\rangle}\right\}\boldsymbol{w}_t \tag{5-15}$$

其中，$K_t=[k_t^1,k_t^2,\cdots,k_t^{d_t}]$ 由当前每个特征的权重信息组成，正则化参数 $\beta>0$。经过投影后的分类器需进行截断，保留给定比例 B 的最为重要特征，实现模型的稀疏处理，具体流程伪代码如算法 5-3 所示。

算法 5-3 稀疏截断

输入：线性分类器 $\boldsymbol{w}_t$.

输出：截断后的最终分类器 $\boldsymbol{w}_{t+1}$.

```
for i = 1,2,⋯, d_t^w do
  if ‖w_t‖ > B·d_t^w then
    利用式(5-15)的投影方法得到 w̄_t 中每个特征的权重;
      保留 w̄_t 中绝对值大的前 B·d_t^w 个元素，其余设为 0，得到 w̄_t^B ;
    w_{t+1} = w̄_t^B ;
  end if
end for
```

5.2.4 算法复杂性分析

OIFE 的算法复杂性分析主要包含时间复杂度和空间复杂度的分析。其中，记第 t 轮实例的特征维度为 d_t，对应本轮分类器的特征维度为 d_t^w。

在特征划分阶段，特征空间预处理、实例与分类器投影以及特征统计信息计算是同步进行的，因此该阶段时间复杂度为 $O(d_t)$；在实例预测阶段，组合预测标签并计算瞬时损失的时间复杂度为 $O(d_t)$；不平衡处理阶段，计算代价敏感因子时间复杂度为 $O(1)$；在投影截断阶段，对分类器处理的时间复杂度为 $O(d_t^w)$。

综上所述，每轮迭代过程中 OIFE 的时间复杂度为 $O(d_t + d_t^w)$，与实例及模型的特征维度成正比。另外，由于 OIFE 以在线方式更新分类器而不保留历史数据，故该算法空间复杂度为 $O(1)$。

5.3 实验设计与结果分析

5.3.1 数据集与实验设置

为验证本章所提方法的有效性，分别在 11 个 UCI 数据集进行实验（所有数据集均可在 http://archive.ics.uci.edu 进行下载），其中包含 6 个相对平衡（IR<5.0）数据集和 5 个不平衡（IR≥5.0）数据集，以及 hyperP、LUdata、Phishing、Chesswekahe 四个概念漂移数据集，涉及生产生活的不同领域，如垃圾邮件、医疗诊断和金融等，数据集详细信息如表 5-1 所示。

通过对这些数据集人工随机移除与增加部分特征来模拟特征演化流，对每个数据实例都随机移除一定百分比的特征，实验设置人工移除特征比例为 40%，阶段 $T_1=T_2=T_3$ 各占数据集总数的 1/3，三个阶段中全局特征空间的特征个数分别占实例特征总数的 30%、60%、100%。

将 OIFE 方法与 OLIDS[86]、OLSF[55]以及 OLVF[57]方法进行对比实验，验证 OIFE 在不同测试数据集上的有效性并对实验结果进行分析讨论。为保证实验对比公平性，权衡参数 C、正则化因子 λ 等参数的设置，均保留算法在相应文献中的设置，以保证各算法的最优性能。所有算法通过 10 次随机重复实验并对实验结果取平均值。

表 5-1 数据集描述

Table 5-1 Description of datasets

数据集	实例数量	特征数量	少数类数量	多数类数量	不平衡率 IR
splice	3190	60	1527	1648	1.1
Credit-a	653	15	296	357	1.2
spambase	4601	57	1813	2788	1.5
WDBC	569	30	212	357	1.7
ionosphere	351	34	126	225	1.8
WPBC	198	34	47	151	3.2
spectrometer	531	93	45	486	10.8
arrhythmia	452	278	25	427	17.1
electricity	322	7	15	307	20.5
car_eval_4	1728	21	65	1663	25.6
hyperP	992	10	50	942	18.8
LUdata	1901	31	924	977	1.1
Phishing	11055	46	4898	6157	1.3
Chessweka	503	8	198	305	1.5

数据流分类任务中，通常使用准确率（Accuracy，ACC）评价模型性能，它能够反映模型在两个类别上整体预测精度，但面对类不平衡问题时，只使用准确率进行模型评价并不理想。几何均值（Geometric mean，G-mean）是对召回率与特异度的几何平均值，同时考虑对不平衡数据流中正、负类样本分类正确率，是不平衡数据流分类模型的有效评价指标。马修斯相关系数（Matthews correlation coefficient，MCC）能够反映实际与预测结果之间的相关性，综合考虑混淆矩阵中的 TP、TN、FP、FN，具有较高的均衡性，其取值范围是[-1,1]，越接近 1 表明分类器分类结果越好，越接近-1 则说明分类器性能越差，MCC 值为 0 则说明分类器在进行随机分类。误分类代价 cost 同样是不平衡分类任务的重要评价指标，用以衡量不同类别所造成的总损失，cost 值越低表明分类效果越好，由于少数类误分类的代价应更高，实验设置少数类和多数类误分类权重分别为 $c_p=10$ 和 $c_n=1$。因此，综合使用 ACC、G-mean、MCC、误分类代价 cost 这四个评价指标从多角度评价的模型分类性能。

5.3.2 平衡的特征演化流上对比实验

为观察 OIFE 算法对特征演化流的分类性能，在平衡的特征演化流上，将 OIFE 与 OLIDS、OLSF、OLVF 方法进行对比实验，最优分类结果用粗体表

示，如表 5-2 所示。

表 5-2 平衡的特征演化流上各对比算法分类性能

Table 5-2 Classification performance of compared algorithms on balanced datasets

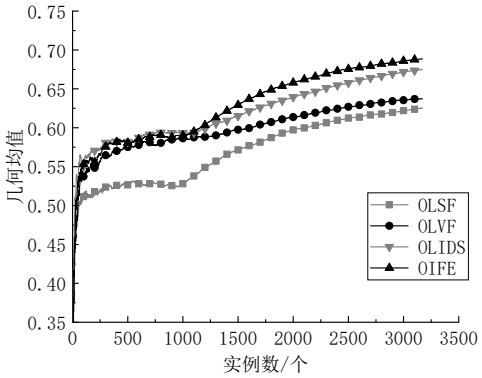
Dataset	评估指标	OLVF	OLSF	OLIDS	OIFE
splice	G-mean	0.637±0.007	0.626±0.010	0.675±0.007	**0.689±0.008**
	ACC	0.637±0.007	0.625±0.010	0.675±0.007	**0.688±0.008**
	MCC	0.275±0.013	0.251±0.019	0.350±0.013	**0.378±0.016**
	cost	6001.400±155.948	6121.100±207.278	5394.900±136.293	**4982.700±185.465**
Credit-a	G-mean	0.650±0.014	0.659±0.019	0.684±0.017	**0.701±0.013**
	ACC	0.652±0.014	0.659±0.018	0.681±0.017	**0.700±0.013**
	MCC	0.301±0.028	0.317±0.037	0.371±0.030	**0.402±0.027**
	cost	1299.100±115.129	1307.800±61.535	1370.800±125.383	**1229.300±76.608**
spambase	G-mean	0.747±0.007	0.763±0.006	0.788±0.005	**0.813±0.005**
	ACC	0.762±0.008	0.774±0.006	0.804±0.004	**0.821±0.004**
	MCC	0.500±0.015	0.527±0.012	0.587±0.009	**0.627±0.008**
	cost	5884.400±282.523	5822.000±198.417	4559.300±147.159	**4555.600±196.051**
WDBC	G-mean	0.907±0.008	0.889±0.005	0.905±0.011	**0.913±0.007**
	ACC	0.909±0.008	0.892±0.005	0.906±0.011	**0.913±0.008**
	MCC	0.807±0.016	0.771±0.011	0.802±0.023	**0.818±0.016**
	cost	**324.600±37.699**	385.600±34.581	346.800±52.423	327.500±42.005
ionosphere	G-mean	0.704±0.024	0.718±0.024	0.747±0.021	**0.760±0.021**
	ACC	0.710±0.020	0.740±0.023	0.756±0.025	**0.769±0.018**
	MCC	0.397±0.044	0.440±0.047	0.486±0.046	**0.512±0.039**
	cost	459.900±50.129	480.200±42.066	407.700±28.831	**384.400±40.329**
WPBC	G-mean	0.566±0.041	0.530±0.033	0.569±0.021	**0.584±0.036**
	ACC	0.549±0.031	0.527±0.028	0.549±0.017	**0.568±0.034**
	MCC	0.115±0.071	0.057±0.057	0.120±0.037	**0.146±0.063**
	cost	257.500±34.740	286.200±34.292	253.900±18.063	**246.700±27.989**

实验结果表明，在 6 个平衡的特征演化流上，OIFE 的 G-mean、ACC 和 MCC 值均为最高，在其中 5 个数据集上 cost 最低。相较于其他算法，OIFE 的 ACC 值提高了 0.4%~6.3%，G-mean 值提高了 0.6%~6.6%，MCC 值提高了 1.1%~12.7%，由于数据流的类别分布相对平衡，ACC 值更能准确反映模型分类效果。整体来看，OIFE 优于其他对比算法的原因在于特征划分原则更为合理，学习过程中能够尽可能充分地继承历史知识以训练模型。当然，OIFE 在提升预测精度的同时，但也损失了一部分稳定性。比如在 Splice 和 WDBC 数据集上，所提算法实验得到 ACC 值的方差较高，一方面是由于该方法在学习过程中保留的部分历史信息可能对分类器决策有干扰；另一方面是因为在数据集本身特征

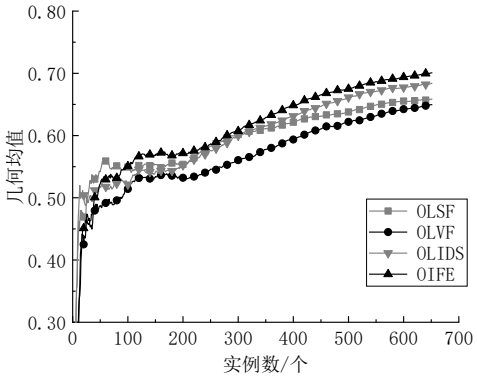
数量较少的前提下，特征还有 40%的特征随机消失，对模型泛化能力造成影响。

对模型训练过程中 G-mean、ACC 和 MCC 评价指标的性能变化曲线进行对比分析，分别如图 5-3、图 5-4 和图 5-5。通过比较在数据流学习初期性能曲线的斜率，算法 OIFE 的曲线斜率较高说明基本能够在短时间内达到相对较高的分类准确度，说明 OIFE 在学习平衡的特征演化流上具有较好的性能优势。对比发现在大多数据集上，OIFE 的性能表现优于其他方法，而在 ionosphere 数据集上前期预测结果波动较大，在中后期才逐渐达到稳定较好的分类效果，这主要是受数据集本身实例数量与特征维度的影响。以图 5-3（c）为例，在实例数量约在 150 以内时，所有算法在分类准确率结果上均表现出不稳定状态，其原因是初期实例数量较少且同时存在特征演化，随机消失或出现的特征很大程度上影响模型预测精度，同时说明数据流的在线学习效果不可避免地受限于训练实例的数量。

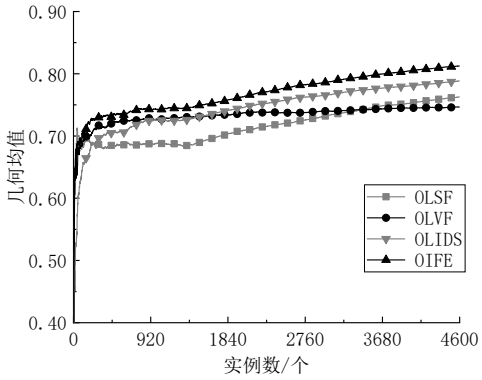
此外，随着新特征的不断加入，分类结果仍保持整体上升趋势，表明 OIFE 分类模型具有良好的泛化能力。以 spambase 数据集为例，通过横向比较图 5-3（c）、图 5-4（c）和图 5-5（c）可以发现所提算法在 G-mean、ACC 和 MCC 上的性能曲线变化趋势基本一致，由于在实例数的 1/3 和 2/3 处，即 t≈1500 和 t≈3000 时，分别加入部分新增特征，性能曲线均有小幅变化并继续上升，且 OIFE 基本始终优于 OLVF、OLSF 和 OLIDS 算法，并最终随着所学知识的积累使分类器性能达到稳定。



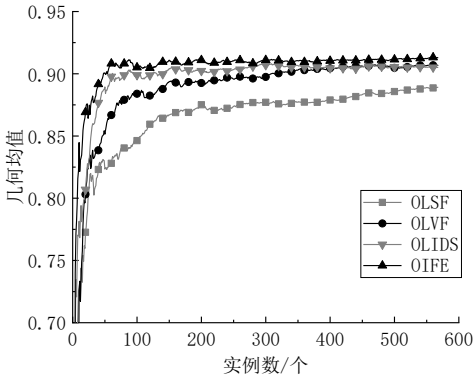
（a） splice 数据集



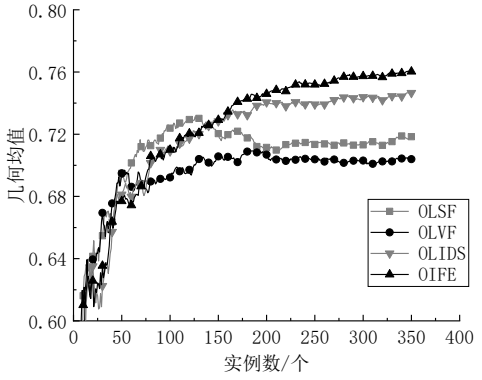
（b） Credit-a 数据集



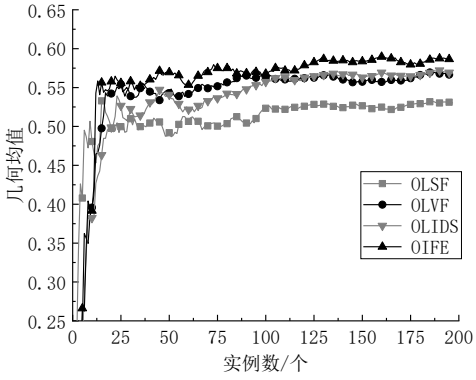
（c） spambase 数据集



（d） WDBC 数据集



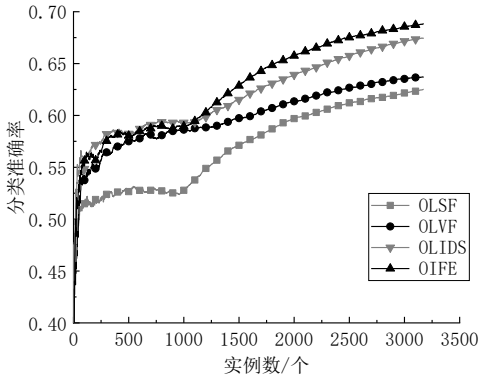
（e） ionosphere 数据集



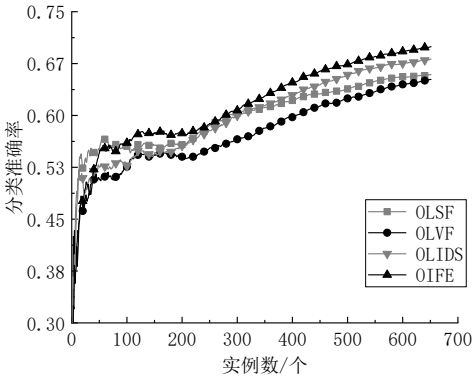
（f） WPBC 数据集

图 5-3 平衡的特征演化流上的 G-mean 结果

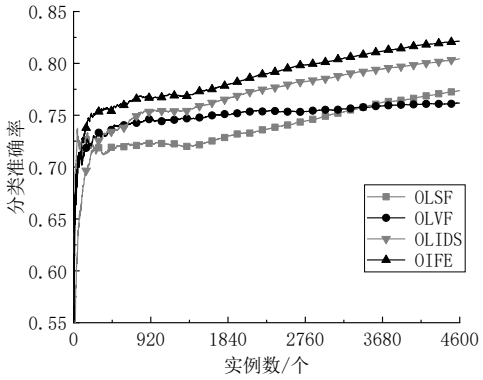
Fig. 5-3 G-mean results on balanced feature evolvable streams



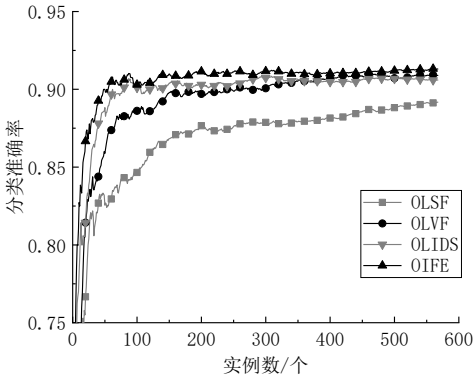
（a） splice 数据集



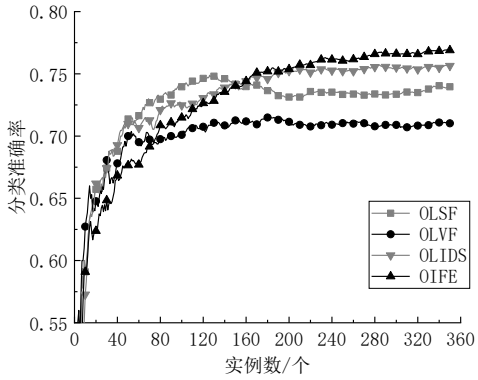
（b） Credit-a 数据集



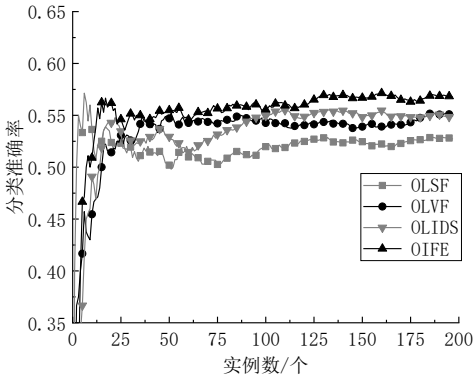
（c） spambase 数据集



（d） WDBC 数据集



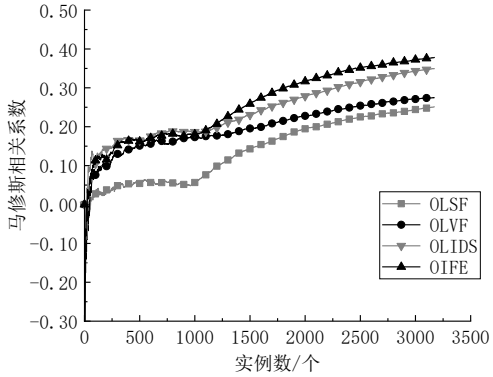
（e） ionosphere 数据集



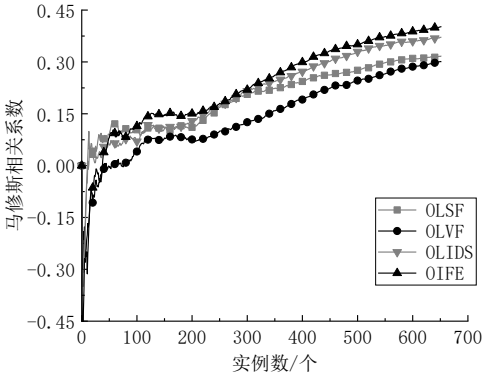
（f） WPBC 数据集

图 5-4 平衡的特征演化流上的 ACC 结果

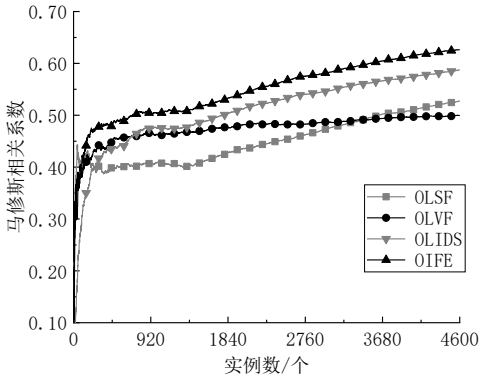
Fig. 5-4 ACC results on balanced feature evolvable streams



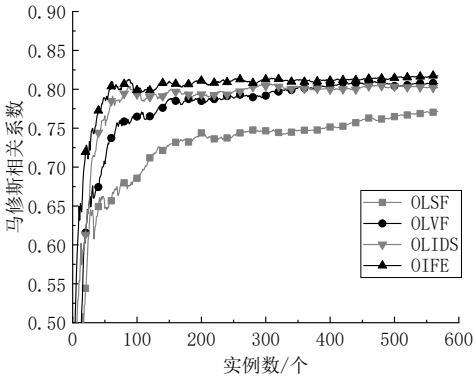
（a） splice 数据集



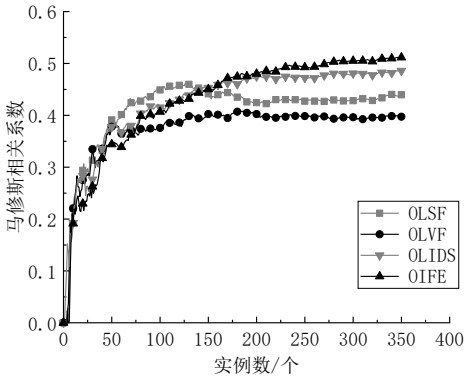
（b） Credit-a 数据集



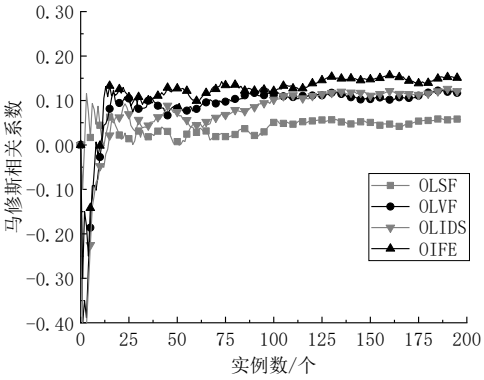
（c） spambase 数据集



（d） WDBC 数据集



（e） ionosphere 数据集



（f） WPBC 数据集

图 5-5 平衡的特征演化流上的 MCC 结果

Fig. 5-5 MCC results on balanced feature evolvable streams

5.3.3 类不平衡的特征演化流上对比实验

为探究使用 OIFE 同时处理类不平衡和特征演化问题的性能表现，在 5 个具有类不平衡的特征演化流上进行实验。同时，为验证 OIFE 算法中类不平衡处理模块的有效性，通过移除该模块以形成“OIFE-imb”方法进行对比。各对比算法基于四种评价指标的实验结果，最优结果均用粗体表示，如表 5-3 所示。

算法 OIFE 在 5 个数据集上的 G-mean 和 MCC 值取得最优结果，这表明所提算法能够在适应特征演化的同时有效处理类不平衡问题。此外，以 arrhythmia 数据集为例，算法 OLSF 的 cost 均值较低，ACC 值也比 OIFE 高出 4.4%，但相应 G-mean 和 MCC 值分别降低了 10.3%和 9.9%。这一现象说明，算法 OLSF 更关注多数类的分类准确率而对少数类实例分类效果欠佳，主要原因

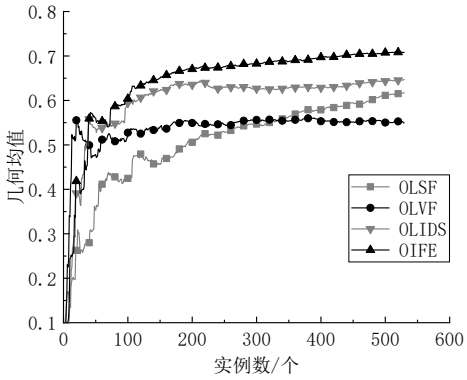
在于 OLSF 更关注提升分类器整体分类性能，而未侧重提高少数类实例的分类准确率。在 G-mean、MCC 和 cost 评价指标下，OIFE-imb 的分类性能相较于 OIFE 均显著下降，说明 OIFE 的不平衡处理机制可有效处理类不平衡问题。而对比同样都采取不平衡处理策略的 OLIDS 和 OIFE，发现在 5 个不平衡的数据集上，OIFE 算法性能整体优于 OLIDS，G-mean 值提升了 0.9%~15.8%，MCC 值提高了 0.8%~18.7%，说明改进的代价敏感策略可更好地提升对类不平衡数据流的学习能力。

图 5-6、图 5-7 和图 5-8 分别展示在 5 个不平衡的特征演化流上算法分类性能的变化曲线。观察图 5-6 可发现，随着数据实例的增加，算法 OIFE 在训练前期性能曲线波动较大，这主要由于实例数量与特征均较少且同时具有类不平衡问题，但随后能够快速达到并保持较好的分类结果，是因为该方法充分保留分类器中学习到的历史知识，从而降低特征演化给分类器带来的性能损失。

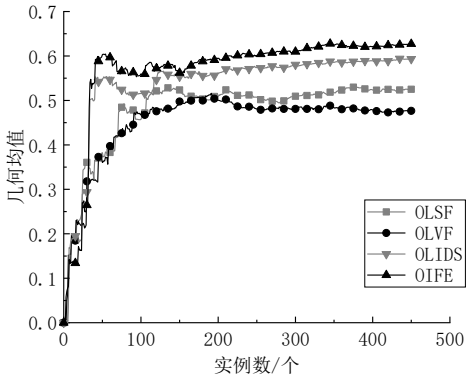
表 5-3 不平衡的特征演化流上的分类性能

Table 5-3 Classification performance on imbalanced feature evolvable streams

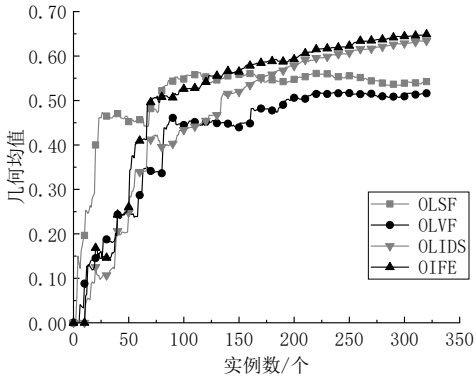
数据集	评估指标	OLVF	OLSF	OLIDS	OIFE-imb	OIFE
spectrometer	G-mean	0.550±0.014	0.618±0.041	0.647±0.016	0.668±0.037	**0.708±0.025**
	ACC	0.618±0.020	0.617±0.020	0.595±0.014	**0.630±0.023**	0.599±0.019
	MCC	0.064±0.014	0.135±0.046	0.171±0.034	0.195±0.044	**0.251±0.030**
	cost	**1818.400±112.768**	1878.100±85.432	2040.500±90.527	1853.500±139.947	2078.600±94.434
arrhythmia	G-mean	0.477±0.054	0.526±0.034	0.592±0.029	0.565±0.032	**0.629±0.041**
	ACC	0.545±0.018	**0.587±0.022**	0.489±0.037	0.516±0.021	0.543±0.030
	MCC	-0.015±0.044	0.029±0.029	0.100±0.028	0.063±0.031	**0.128±0.042**
	cost	1926.600±73.828	**1749.300±94.975**	2251.400±174.212	2102.300±93.360	2008.300±135.224
electricity	G-mean	0.516±0.120	0.544±0.054	0.634±0.031	0.611±0.050	**0.648±0.046**
	ACC	0.516±0.046	0.506±0.021	0.537±0.016	0.532±0.018	**0.552±0.014**
	MCC	0.020±0.102	0.040±0.048	0.123±0.031	0.103±0.049	**0.136±0.044**
	cost	1497.000±125.784	1535.100±58.545	1460.500±53.153	1468.200±58.750	**1412.300±42.332**
car_eval_4	G-mean	0.584±0.033	0.525±0.024	0.670±0.015	0.619±0.033	**0.679±0.018**
	ACC	0.522±0.016	0.527±0.013	0.536±0.020	0.513±0.016	**0.542±0.017**
	MCC	0.069±0.029	0.019±0.018	0.145±0.013	0.103±0.032	**0.153±0.015**
	cost	8055.900±267.419	7889.000±208.959	7940.200±357.403	8274.200±311.848	**7845.200±296.637**



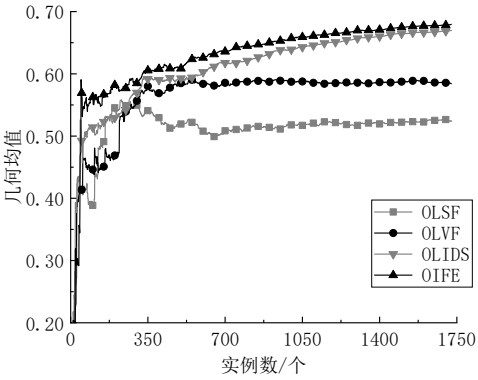
（a） spectrometer 数据集



（b） arrhythmia 数据集



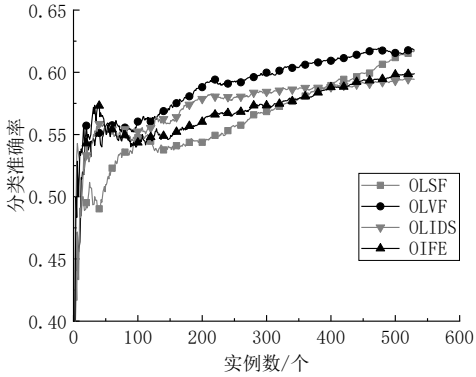
（c） electricity 数据集



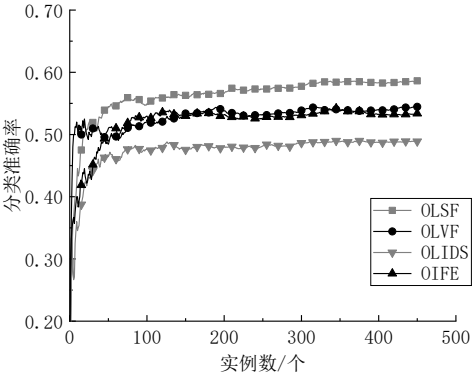
（d） car_eval_4 数据集

图 5-6 类不平衡的特征演化流上的 G-mean 结果

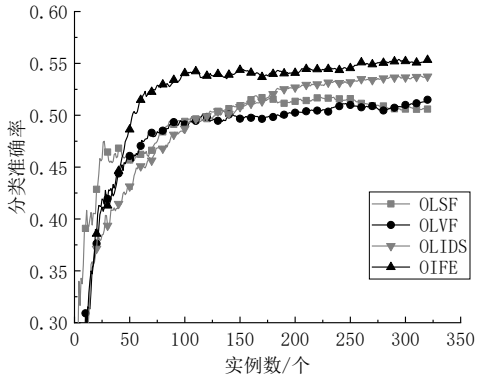
Fig. 5-6 G-mean results on imbalanced feature evolvable streams



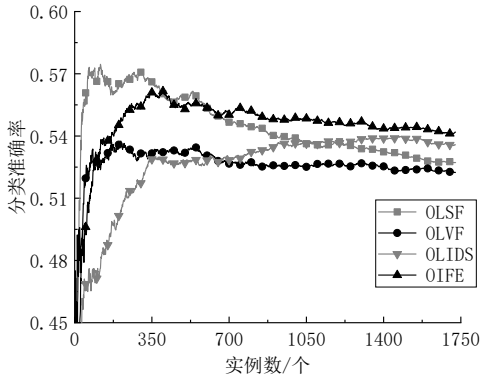
（a） spectrometer 数据集



（b） arrhythmia 数据集



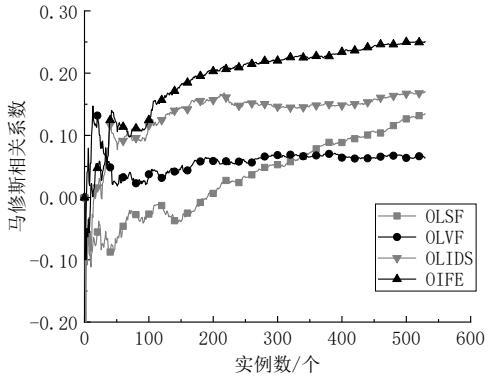
（c） electricity 数据集



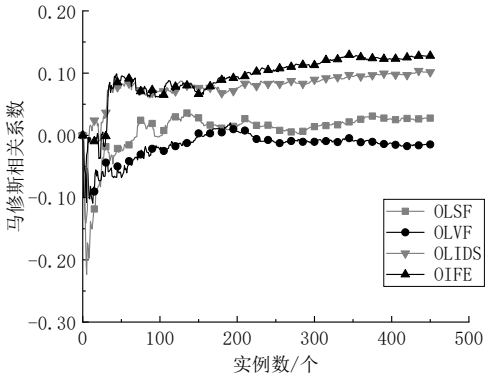
（d） car_eval_4 数据集

图 5-7 类不平衡的特征演化流上的 ACC 结果

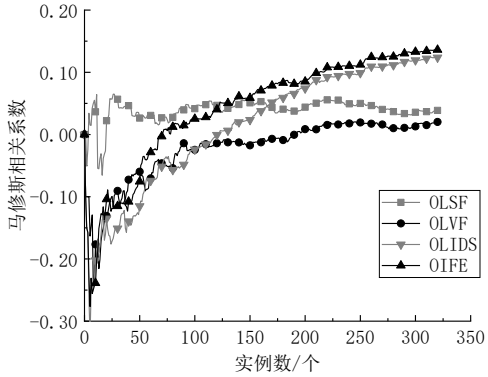
Fig. 5-7 ACC results on imbalanced feature evolvable streams



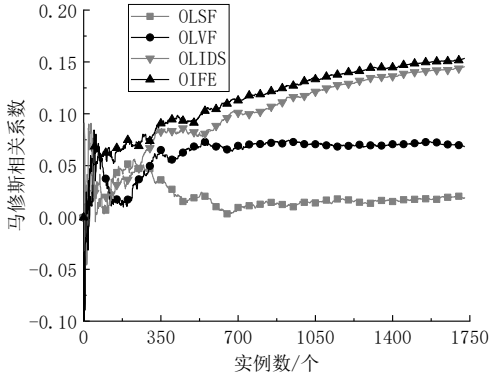
（a） spectrometer 数据集



（b） arrhythmia 数据集



（c） electricity 数据集



（d） car_eval_4 数据集

图 5-8 类不平衡的特征演化流上的 MCC 结果

Fig. 5-8 MCC results on imbalanced feature evolvable streams

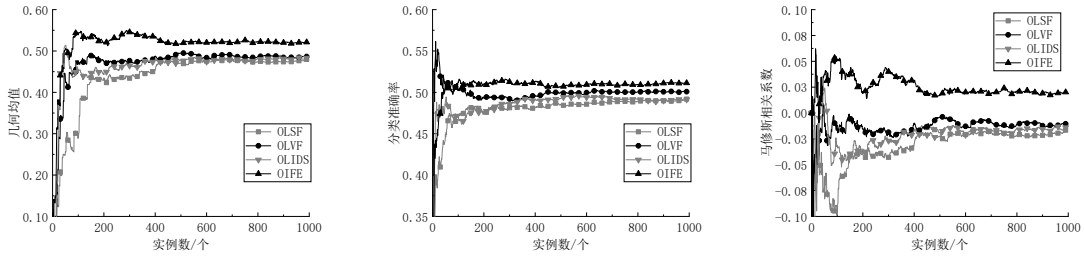
5.3.4 具有概念漂移的特征演化流上对比实验

为验证 OIFE 对概念漂移问题的适应能力，在 4 个具有概念漂移的特征演化流上进行对比实验，各对比算法基于四种评价指标的实验结果，最优结果均用粗体表示，如表 5-4 所示。

表 5-4 具有概念漂移的特征演化流上的分类性能

Table 5-4 Classification performance of feature evolvable streams with concept drift

Dataset	评估指标	OLVF	OLSF	OLIDS	OIFE
hyperP	G-mean	0.486±0.031	0.479±0.038	0.483±0.024	**0.521±0.035**
	ACC	0.501±0.009	0.492±0.013	0.491±0.013	**0.511±0.016**
	MCC	-0.010±0.025	-0.017±0.032	-0.014±0.021	**0.020±0.032**
	cost	4715.300±101.053	4798.600±119.430	4812.200±127.740	**4641.200±164.811**
phishing	G-mean	0.847±0.003	0.809±0.004	0.869±0.002	**0.875±0.002**
	ACC	0.849±0.003	0.809±0.004	0.871±0.002	**0.876±0.002**
	MCC	0.694±0.006	0.615±0.007	0.739±0.004	**0.749±0.005**
	cost	8838.900±201.235	10678.500±222.702	7787.400±150.633	**7206.800±229.476**
chessweka	G-mean	0.542±0.054	0.606±0.022	0.637±0.018	**0.646±0.018**
	ACC	0.540±0.048	0.602±0.022	0.629±0.018	**0.637±0.018**
	MCC	0.085±0.106	0.209±0.044	0.275±0.035	**0.290±0.035**
	cost	1014.600±175.059	861.500±62.253	733.000±61.193	**717.100±44.210**
LUdata	G-mean	0.831±0.009	0.807±0.009	0.820±0.011	**0.833±0.008**
	ACC	0.831±0.009	0.807±0.009	0.820±0.010	**0.833±0.008**
	MCC	0.663±0.018	0.614±0.017	0.642±0.020	**0.667±0.016**
	cost	1971.600±126.169	2099.900±101.994	2194.600±164.847	**1990.400±155.634**



（a）几何均值　　（b）分类准确率　　（c）马修斯相关系数

图 5-9 在 hyperP 数据集上分类性能表现

Fig. 5-9 Classification performance on hyperP dataset

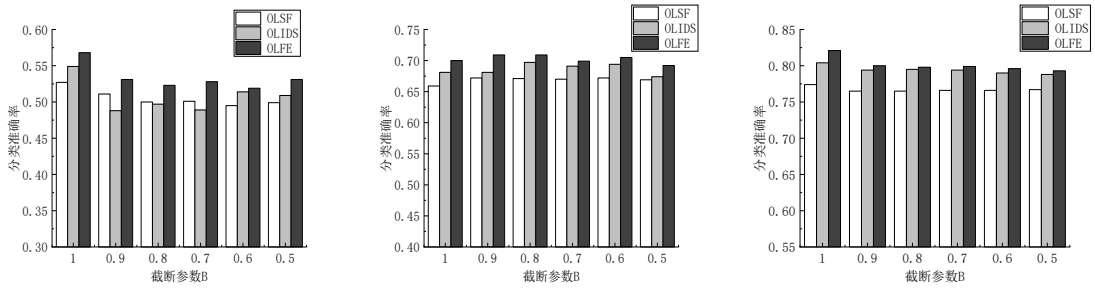
OIFE 在 4 个数据集上的 G-mean、ACC、MCC 以及 cost 值均取得最优结果，这表明所提算法能够有效对概念漂移数据流进行自适应学习，而 OLSF 的整体分类性能较差，与 OIFE 之间的 ACC 值最高相差 6.7%。同时，对比图 5-9（a）、（b）、（c）可发现，各对比算法在 hyperP 数据集上各项性能表现随着实

例数量的增加而先增后减，主要原因是该数据集的不平衡率达到 18.8，即该数据流上同时具有类不平衡、概念漂移以及特征演化问题，这不可避免地导致分类性能逐渐下降。但在这种情况下 OIFE 波动程度仍相对较小，且保持较高分类水平，说明该方法对复杂环境具有一定的泛化能力。

5.3.5 参数敏感性分析

为验证 OIFE 算法中投影截断模块的有效性，在[0.5,1]范围内选取不同的截断参数 B，间隔步长为 0.1，分别对同样采取投影截断技术的 OLSF、OLIDS 算法进行对比实验。

分析图 5-10 所示柱状图，OLSF 对分类器稀疏时仅仅保留权重较高的分类器特征，而 OLIDS 考虑到特征演化流中，部分特征因出现次数少而分配较低权重的不合理性，在模型截断时加入特征的方差信息，OIFE 则使用变异系数修正特征权重以剔除对分类任务不重要的特征。由此可知，OIFE 的分类性能整体较优且相对稳定，说明利用变异系数衡量特征的重要性，能够尽量消除稀疏截断特征时存在的偏见。



（a）WPBC 数据集　　（b）Credit-a 数据集　　（c）spambase 数据集

图 5-10 取不同 B 值时分类准确率结果

Fig. 5-10 ACC results with different B values

5.4 本章小结

针对数据流中同时存在特征演化和类不平衡的问题，提出具有类不平衡的特征演化数据流在线学习方法 OIFE。将实例特征空间划分为共享特征空间和新增特征空间，利用在线被动-主动算法分别训练不同特征空间下分类器以适应特征演化流；通过定义新的代价敏感因子，将最小化误分类代价和在线学习目标函数相结合，从而动态调整类别权重以处理类不平衡问题；最后，利用变异系数计算特征重要性对模型进行稀疏截断，以保障分类模型性能。

实验表明，相较于现有的特征演化数据流在线学习算法，OIFE 在学习特征演化流时表现出性能优势，同时在此基础上能够稳定地学习具有类不平衡的特征演化流。然而，数据流中往往还存在概念漂移现象，这将导致模型分类性能下降，如何进一步学习具有概念漂移的特征演化流值得深入研究。

第6章 总结与展望

本章对论文研究工作进行总结，并对特征演化数据流的分类方法的未来研究工作进行讨论。

6.1 工作总结

本文研究旨在解决数据流分类工作中的特征演化问题。在现实应用中，数据采集阶段环境复杂多变，进行数据流学习时往往面临噪声影响、类不平衡、特征演化等问题。因此，本文针对特征演化数据流特性，结合增量学习、集成学习以及特征映射等方法，构建一系列特征演化数据流分类方法，提高模型分类准确率、稳定性与泛化性。论文主要研究工作总结如下：

（1）提出一种基于映射策略的特征演化数据流分类方法。首先，该方法采用有效的模糊隶属函数来提高数据的可靠性，并采用孪生支持向量机算法对训练分类模型；随着新特征的出现，利用共现数据并结合局部权重线性回归算法，建立新特征空间与已有特征空间的映射关系；最后，利用这一映射关系在新的特征空间中重用历史分类模型以继续更新预测。在人工数据集上的对比实验结果表明，该方法对特征演化流的学习具有良好的分类性能，并且具有良好的抗噪能力。

（2）提出一种基于集成学习的特征演化数据流分类方法。首先，该方法通过引入模糊隶属度函数并结合增量孪生支持向量机，鲁棒地训练与更新分类器；当出现新特征时，重新训练新分类器，同时结合局部线性加权回归算法拟合新旧特征之间的映射关系，从而在旧特征消失时，利用所学到的映射关系，将已训练好的旧分类器投影至新特征空间继续更新；最后，结合两种不同的集成策略以合并新旧两分类器实现共同预测。通过大量仿真实验，所提方法分类准确率相较于对比方法有所提升；在含不同信噪比数据集上，分类模型性能整体优于对比模型，并随着人工增加噪声比例，模型分类效果受负面影响较小。因此验证该方法能够构建性能高效稳定的分类模型，在提升模型预测精度的同时能减少噪声对分类性能的干扰，增强模型对特征演化数据流自适应学习能力。

（3）提出一种具有类不平衡的特征演化数据流在线学习方法。该方法首先对实例特征进行划分，并将分类器分别投影至对应特征空间，结合在线被动-主动算法分别训练不同特征空间下的分类器；然后，将代价敏感指标最小化问题融入模型在线优化目标函数中，根据不平衡率定义新的代价敏感因子，动态调整类别权重以解决类不平衡问题；最后，为提高分类器泛化性能，利用变异系数筛选出重要特征，从而对分类器稀疏截断处理。大量仿真实验结果表明，该方法在 11 个 UCI 数据集上均获得较高的 ACC 值、G-mean 值和 MCC 值，结果表明所提方法能够高效学习特征演化数据流，同时能有效处理特征演化流中的类不平衡问题。

6.2 未来展望

在动态开放的环境下进行特征演化数据流挖掘取得较多研究成果，但仍然面临很多挑战，尚有较多问题值得进一步讨论与探索：

（1）数据标签匮乏问题。目前对于特征演化数据流的标签稀缺问题，已有部分工作利用在线半监督学习或在线主动学习给出解决方案，面对完全无标记的聚类问题也有利用微簇结构来增量更新的算法被提出。如何利用半监督学习、主动学习、迁移学习等方法来充分利用少量有标签数据提高分类性能，仍是未来重要研究工作之一。

（2）混合数据类型问题。特征演化问题的关键是构建不同特征之间的映射关系以辅助历史模型在新特征空间中进行预测，然而同时包含数值型和分类型的混合数据流普遍存在，经典度量方法不适用于不同数据特征类型之间的相似度计算。因此，需设计更加合理的度量方法，以适应具有混合数据类型的特征演化数据流挖掘任务。

（3）概念漂移问题。现实应用中收集的数据流往往随时间发生不可预测的变化，除特征演化外还存在概念漂移问题，这些问题为数据流挖掘工作带来巨大挑战，因此能够同时处理特征演化与概念漂移的数据流分类任务值得进一步研究。

参考文献

[1] Nguyen H L, Woon Y K, Ng W K. A Survey on Data Stream Clustering and Classification[J]. Knowledge and Information Systems, 2015, 45(3): 535-569.
[2] Gomes H M, Barddal J P, Enembreck F, et al. A survey on ensemble learning for data stream classification[J]. ACM Computing Surveys (CSUR), 2017, 50(2): 1-36.
[3] 文益民, 刘帅, 缪裕青, 等. 概念漂移数据流半监督分类综述[J]. 软件学报, 2022, 33(4): 1287-1314.
[4] Zhou Z H. Open Environment Machine Learning[J]. National Science Review, 2022, 9(8): nwac123.
[5] Dietterich T G. Steps Toward Robust Artificial Intelligence[J]. AI Magazine, 2017, 38(3): 3-24.
[6] 陈志强, 韩萌, 李慕航, 等. 数据流概念漂移处理方法研究综述[J]. 计算机科学, 2022, 49(9): 14-32.
[7] 聂秀山, 林熙明. 流数据概念漂移检测研究进展[J]. 山东建筑大学学报, 2022, 37(3): 90-100.
[8] 蔡桓, 陆克中, 伍启荣, 等. 面向概念漂移数据流的自适应分类算法[J]. 计算机研究与发展, 2022, 59(3): 633-646.
[9] 侯博建. 特征变化环境的机器学习方法研究[D]. 南京: 南京大学, 2020.
[10] Wang C, Xie L, Wang W, et al. Probing into the Physical Layer: Moving Tag Detection for Large-Scale RFID Systems[J]. IEEE Transactions on Mobile Computing, 2019, 19(5): 1200-1215.
[11] Baker S B, Xiang W, Atkinson I. Internet of Things for Smart Healthcare: Technologies, Challenges, and Opportunities[J]. IEEE Access, 2017, 5: 26521-26544.
[12] Manzoor E, Lamba H, Akoglu L. xstream: Outlier detection in feature-evolving data streams[C]// Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2018: 1963-1972.
[13] 李艳红, 谢梦娜, 王素格, 等. 基于特征扩展的微博短文本流热点话题检测方法[J]. 数据采集与处理, 2022, 37(3): 621-632.
[14] 赵旭剑, 杨春明, 李波, 等. 一种基于特征演变的新闻话题演化挖掘方法[J]. 计算机学报, 2014, 37(4): 819-832.
[15] Malave N, Nimkar A V. A survey on effects of class imbalance in data pre-processing stage of classification problem[J]. International Journal of Computational Systems Engineering, 2020, 6(2): 63-75.
[16] 翟婷婷, 高阳, 朱俊武. 面向流数据分类的在线学习综述[J]. 软件学报, 2020, 31(4): 912-931.
[17] Zinkevich M. Online convex programming and generalized infinitesimal gradient ascent[C]// Proceedings of the 20th international conference on machine learning (icml-03), 2003: 928-936.
[18] Crammer K, Dekel O, Keshet J, et al. Online Passive-Aggressive Algorithms[J]. Journal of Machine Learning Research, 2003, 7(3): 551-585.
[19] Dredze M, Crammer K, Pereira F C. Confidence-weighted linear classification[C]// Proceedings of the 25th international conference on Machine

learning, 2008: 264-271.
[20] Crammer K, Kulesza A, Dredze M. Adaptive regularization of weight vectors[J]. Machine learning, 2013, 91(2): 155-187.
[21] Krawczyk B, Minku L L, Gama J, et al. Ensemble learning for data stream analysis: A survey[J]. Information Fusion, 2017, 37(2): 132-156.
[22] Marcot B G, Penman T D. Advances in Bayesian network modelling: Integration of modelling technologies[J]. Environmental modelling & software, 2019, 111: 386-393.
[23] Ma J, Yang L, Sun Q. Capped L1-norm distance metric-based fast robust twin bounded support vector machine[J]. Neurocomputing, 2020, 412: 295-311.
[24] Altman N S. An introduction to kernel and nearest-neighbor nonparametric regression[J]. The American Statistician, 1992, 46(3): 175-185.
[25] Khemchandani R, Chandra S. Twin support vector machines for pattern classification[J]. IEEE Transactions on pattern analysis and machine intelligence, 2007, 29(5): 905-910.
[26] Lin C-F, Wang S-D. Fuzzy support vector machines[J]. IEEE transactions on neural networks, 2002, 13(2): 464-471.
[27] Krawczyk B, Cano A. Online ensemble learning with abstaining classifiers for drifting and noisy data streams[J]. Applied Soft Computing, 2018, 68: 677-692.
[28] Wang B, Pineau J. Online bagging and boosting for imbalanced data streams[J]. IEEE Transactions on Knowledge and Data Engineering, 2016, 28(12): 3353-3366.
[29] Wang H, Fan W, Yu P S, et al. Mining concept-drifting data streams using ensemble classifiers[C]// Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, 2003: 226-235.
[30] Zhang P, Li J, Wang P, et al. Enabling fast prediction for ensemble models on data streams[C]// Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, 2011: 177-185.
[31] Grossi V, Turini F. An adaptive selective ensemble for data streams classification[C]// International Conference on Agents and Artificial Intelligence, 2011: 136-145.
[32] Brzezinski D, Stefanowski J. Reacting to different types of concept drift: The accuracy updated ensemble algorithm[J]. IEEE transactions on neural networks and learning systems, 2013, 25(1): 81-94.
[33] 李昂, 韩萌, 穆栋梁, 等. 多类不平衡数据分类方法综述[J]. 计算机应用研究, 2022, 39(12): 3534-3545.
[34] Fu G H, Wu Y J, Zong M J, et al. Hellinger distance-based stable sparse feature selection for high-dimensional class-imbalanced data[J]. BMC bioinformatics, 2020, 21(1): 1-14.
[35] Shahee S A, Ananthakumar U. An effective distance based feature selection approach for imbalanced data[J]. Applied Intelligence, 2020, 50(3): 717-745.
[36] 王乐, 韩萌, 李小娟, 等. 不平衡数据集分类方法综述[J]. 计算机工程与应用, 2021, 57(22): 42-52.
[37] Chawla N V, Bowyer K W, Hall L O, et al. SMOTE: synthetic minority over-sampling technique[J]. Journal of artificial intelligence research, 2002, 16: 321-357.
[38] Soltanzadeh P, Hashemzadeh M. RCSMOTE: Range-Controlled synthetic minority over-sampling technique for handling the class imbalance problem[J]. Information Sciences, 2021, 542: 92-111.

[39] 蒋华, 江日辰, 王鑫, 等. ADASYN 和 SMOTE 相结合的不平衡数据分类算法[J]. 计算机仿真, 2020, 37(3): 254-258+420.
[40] Chawla N V, Lazarevic A, Hall L O, et al. SMOTEBoost: Improving prediction of the minority class in boosting[C]// Knowledge Discovery in Databases: PKDD 2003: 7th European Conference on Principles and Practice of Knowledge Discovery in Databases, Cavtat-Dubrovnik, Croatia, September 22-26, 2003. Proceedings 7, 2003: 107-119.
[41] Chen S, He H, Garcia E A. RAMOBoost: Ranked minority oversampling in boosting[J]. IEEE Transactions on Neural Networks, 2010, 21(10): 1624-1642.
[42] Johnson J M, Khoshgoftaar T M. Survey on deep learning with class imbalance[J]. Journal of Big Data, 2019, 6(1): 1-54.
[43] 李莉, 任振康, 石可欣. 代价敏感的 Boosting 软件缺陷预测方法[J]. 计算机工程, 2022, 48(03): 175-180.
[44] 万建武, 杨明. 代价敏感学习方法综述[J]. 软件学报, 2020, 31(01): 113-136.
[45] Wang J, Zhao P, Hoi S C. Cost-sensitive online classification[J]. IEEE Transactions on Knowledge and Data Engineering, 2013, 26(10): 2425-2438.
[46] Zhao P, Zhang Y, Wu M, et al. Adaptive cost-sensitive online classification[J]. IEEE Transactions on Knowledge and Data Engineering, 2018, 31(2): 214-228.
[47] Hou B-J, Zhang L, Zhou Z-H. Learning With Feature Evolvable Streams[J]. IEEE Transactions on Knowledge and Data Engineering, 2021, 33(6): 2602-2615.
[48] 刘艳芳, 李文斌, 高阳. 基于被动-主动的特征演化流学习[J]. 计算机研究与发展, 2021, 58(8): 1575-1585.
[49] 刘艳芳, 李文斌, 高阳. 特征演化的置信-加权学习方法[J]. 软件学报, 2022, 33(4): 1315-1325.
[50] Hou C, Zhou Z H. One-Pass Learning with Incremental and Decremental Features[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2018, 40(11): 2776-2792.
[51] 刘兆清, 古仕林, 侯臣平. 面向特征继承性增减的在线分类算法[J]. 计算机研究与发展, 2022, 59(8): 1668-1682.
[52] Hou B J, Zhang L, Zhou Z-H. Prediction With Unpredictable Feature Evolution[J]. IEEE Transactions on Neural Networks and Learning Systems, 2022, 33(10): 5706-5715.
[53] Zhang Z, Zhao P, Jiang Y, et al. Learning with Feature and Distribution Evolvable Streams[C]// Proceedings of the International Conference on Machine Learning. PMLR, 2020: 11317-11327.
[54] Lian H, Atwood J S, Hou B-J, et al. Online Deep Learning from Doubly-Streaming Data[C] //Proceedings of the 30th ACM International Conference on Multimedia. 2022: 3185-3194.
[55] Zhang Q, Zhang P, Long G, et al. Online learning from trapezoidal data streams[J]. IEEE Transactions on Knowledge and Data Engineering, 2016, 28(10): 2709-2723.
[56] He Y, Wu B, Wu D, et al. Toward mining capricious data streams: A generative approach[J]. IEEE transactions on neural networks and learning systems, 2020, 32(3): 1228-1240.
[57] Beyazit E, Alagurajah J, Wu X. Online learning from data streams with varying feature spaces[C]// Proceedings of the AAAI Conference on Artificial Intelligence, 2019: 3232-3239.
[58] Schlimmer J C, Granger R H. Incremental Learning from Noisy Data[J].

Machine Learning, 1986, 1(3): 317-354.
[59] 张本才. 概念漂移数据流增量学习算法及其应用研究[D]. 北京: 北京交通大学, 2020.
[60] He Y, Wu B, Wu D, et al. Online Learning from Capricious Data Streams: A Generative Approach[C]// Proceedings of the 28th International Joint Conference on Artificial Intelligence. 2019: 2491-2497.
[61] Guan S U, Li S. Incremental Learning with Respect to New Incoming Input Attributes[J]. Neural Processing Letters, 2001, 14(3): 241-260.
[62] Beyazit E, Hosseini M, Maida A, et al. Learning Simplified Decision Boundaries from Trapezoidal Data Streams[C]// Proceedings of the Artificial Neural Networks and Machine Learning–ICANN 2018: 27th International Conference on Artificial Neural Networks, Rhodes, Greece, October 4-7, 2018, Proceedings, Part I 27. Springer International Publishing, 2018: 508-517.
[63] Schreckenberger C, Glockner T, Stuckenschmidt H, et al. Restructuring of Hoeffding Trees for Trapezoidal Data Streams[C]// Proceedings of the 2020 International Conference on Data Mining Workshops (ICDMW). IEEE, 2020: 416-423.
[64] Gu S, Qian Y, Hou C. Incremental Feature Spaces Learning with Label Scarcity[J]. ACM Transactions on Knowledge Discovery from Data (TKDD), 2022, 16(6): 1-26.
[65] Liu Y, Fan X, Li W, et al. Online Passive-Aggressive Active Learning for Trapezoidal Data Streams[J]. IEEE Transactions on Neural Networks and Learning Systems, 2023, 34(10): 6725-6739.
[66] Ye H-J, Zhan D-C, Jiang Y, et al. Rectify Heterogeneous Models with Semantic Mapping[C]// Proceedings of International Conference on Machine Learning. PMLR, 2018: 5630-5639.
[67] Dong J, Cong Y, Sun G, et al. Evolving Metric Learning for Incremental and Decremental Features[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2022, 32(4): 2290-2302.
[68] Peng J, Guo J, Yang Q, et al. A general framework for mining concept-drifting data streams with evolvable features[C]// Proceedings of the 2021 IEEE International Conference on Data Mining (ICDM), 2021: 1276-1281.
[69] Tropp J A. Improved Analysis of the subsampled Randomized Hadamard Transform[J]. Advances in Adaptive Data Analysis, 2011, 3(1-2): 115-126.
[70] Hou B-J, Yan Y-H, Zhao P, et al. Storage Fit Learning with Feature Evolvable Streams[C]// Proceedings of the AAAI Conference on Artificial Intelligence, 2021: 7729-7736.
[71] Vitter J S. Random sampling with a reservoir[J]. ACM Transactions on Mathematical Software (TOMS), 1985, 11(1): 37-57.
[72] Zhang Z-Y, Zhao P, Jiang Y, et al. Learning with Feature and Distribution Evolvable Streams[C]// Proceedings of the International Conference on Machine Learning. PMLR, 2020: 11317-11327.
[73] Wu D, He Y, Luo X, et al. Online Feature Selection with Capricious Streaming Features: A General Framework[C]// Proceedings of the 2019 IEEE International Conference on Big Data (Big Data), 2019: 683-688.
[74] Luo X, Zhou M, Xia Y, et al. An Efficient Non-Negative Matrix-Factorization-Based Approach to Collaborative Filtering for Recommender Systems[J]. IEEE Transactions on Industrial Informatics, 2014, 10(2): 1273-1284.
[75] Koren Y, Bell R M, Volinsky C. Matrix Factorization Techniques for

Recommender Systems[J]. Computer, 2009, 42(8): 30-37.
[76] You D, Sun M, Liang S, et al. Online feature selection for multi-source streaming features[J]. Information Sciences, 2022, 590: 267-295.
[77] He Y, Yuan X, Chen S, et al. Online Learning in Variable Feature Spaces under Incomplete Supervision[C]// Proceedings of the AAAI Conference on Artificial Intelligence, 2021: 4106-4114.
[78] 佟强, 刁恩虎, 李丹, 等. 分类任务中标签噪声的研究综述[J]. 科学技术与工程, 2022, 22(31): 13626-13635.
[79] Cleveland W S, Devlin S J. Locally weighted regression: an approach to regression analysis by local fitting[J]. Journal of the American statistical association, 1988, 83(403): 596-610.
[80] Dong X, Yu Z, Cao W, et al. A survey on ensemble learning[J]. Frontiers of Computer Science, 2020, 14(2): 241-258.
[81] 许冠英, 韩萌, 王少峰, 等. 数据流集成分类算法综述[J]. 计算机应用研究, 2020, 37(1): 1-8+15.
[82] 申明尧, 韩萌, 杜诗语, 等. 数据流决策树集成分类算法综述[J]. 计算机应用与软件, 2022, 39(9): 1-10.
[83] Mello A R D, Stemmer M R, Lameiras Koerich A. Incremental and Decremental Fuzzy Bounded Twin Support Vector Machine[J]. Information Sciences, 2020, 526: 20-38.
[84] Gao B-B, Wang J-J, Wang Y, et al. Coordinate descent fuzzy twin support vector machine for classification[C]// Proceedings of the 2015 IEEE 14th international conference on machine learning and applications (ICMLA), 2015: 7-12.
[85] Gomes H M, Read J, Bifet A, et al. Machine learning for streaming data: state of the art, challenges, and opportunities[J]. ACM SIGKDD Explorations Newsletter, 2019, 21(2): 6-22.
[86] You D, Xiao J, Wang Y, et al. Online Learning From Incomplete and Imbalanced Data Streams[J]. IEEE Transactions on Knowledge and Data Engineering, 2023, 35(10): 10650-10665.

攻读学位期间参与的科研项目

[1] 安徽省高校自然科学研究重点项目（KJ2021A0516）

攻读学位期间发表的学术论文目录

[1] 陈燕菲, 刘三民. 具有类不平衡的特征演化流在线学习方法[J/OL].计算机工程,1-13. http://kns.cnki.net/kcms/detail/31.1289.tp.20240119.1348.008.html.

[2] 陈燕菲, 刘三民. 面向特征演化数据流的增量学习方法研究[J/OL].重庆工商大学学报(自然科学版),1-12. http://kns.cnki.net/kcms/detail/50.1155.N.20230731.1508.002.html.

[3] Y. Chen, S. Liu. A Novel Learning Method for Feature Evolvable Stream[J]. Evolving Systems. https://doi.org/10.1007/s12530-024-09590-9. (WOS:001220299400001)

致 谢

三年很短，似弹指间，仿若昨日才为顺利录取而欣喜若狂，转眼今日便因面临毕业而思绪万千；三年很长，漫漫亦灿灿，时而对辗转反侧的深夜无奈苦笑，时而为备忘录里的小灵感欢欣鼓舞。读研三年，沉淀三年，自是进一寸有一寸的欢喜。

一敬师长，饮水思源谢师恩。感谢我的导师刘三民教授，先生教我严谨求真的学术态度，教我豁达乐观的处事方式，教我迎难而上，教我沉心静气，耐心认真地与我沟通交流，不厌其烦地为我答疑解惑，得恩师如此，乃吾辈之至幸。同时衷心感谢计算机与信息学院的全体老师，老师们在课堂上传授的专业知识与生活中的关心体贴，给予我极大的帮助与支持。学生在此祝老师们工作顺利！

二叩父母，春晖寸草难为报。感谢我的父母，父母为我营造和谐民主的家庭氛围，培养我广泛阅读与独立思考的良好习惯，教会我积极勇敢地面对生活中的挑战。十八岁背上行囊离家读书，我知父母不求我大富大贵，但求我顺遂无虞；如今成材可为他们遮风避雨，望父母知我愿你们无忧无虑，从此多喜乐常欢愉。女儿在此祝爸爸妈妈身体健康！

三谢好友，山水一程皆缘分。感谢我的朋友们，感谢我的师兄师弟们、同届“战友”们以及可爱的姐妹们，同窗三载，朝夕相处，感谢相遇。大家在学习和生活上对我的关心和鼓励，让我这个异乡客感受到大家庭的温暖。在此祝朋友们前程似锦！

最后，谢谢亲爱的自己。感谢普通平凡的你，感谢独一无二的你，感谢不言放弃的你，人生是旷野，大胆走弯路，便尽是我能走的路。“回首向来萧瑟处，归去，也无风雨也无晴”，在此祝自己乘风破浪、光芒万丈！

“三”，三生万物，又泛指多数，是个很妙的字眼。我是万千生灵中平凡的个体，选择用三年时光打磨自己，现今不知炼成功力几何，但我很快乐，很知足，也很幸福。

过去已过去，未来还未来，再见又再见。