计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.210200047

基于信息熵更新权重的数据流集成分类算法

夏　源[1]　赵蕴龙[1,2]　范其林[1]
1 南京航空航天大学计算机科学与技术学院　南京 211106
2 软件新技术与产业化协同创新中心　南京 210023
(xiayuan@nuaa.edu.cn)

摘　要　在动态的数据流中,由于其不稳定性以及存在概念漂移等问题,集成分类模型需要有及时适应新环境的能力。目前通常使用监督信息对基分类器的权重进行更新,以此来赋予符合当前环境的基分类器更高的权重,然而监督信息在真实数据流环境下无法立即获得。为了解决这个问题,文中提出了一种基于信息熵更新基分类器权重的数据流集成分类算法。首先使用随机特征子空间对每个基分类器进行初始化来构建集成分类器;其次基于每个新到来的数据块构建一个新的基分类器来替换集成中权重最低的基分类器;然后基于信息熵的权重更新策略实时对基分类器中的权重进行更新;最后满足要求的基分类器参与加权投票,得到分类结果。将所提算法和几个经典学习算法进行对比,实验结果表明,所提方法的分类准确性有着明显优势,并且适合多种类型的概念漂移环境。

关键词:数据流;概念漂移;信息熵;分类;集成算法
中图法分类号　TP391

Data Stream Ensemble Classification Algorithm Based on Information Entropy Updating Weight

XIA Yuan[1],ZHAO Yun-long[1,2] and FAN Qi-lin[1]
1 School of Computer Science and Technology,Nanjing University of Aeronautics and Astronautics,Nanjing 211106,China
2 Collaborative Innovation Center of Novel Software Technology and Industrialization,Nanjing 210023,China

Abstract　In the dynamic data stream,due to its instability and the existence of concept drift,the ensemble classification model needs the ability to adapt to the new environment in time. At present,the weight of the base classifier is usually updated by using the supervision information,so as to give higher weight to the base classifier suitable for the current environment. However,supervision information cannot be obtained immediately in a real data stream environment. In order to solve this problem,this paper presents a data stream ensemble classification algorithm,which updates the weight of the base classifier through information entropy. Firstly,the random feature subspace is used to initialize each base classifier to construct the ensemble classifier. Secondly,a new base classifier is constructed based on each new data block to replace the base classifier with the lowest weight in the ensemble. Then,the weight update strategy based on information entropy will update the weights in the base classifier in real time. Finally,the base classifier that meets the requirements participates in weighted voting to obtain the classification result. Comparing the proposed algorithm with several other classic learning algorithms,the experimental results show that the proposed method has obvious advantages in classification accuracy and is suitable for various types of concept drift environments.

Keywords　Data stream,Concept drift,Information entropy,Classification,Ensemble algorithm

1　引言

由于信息技术的快速发展,互联网、通信、工业等各行各业都在持续产生海量的数据,而且在以一个惊人的速度不断增长,数据以流的形式不断产生。

在传统的机器学习环境中,数据通常以静态数据的形式存储在数据库中。也就是说,所有的数据可以立刻拿到并可以全部直接拿来使用。而在数据流挖掘中,数据流以顺序、连续、快速的方式生成数据,且在本质上有可能是短暂存在的,不能长期存储在内存当中[1]。除此之外,由于数据流的不稳定性,概念漂移的存在也是其一大显著的特点。概念漂移是一种现象,其表现在目标域的统计属性随着时间的发展而发生任意方式的变化,也就是说数据流中的数据分布发生了不可预知的变化[2]。由于概念漂移的存在,不平稳数据流中的分类模型需要有及时适应新环境的能力。

针对上述问题,采用集成分类器对不平稳环境下的数据流进行分类任务是一种有效的策略。集成分类器本身由多个基础分类器组成,由于集成分类器天然具有模块化结构,使得其能够很自然地适应不平稳数据流中数据分布随着时间而产生的变化。数据流的集成分类算法主要分为两大类:基于

到稿日期:2021-02-04　返修日期:2021-07-08
通信作者:赵蕴龙(zhaoyunlong@nuaa.edu.cn)

数据块的集成分类算法和基于在线学习的集成分类算法。

基于数据块的集成分类算法在数据流传输过来的时候将数据打包为某些指定大小的数据块。这类算法的性能很大程度上取决于数据块的大小。最初的一种基于数据块的集成分类算法是 Street 等提出的流集成算法 SEA[3]。SEA 的显著缺点在于替换分类器的方式，为了解决这个问题，精度加权集合 AWE[4]算法被提出。Brzezinski 等在 AWE 上进行改进，提出了 AUE 算法[5]，在该算法中，所有的基分类器都是用新的数据块中的一部分实例进行增量更新。这有助于减少因用少量数据创建而导致性能较差的基础分类器。该算法的另一个优点在于用非线性误差函数对分类器进行加权，可以更好地提高集成分类器的准确性。为了在集成分类器中主动解决概念漂移问题，Elwell 等提出了 Learn++.NSE[6]算法。

基于在线学习是另一种数据流集成学习方法[7]。与基于块的方法不同，在线方法按顺序处理元素，而不是分块处理。DWM 是由 Kolter 等首先提出的在线集成方法[8]。对于一些测试例子，基础分类器各自提供一个预测，结合权重输出整体的预测。如果个别分类器预测错误，则相应的权重会降低。类似地，基于在线学习的分类器还有 FHDDMS[9]和 AES[10]等。然而使用基于在线学习的集成方式对于渐变类型的概念漂移的处理效果不太理想。

针对基于数据块和基于在线学习的集成分类算法的优缺点，Ren 等提出了一种结合数据块和在线学习的混合类型的集成分类器[11]。该算法同时利用监督信息和非监督信息来对漂移进行检测以及对基分类器进行权重更新。Cano 等提出了 KUE[12]方法。该算法使用 Kappa 统计信息进行动态加权和基础分类器的选择。

然而上述提到的算法在对基分类器进行更新时，通常需要使用到数据的标签信息来更新基分类器的权重，然而在实际数据流应用环境中，往往无法实时获取数据的标签信息。针对该问题，本文提出了一种基于信息熵的权重更新算法 IEWU(Information Entropy-based Weight Updating)，并据此给出数据流集成分类解决方案。IEWU 整合了基于数据块和在线学习的思想对概念漂移进行快速自适应。该算法通过使用信息熵这个非监督信息来进行各个基分类器评估和权重更新，使当前环境下更有效的基分类器能够被赋予更高的权重；并设定一个权重阈值，只有高于该阈值的基分类器才能参与最后的加权投票。不同于大多数集成分类器，IEWU 使用非标签信息进行模型的权重更新，使其在类标签稀有的环境下也能够使用。

2　相关工作

2.1　数据流描述

给定一个时间段$[0,t]$，可以把在$[0,t]$中的数据流样本表示为$S_{0,t}=\{d_0,\cdots,d_t\}$，其中$d_i=(\boldsymbol{X}_i,y_i)$表示在第 i 时间点上数据实例。每个实例都是独立的，$\boldsymbol{X}_i$表示特征向量，y_i表示类标记。

与传统静态数据相比，数据流具有以下几个特点[13]：

(1)数据实例不是事先给出的，而是以流的形式顺序到达或者以数据块的形式逐渐变得可用。

(2)数据实例可能会迅速到达。

(3)数据流的大小可能是无限的。

(4)每个实例数据只能访问有限的次数。

(5)数据流中的数据必须在有限的时间内被处理，避免数据堆积。

(6)每个数据流中的数据的标签查询或获取成本较高，很难访问真正的类标签。

2.2　概念漂移描述

假设$S_{0,t}$遵循一定的数据分布$F_{0,t}(\boldsymbol{X},y)$。当$F_{0,t}(\boldsymbol{X},y)\neq F_{t+1}(\boldsymbol{X},y)$时，则概念漂移发生在时间点 $t+1$ 处，也表示为$P_t(\boldsymbol{X},y)\neq P_{t+1}(\boldsymbol{X},y)$[14]。由于联合概率$P_t(\boldsymbol{X},y)$可以分解为两部分：$P_t(\boldsymbol{X},y)=P_t(\boldsymbol{X})\times P_t(y|\boldsymbol{X})$。因此概念漂移可以概括为以下 3 种情况：

(1)$P_t(\boldsymbol{X})\neq P_{t+1}(\boldsymbol{X})$而$P_t(y|\boldsymbol{X})=P_{t+1}(y|\boldsymbol{X})$。由于$P_t(\boldsymbol{X})$漂移不影响决策边界，因此该情况也称为虚假漂移。

(2)$P_t(y|\boldsymbol{X})\neq P_{t+1}(y|\boldsymbol{X})$而$P_t(\boldsymbol{X})=P_{t+1}(\boldsymbol{X})$。这种情况会导致决策边界发生变化，并导致模型的准确性下降，因此称之为真实漂移。

(3)第 3 种情况结合了前两种情况，即$P_t(\boldsymbol{X})\neq P_{t+1}(\boldsymbol{X})$且$P_t(y|\boldsymbol{X})\neq P_{t+1}(y|\boldsymbol{X})$。

3　基于信息熵更新权重的集成分类算法

本节具体介绍了基于信息熵更新权重的数据流集成分类算法 IEWU。

3.1　集成结构和初始化

在 IEWU 算法中，第一步要做的就是构造一个集成分类器。假设 E 是一个由 k 个基础分类器 γ 组成的集成分类器，其中$\gamma_i\in E$。令 S 为数据流，该数据流被均匀划分成大小相等的$B_1,B_2,\cdots,B_n$。当数据流 S 中的数据块B_i到达时，基础分类器开始被初始化，整个集成分类器也开始初始化。由于 IEWU 算法是基于数据块和在线学习融合的混合模型，因此当一个新的数据块到达时，若是数据块中的所有数据的标签都已经能够获取，则对该数据块进行训练，得到由最新数据块训练成的基础分类器。当集成分类器 E 中的基础分类器个数未满足 k 个时，该算法将不断构建基分类器加入到集成分类器 E 中，直到集成分类器 E 中的基分类器个数满足 k 个。

集成分类器由多个基础分类器组成，想要形成一个性能良好的集成分类器，基础分类器不仅应该有一定的准确性，而且也应该具有一定的多样性[15]。为了促进基础分类器之间的多样性，本节探索了不同维度的特征子空间。每个基分类器γ_i都建立在不同的 r 维随机子空间上，其中$1\leqslant r\leqslant f$，f 是原始维度。因此，每个组件的维度和特征子空间都是随机的。另一方面，IEWU 算法还采用了 Poisson(1)分布对子空间内的实例进行权重更新。

3.2　集成模型更新

为了解决数据流的不稳定问题，IEWU 算法需要实时对集成分类器进行更新。该算法对集成模型的更新分为两个部分，一是对基分类器进行更新，二是对基分类器权重进行更新。

3.2.1　基分类器更新

由于 IEWU 算法融合了基于数据块集成分类器的思想，

因此当带有标签的数据量达到了能组成一个数据块的时候，该算法将获取这个新数据块，然后利用其进行训练得到一个候选的基分类器。新的基分类器的构建使用上述相同的构建方式，采用 r 维的随机子空间来构建，并使用 Poisson(1)分布对每个实例进行权重赋值。基分类器的权重也会随着对数据块进行预测分类的性能的变化而相应地变化，以此来应对不稳定的数据流。数据块大小的设置对整个集成分类器来说尤为重要且不容易确定。较大的数据块往往会生成分类性能更好的基分类器，但是可能其数据块内部就包含概念漂移。较小的数据块通常在带有概念漂移的数据流上更有效，但是其对于分类性能可能不太理想。之后用构建出来的候选基分类器来替换集成分类器中权重最低的基分类器，以此来遗忘由于数据概念漂移而不再有效的旧分类器，改善整个集成分类器的性能。

3.2.2 基于信息熵的权重更新

由于 IEWU 是一个混合类型的集成分类器，因此在 IEWU 中，不仅需要通过新的基分类器替换旧的基分类器，还需要对已有的基分类器进行在线学习，并基于信息熵来对每个基分类器的权重进行更新。通过这种方式，可以使当前时刻性能更好的分类器参与投票，以此来提高集成模型在不平稳数据流中的自适应能力。

信息熵作为信息论中一个重要的参数，被用作度量信息量。熵的概念原先来自于热力学领域，用来定义分子之间的混乱程度。分子之间越混乱，代表熵的值越大。Shannon 将熵从热力学引入到信息论中，提出了信息熵这个概念[16]。信息熵用来度量一个信息量的混乱程度，即用来衡量一个事件的不确定性。一个事件的信息量越大，或者说一个事件的不确定性越大，则信息熵的值越大。信息熵的计算公式如下：

$$H=E[-\log p_i]=-\sum_{i=1}^{n} p_i \log p_i \tag{1}$$

其中，p_i 表示分类器分类为第 i 类的概率。

IEWU 算法通过信息熵计算公式求得当前数据实例的信息熵值。由于信息熵代表着分类结果的不确定性，因此信息熵越大，则分类结果越不确定，相反地，信息熵越少，则代表分类结果越确定。当 IEWU 算法对当前实例求得的信息熵的值足够小时，该算法判定当前的分类结果是确定且准确的。由于不同数据流的样本的输出空间不一样，因此在不同数据流下的熵的区间值都不相同。于是用来判定当前分类结果是否确定的熵的阈值采用了动态自适应的方式来求得，其针对不同的数据流都有效。

$$e_{\text{thre}}=e_{\text{mean}}-\frac{2*(e_{\text{mean}}-e_{\min})}{3} \tag{2}$$

其中，e_{thre} 表示熵的阈值，e_{mean} 表示到目前为止熵的平均值，$e_{\min}$ 表示熵的最小值。e_{mean} 和 $e_{\min}$ 的值会随着数据流的不断到来而不断更新。当 IEWU 对当前数据求得的信息熵的值小于 e_{thre} 时，该算法判定当前分类结果是正确的，然后每个基分类器判断当前分类结果是否与整个集成分类结果相同，若不相同，则取值为 1。

$$S_x=\begin{cases}1, & \text{if } (e<e_{\text{thre}}) \text{ and } (f_i(x)\neq f(x))\\ 0, & \text{other}\end{cases} \tag{3}$$

其中，e 表示集成分类器对当前实例求得的信息熵，$f_i(x)$ 是第 i 个基分类器的分类结果，$f(x)$ 是集成分类器的分类结果。第 i 个基分类器的权重计算如下：

$$w_i=1-\frac{\sum S_x}{|X|},\ \forall x\in X \tag{4}$$

其中，w_i 表示第 i 个基分类器的权重值，$|X|$ 表示当前数据块中数据实例的个数。

一般来说，基分类器刚开始创建时会被赋予最高的权重值，随着数据块的到来，每个基分类器会通过上述方式求得对当前数据块的预测结果，然后实时调整其相应的权重，使得当前环境下性能更好的基分类器能够有更高的权重。

3.3 集成决策

集成分类器的最终预测结果由所有的基分类器的预测结果进行加权投票。加权投票的公式如下：

$$H(x)=\arg\max_{j}\sum_{i=1}^{k}\begin{cases}w_i h_i^j(x), & \text{if } w_i>\theta\\ 0, & \text{other}\end{cases} \tag{5}$$

其中，$H(x)$ 表示整个集成分类器的预测结果，$h_i^j(x)$ 表示第 i 个基分类器分类为第 j 类的输出结果，$\boldsymbol{w}_i$ 表示第 i 个基分类器的权重，k 表示集成分类器中基分类器的个数。

IEWU 算法还用到了抛弃策略，并非所有的基分类器最后都会参与投票。由于数据流的不稳定性，当有概念漂移产生时，某几个基分类器的分类性能会因此大大降低。这时若将准确性差的分类器加入到最终的加权投票中，反而会大大降低整个集成分类器的准确性。因此该算法将抛弃能力不够好的一部分基分类器。给定一个固定的权重阈值，该算法只会将权重值大于阈值的基分类器加入最终的加权投票中，而权重值小于阈值的基分类器则会被抛弃。

3.4 算法整体流程

IEWU 的伪代码如算法 1 所示。

算法 1

输入：数据流 S，集成中基分类器的最大个数 k，特征维度值 f，权重 w，判断基分类器是否参与加权投票的阈值 θ，特征子空间 φ

输出：分类结果

```
1. for S_i∈{S_1,…,S_n} do
2.     if S_1 then
3.         for j∈{1,…,k} do
4.             r←[1,f]中的随机整数
5.             φ_j←S_1中的 r 维随机子空间，其中实例的权重根据 Poisson(1)加权
6.             γ_j←在φ_j上训练基分类器
7.         end for
8.     else
9.         if 数据标签可用 do
10.            φ_j←S_1中的 r 维随机子空间，其中实例的权重根据 Poisson(1)加权
11.            γ_j←在φ_j上增量更新基分类器
12.            w_j←计算信息熵，并用信息熵更新权重
13.        end if
14.        if 收集到一个新的带标签的数据块S_1 then
15.            r←[1,f]中的随机整数
16.            φ←S_1中的 r 维随机子空间，其中实例的权重根据 Poisson(1)加权
17.            γ←在φ_j上训练基分类器
```

18.　　end if
19.　　用 γ 替换集成中权重最低的基分类器
20.　end if
21. end for

4　实验结果及分析

本文的实验环境如下：Windows 10 专业版 64 位操作系统；Intel i5-4460 CPU @ 3.20GHz 处理器；8GB 内存；并使用 Java 编程语言。

本算法以及对比实验算法均在大规模在线分析系统(Massive Online Analysis，MOA)平台上进行实验。MOA 是一个开源的数据流挖掘平台，它为数据流挖掘提供算法和运行实验的环境[17]。

4.1　实验数据集

本实验采用的数据集包括 3 个人工合成数据集以及 2 个真实场景下的数据集。其中人工数据集中有 2 个人工合成数据集包含突变类型的概念漂移，1 个人工合成数据集包含渐变类型的概念漂移。详细信息如表 1 所列。

表 1　实验中使用的数据集

Table 1　Data sets used in experiment

数据集	实例数	属性数	类别数	漂移数	漂移类型
$RanTree_s$	1 M	5	2	3	突变
$Agrw_s$	1 M	9	2	3	突变
SEA_g	1 M	3	2	3	渐变
CoverType	581 K	53	7	—	未知
Poker	829 K	10	10	—	未知

(1)人工数据集

RanTree：随机树生成器基于一个随机数来生成数据流[18]。使用该生成器生成一个突变类型数据集。每个数据集包含 100 000 个实例，每个实例有 5 个属性，分为 2 类。$RanTree_s$ 表示带有突变类型的概念漂移，每隔 25 000 个实例后会发生突变漂移，总共有 3 个漂移。

Agrw：Agrw 数据流生成器最初来源于 Agrawal 等[19]的论文。使用该生成器生成一个突变类型数据集。数据集包含 100 000 个实例，每个实例有 9 个属性，分为 2 类。$Agrw_s$ 表示带有突变类型的概念漂移，每隔 25 000 个实例后会发生突变漂移，总共有 3 个漂移。

SEA：SEA 数据集生成器可以模拟概念漂移的产生。实验模拟了带有渐变类型的概念漂移的数据集 SEA_g，SEA_g 包含有 1 000 000 个实例，并且每隔 25 000 个实例后会发生渐变漂移，总共有 3 个漂移生成。由于 SEA 生成器还可以模拟噪声数据，因此添加了 10% 的噪声数据到数据流中，以此来评估分类器的鲁棒性。

(2)真实数据集

CoverType：即森林覆盖类型(forest covertype)。此数据集包含从美国森林服务(USFS)区域 2 资源信息系统(RIS)数据获得的 30 m * 30 m 的森林覆盖面积。它包含 581 012 个数据实例。每个数据实例有 54 个属性，分为 7 种类别。该数据集是归一化后的数据集。

Poker：即 Pokwe hand。该数据集包含 1 000 000 个数据实例，每个数据实例有 10 个属性。数据集中的每一条数据都是一个手牌的记录，该手牌由从 52 张标准牌组中抽出的 5 张纸牌组成，每张纸牌使用两个属性(花色和数字)进行描述，总共有 10 个预测类别。该数据集是归一化后的数据集。

4.2　对比算法介绍与参数分析

(1)对比算法

朴素贝叶斯(Naïve Bayesian，NB)[20]：朴素贝叶斯不带任何处理概念漂移的方法，可以作为基准参考算法。

精度更新集成分类器(Accuracy Updated Ensemble，AUE)[5]：AUE 是一种基于数据块的集成分类器。

精度加权集合(Accuracy Weighted Ensemble，AWE)[4]：AWE 根据从顺序的数据块中读取的数据实例来构造各个基础分类器，然后将其添加到固定大小的集合中。AWE 是一个基于数据块的集成分类器。

动态加权多数(Dynamic Weighted Majority，DWM)[8]：DWM 是经典的基于在线学习的集成分类算法。通过使用权重和增量学习，使 DWM 算法能够处理概念漂移问题。

在线 Bagging(Online Bagging，OBag)[21]：OBag 是 Bagging 算法的在线版本。

在线 Boosting(Online Boosting，OBoost)[21]：OBoost 是 Boosting 算法的在线版本。

基于信息熵更新权重的集成分类算法(IEWU)：IEWU 算法是本文提出的算法，是一种基于数据块和在线学习的混合型算法。

不带信息熵更新权重的集成分类算法(IEWU-NoUpdate)：IEWU-NoUpdate 是 IEWU 不使用信息熵对基分类器权重进行更新的版本，该算法的基分类器的权重保持不变。该对比算法是为了证明用信息熵进行权重更新的有效性。

(2)实验参数设置

本节实验以及对比实验完全在 MOA 框架中实现运行。在本节实验中，IEWU 都与监督算法进行比较，并通过标记大量样本数据进行实验。IEWU 采用 Hoeffding 树作为基分类器，总共设定基分类器的个数为 10，设定数据块的大小为 1 000，设定判断基分类器是否参与加权投票的阈值 。实验最终主要通过评估分类器的准确性来判断各个分类的性能。

4.3　实验结果分析

(1)数据块大小对实验影响分析

表 2 列出了本文算法 IEWU 在不同数据块大小下的分类准确度。实验结果可以表明，当使用不同的数据块大小时，该算法的性能不存在显著差异，但是总体而言，当取值为 1 000 时，精度相对要高一点。因此，本文的数据块大小设定为 1 000。

表 2　不同数据块下的分类准确度

Table 2　Classification accuracy under different data blocks

(单位：%)

数据集	500	750	1 000	1 250	1 500
$RanTree_s$	93.00	**93.89**	92.49	91.99	91.57
$Agrw_s$	90.56	91.24	**91.73**	91.48	90.87
SEA_g	87.85	88.10	**88.50**	87.94	88.38
CoverType	**86.77**	86.24	85.30	85.13	84.63
Poker	84.90	84.59	**85.13**	83.49	84.19

(2)分类准确性分析

图 1 给出了各种算法在 $RanTree_s$ 数据集上的分类性能。可以明显看出,本文提出的算法 IEWU 从头到尾都保持着最高的准确率并一直呈现上升趋势。与 IEWU-NoUpdate 相比,IEWU 准确性一直领先,且 IEWU-NoUpdate 在经过第一个漂移点时准确性有些下降,说明基于信息熵的基分类器权重更新能够在不平稳的环境中有着快速适应的效果,因此其准确性有提升。AUE,AWE,DWM,OBag 和 OBoost 在整个过程中一直保持着平稳或上升的趋势且 NB 算法在经过第一个漂移点时准确性快速下降,说明这些算法对数据集上的概念漂移能够快速适应。NB 作为基准参考模型,不能很快地适应概念漂移。实验结果证明,IEWU 算法在 $RanTree_s$ 数据集上有快速适应的能力和最好的性能。

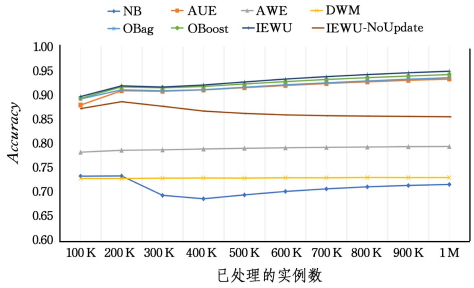

图 1 各种算法在 $RanTree_s$ 上的分类准确性

Fig. 1 Classification accuracy of the algorithm on $RanTree_s$

图 2 给出了各种算法在 $Agrw_s$ 数据集上的分类性能。可以看出,到达第 1 个漂移点时,NB,DWM 和 IEWU-NoUpdate 的准确性开始下降,而其他 5 个分类器的准确性没有受到很大的影响。在第 2 个漂移点时,所有分类器的准确性都开始快速降低,相比于其他 7 种分类器,IEWU 的准确性下降幅度最低,在准确性上也最高。在经过第 3 个漂移点之后,除了 NB 之外,其他每个分类器的性能都慢慢趋于平稳或者上升。实验结果表明,在 $Agrw_s$ 数据集上,IEWU 算法有着最好的性能。IEWU 与 IEWU-NoUpdate 相比,IEWU 也展示出了基于信息熵的基分类器权重更新的有效性。其能够快速适应概念漂移,在不平稳的数据流下一直有着良好的性能。

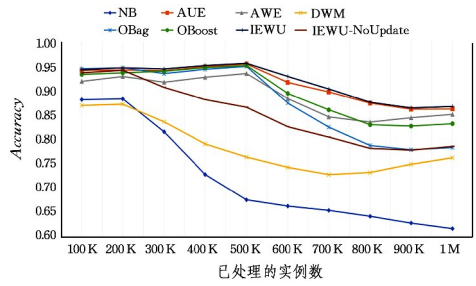

图 2 各种算法在 $Agrw_s$ 上的分类准确性

Fig. 2 Classification accuracy of the algorithm on $Agrw_s$

图 3 给出了各种算法在 SEA_g 数据集上的分类性能。SEA_g 带有 3 个渐近类型的概念漂移,并有 10%的噪声。从实验结果中可以看出,在带有渐近类型的漂移的数据集中,在经过第 1 个漂移点时,NB 的准确性首先开始下降,其他分类器的准确性都还保持稳定。在经过第 2 个漂移点时,OBag 和 IEWU-NoUpdate 的准确性开始下降。而 AUE, AWE, DWM, OBoost 和 IEWU 在整个过程中都能够保持准确性的稳定且有着较好的分类准确性。在整个实验中,IEWU 有着最高的分类准确性,NB 由于其不带任何的概念漂移处理机制,因此分类准确性最低。IEWU 与 IEWU-NoUpdate 相比,其不仅准确性更高,在处理概念漂移的能力上也远优于 IEWU-NoUpdate。因此可以证明,在 SEA_g 数据集上基于信息熵的基分类器权重更新不仅能够提高集成分类器的准确性,还能够快速适应不稳定的数据流。

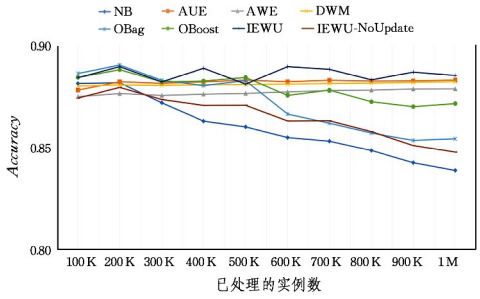

图 3 各种算法在 SEA_g 上的分类准确性

Fig. 3 Classification accuracy of the algorithm on SEA_g

从上述分类器在带有概念漂移的人工数据集上的表现来看,综合所有的数据集,本文所提的集成分类算法 IEWU 有着更好的表现,在每个数据集上的准确性都能够排在前列。并且通过 IEWU 和 IEWU-NoUpdate 的对比发现,基于信息熵的基分类器权重更新策略能够有效提升分类器的性能,并在带有概念漂移的数据流下有着较好的适应效果。

图 4 给出了各种算法在真实数据集 CoverType 下的分类性能。由于真实数据集无法知道漂移点的个数和位置,因此无法通过漂移点的位置对算法的性能进行分析,只能对算法的准确性进行常规分析。在 CoverType 数据集上,OBoost 相比其他 7 种算法,其有着最好的分类准确性。而其他 7 个分类器中一开始准确性最高的是 OBag,但随着数据的到来,IEWU 准确性慢慢高于 OBag,这说明 IEWU 在该数据流下的适应能力比 OBag 要强。NB 的分类准确性最差,因为 NB 不带有任何处理概念漂移的策略。IEWU 的准确性一直高于 IEWU-NoUpdate,这表明在真实数据集 CoverType 上,基于信息熵的基分类器权重更新对分类器准确性的提升有着很好的效果。

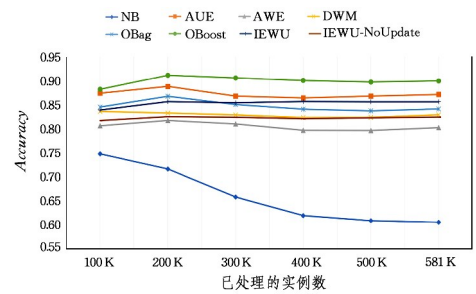

图 4 各种算法在 CoverType 上的分类准确性

Fig. 4 Classification accuracy of the algorithm on CoverType

图 5 给出了各种算法在真实数据集 Poker 上的分类性能。在 Poker 数据集上,OBoost 和 IEWU 的准确性几乎没有差别,都比其他分类器的准确性要高,并且其准确性随着时间

的推移不断提高，说明 OBoost 和 IEWU 在 Poker 下的适应性能力很好。AUE，AWE 和 NB 与其他 5 个分类器相比，准确性都比较差。IEWU 的准确性一直要高于 IEWU-NoUpdate，这表明在真实数据集 Poker 上，基于信息熵的基分类器权重更新对分类器准确性的提升有着很好的效果。

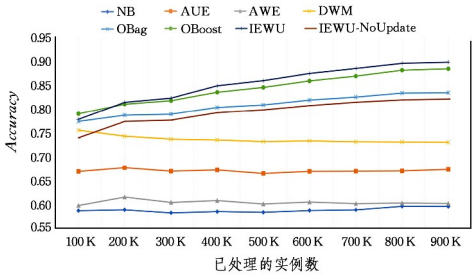

图 5　各种算法在 Poker 上的分类准确性

Fig. 5　Classification accuracy of algorithms on Poker

从分类器在两个真实数据集上的表现来看，本文提出的算法 IEWU 有着较为不错的性能，与其他分类器相比有更好的准确性。并且通过对比 IEWU 和 IEWU-NoUpdate 发现，基于信息熵的基分类器权重更新策略也能够在真实的数据集下有效地提升分类的性能。

表 3 列出了对比算法在所有数据集上的实验结果。从表 3 可以明显看出，由于 NB 不带有适应概念漂移的策略，因此 NB 在所有数据集上的准确性都处于最低。而本文提出的 IEWU 算法在 RanTree_s，Agrw_s，SEA_g 和 Poker 这 4 个数据集上有着最好的性能和最高的准确性。在 CoverType 上 OBoost 有着最高的准确性，其次是 IEWU 算法。而通过对比 IEWU 与 IEWU-NoUpdate 算法可知，不管在哪个数据集上 IEWU 算法始终能够保持更好的性能。上述结果说明，相比不带信息熵更新权重的算法，本文提出的基于信息熵的基分类器权重更新策略更有效。信息熵作为一个非监督信息，能够在各个数据集上保持着最高或者次高的准确性，并且在快速适应带有概念漂移的数据流下也有着相当不错的性能表现。

表 3　对比算法的平均分类准确度

Table 3　Average classification accuracy of comparison algorithm

（单位：%）

数据集	NB	AUE	AWE	DWM	OBag	OBoost	IEWU	IEWU-NoUpdate
RanTree_s	70.99	91.62	79.08	73.01	91.89	92.49	92.95	86.57
Agrw_s	71.77	91.31	88.79	78.35	87.59	89.43	91.73	85.00
SEA_g	85.96	88.20	87.70	88.11	87.16	87.89	88.50	86.51
CoverType	65.91	87.21	80.48	82.89	84.70	89.94	85.30	82.23
Poker	58.81	66.98	60.40	73.54	80.68	84.19	85.13	79.23

(3)运行时间对比

为了进一步比较各种算法在各个数据集环境下的性能效果，表 4 列出了在不同数据集上各种算法的运行时间。可以看出，在大多数数据集中，AWE 算法有着最高的运行时间。由于 NB 是单分类器，而其他对比算法都是集成分类器，因此在所有的数据集中，NB 有着最少的运行时间并且远小于对比的集成分类器。而在集成分类器算法中，IEWU 算法在绝大部分数据集都有最低的运行时间。IEWU-NoUpdate 算法的运行时间和 IEUE 相差不多。上述结果表明了在集成分类器中，IEUE 算法在运行时间上有着较大的优势，总体上优于其他集成算法。

表 4　对比算法的运行时间

Table 4　Comparison algorithm running time

（单位：s）

数据集	NB	AUE	AWE	DWM	OBag	OBoost	IEWU	IEWU-NoUpdate
RanTree_s	2.92	110.59	220.72	88.05	67.53	81.55	60.71	61.35
Agrw_s	3.61	88.66	152.16	89.25	65.80	84.64	62.92	61.42
SEA_g	2.86	31.59	53.80	67.06	20.84	25.22	19.68	22.24
CoverType	10.45	150.59	295.25	119.83	69.86	93.95	65.57	66.83
Poker	4.92	53.53	63.83	37.59	35.23	50.03	40.36	39.23

结束语　本文首先介绍了一种基于信息熵更新权重的数据流集成分类算法 IEWU，该算法能够在不平稳的数据流下有良好的分类性能，并且能够对数据流中的概念漂移进行快速的适应。在本文中，为了促进基分类器之间的多样性，每个基分类器都建立在不同的 r 维随机子空间上，并采用了 Poisson(1)分布对子空间内的实例进行权重更新。基于数据块的更新策略则会在满足要求时创建一个新的候选分类器，并用该候选分类器来替换集成分类器中权重最低的基分类器。基于信息熵的权重更新策略会实时对基分类器中的权重进行更新，使得在整个数据流过程中当前时刻下性能好的基分类器往往能被赋予更高的权重。信息熵作为一个非标签信息，还能够在标签稀少的环境下拥有较好的性能。然后，本文介绍了整个集成算法的最终决策过程，只有权重值满足要求的基分类器才能参与最终的加权投票。实验结果表明，本文提出的基于信息熵来更新权重的策略在准确率和适应概念漂移的性能上都有着较大的优势。

IEWU 算法采用了信息熵参数来调整基分类器的权重，信息熵代表着分类结果的不确定性，如何设定一个合理且准确的阈值来判断当前分类结果对实验结果有着较大的影响，是后续工作需要着重研究的方向。

参 考 文 献

[1] KRAWCZYK B，MINKU L L，GAMA J，et al. Ensemble learning for data stream analysis：A survey[J]. Information Fusion，2017，37：132-156.

[2] KHAMASSI I，SAYED-MOUCHAWEH M，HAMMAMI M，

et al. Discussion and review on evolving data streams and concept drift adapting[J]. Evolving Systems, 2018, 9(1): 1-23.
[3] STREET W N, KIM Y S. A streaming ensemble algorithm (SEA) for large-scale classification[C]// Proc. of the Acm Sigkdd Int. Conference on Knowledge Discovery & Data Mining. 2001: 377-382.
[4] 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.
[5] 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.
[6] ELWELL R, POLIKAR R. Incremental learning of concept drift in nonstationary environments[J]. IEEE Transactions on Neural Networks, 2011, 22(10): 1517-1531.
[7] LV Y, PENG S, YUAN Y, et al. A classifier using online bagging ensemble method for bigdata stream learning[J]. Tsinghua Science and Technology, 2019, 24(4): 379-388.
[8] KOLTER J Z, MALOOF M A. Dynamic weighted majority: An ensemble method for drifting concepts[J]. Journal of Machine Learning Research, 2007, 8(12): 2755-2790.
[9] PESARANGHADER A, VIKTOR H, PAQUET E. Reservoir of diverse adaptive learners and stacking fast hoeffding drift detection methods for evolving data streams[J]. Machine Learning, 2018, 107(11): 1711-1743.
[10] OLORUNNIMBE M K, VIKTOR H L, PAQUET E. Dynamic adaptation of online ensembles for drifting data streams[J]. Journal of Intelligent Information Systems, 2018, 50(2): 291-313.
[11] REN S, LIAO B, ZHU W, et al. Knowledge-maximized ensemble algorithm for different types of concept drift[J]. Information Sciences, 2018, 430: 261-281.
[12] CANO A, KRAWCZYK B. Kappa Updated Ensemble for drifting data stream mining[J]. Machine Learning, 2020, 109(1): 175-218.
[13] RAMÍREZ-GALLEGO S, KRAWCZYK B, GARCÍA S, et al. A survey on data preprocessing for data stream mining: Current status and future directions[J]. Neurocomputing, 2017, 239: 39-57.
[14] LOSING V, HAMMER B, WERSING H. KNN classifier with self adjusting memory for heterogeneous concept drift[C]// 2016 IEEE 16th International Conference on Data Mining (ICDM). IEEE, 2016: 291-300.
[15] ZHOU Z H. Machine learning[M]. Beijing: Tsinghua University Press, 2016: 211-214.
[16] SHANNON C E. A mathematical theory of communication[J]. ACM SIGMOBILE Mobile Computing and Communications Review, 2001, 5(1): 3-55.
[17] BIFET A, HOLMES G, PFAHRINGER B, et al. Moa: Massive online analysis, a framework for stream classification and clustering[C]// Proceedings of the First Workshop on Applications of Pattern Analysis. PMLR, 2010: 44-50.
[18] DOMINGOS P, HULTEN G. Mining high-speed data streams [C]// Proceedings of the sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2000: 71-80.
[19] AGRAWAL R, IMIELINSKI T, SWAMI A. Database mining: A performance perspective[J]. IEEE Transactions on Knowledge and Data Engineering, 1993, 5(6): 914-925.
[20] LANGLEY P, IBA W, THOMPSON K. An analysis of Bayesian classifiers[C]// AAAI. 1992: 223-228.
[21] OZA N C, RUSSELL S. Experimental comparisons of online and batch versions of bagging and boosting[C]// Proceedings of the seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2001: 359-364.

XIA Yuan, born in 1995, postgraduate. His main research interests include data mining and so on.

ZHAO Yun-long, born in 1975, Ph. D, professor, is a member of China Computer Federation. His main research interests include pervasive computing, collective computing, wearable computing and swarm intelligence.

(责任编辑:喻藜)