文章编号:0253-2778(2017)7-0087-08

基于信息熵的数据流自适应集成分类算法

孙艳歌[1,2],王志海[1],原继东[1],白 洋[1]

(1.北京交通大学计算机与信息技术学院,北京 100044;
2.信阳师范学院计算机与信息技术学院,河南信阳 464000)

摘要:数据流分类模型是面向连续变化的实时分析的基本问题.目前大多数的数据流算法只针对突变式或渐变式概念漂移进行处理的,并未充分考虑概念会重现的特点.为此提出了一种具有概念漂移检测机制的自适应集成算法.从信息熵的角度出发,用Jensen-Shannon散度度量相邻两个窗口间数据分布的距离,不仅能检测出不同类型的概念漂移,且能有效地发现重现的概念;采用分类器池机制来保存历史概念,从而实现对概念的重用.将所提出的算法与几种经典的学习算法,在人工合成和真实数据集上进行了广泛的对比实验.实验结果表明,所提出的算法在平均分类准确率上具有明显的优势,比其他集成算法消耗更少的时间,适合多种类型概念漂移的环境,并具有较高的抗噪性.

关键词:数据流;概念漂移;集成分类器;信息熵;重复概念

中图分类号:TP391 **文献标识码**:A doi:10.3969/j.issn.0253-2778.2017.07.010

引用格式:孙艳歌,王志海,原继东,等. 基于信息熵的数据流自适应集成分类算法[J]. 中国科学技术大学学报,2017,47(7):323-330.

SUN Yange, WANG Zhihai, YUAN Jidong, et al. Adaptive ensemble based on entropy for data streams[J]. Journal of University of Science and Technology of China, 2017,47(7):323-330.

Adaptive ensemble based on entropy for data streams

SUN Yange[1,2], WANG Zhihai[1], YUAN Jidong[1], BAI Yang[1]

(1.*School of Computer and Information Technology, Beijing Jiaotong University, Beijing* 100044, *China*;
2.*School of Computer and Information Technology, Xinyang Normal University, Xingyang* 464000, *China*)

Abstract: The processing of streaming data implies new requirements concerning limited amount of memory, small processing time, and one scan of incoming instances. Most of the approaches in the literature to deal with concept drift only focus on gradual or abrupt concept drift and have not addressed the problem of recurring concepts. Motivated by this challenge, an ensemble with internal change detection was proposed to enhance performance by exploring the recurring concepts. It is done by maintaining a pool of classifiers, which dynamically adds and removes classifiers in response to the change detector. The algorithm adopts a two window change detection model, which adopts the Jensen-Shannon divergence to measure the distance of the distributions between two consecutive windows. When a change is detected, the repository of stored historical concepts is checked for reuse. The proposed algorithm has been

收稿日期:2016-08-28;**修回日期**:2016-12-08

基金项目:国家自然科学基金(61672086),河南省科技计划(172102210454),北京交通大学人才基金(2016RC048),信阳师范学院青年骨干教师计划(2016GGJS-08)资助.

作者简介:孙艳歌,女,1982年生,博士生/讲师.研究方向:数据挖掘和机器学习,E-mail:13112074@bjtu.edu.cn.

通讯作者:王志海,博士/教授. E-mail:zhhwang@bjtu.edu.cn

experimentally compared with the state-of-the-art algorithms on synthetic and real datasets. The results show the suitability of the proposed algorithm for different types of drift as well as static environments.
Key words: data streams; concept drift; ensemble classifier; entropy; recurring concepts

0 引言

传感器网络异常检测、信用卡欺诈行为监测、天气预报和电价预测等众多实际问题中，数据都是以流的形式不断产生的.这种快速到达、实时、连续和无界的数据序列称为数据流(data streams)[1-4].真实的数据流环境中，数据分布会随时间的变化发生改变，这种现象说明数据流本质上可能具有不稳定性[5].例如，天气预报所依据的规律可能会随着季节变化而发生改变；顾客网上购物偏好分析方法可能会随顾客群体的兴趣、商家信誉、服务类型等因素的变化而改变；工业用电量会随着季节交替出现周期性变化.一般地，把这种数据流中的数据分布随着时间以某种方式发生变化的现象称为概念漂移(concept drift)[6-8].基于此，许多实际应用问题需要我们研究开发一种特定的面向数据流变化特征的学习机制来快速、实时地加以处理.

根据改变速度可以把概念漂移分为突变式(abrupt concept drift)和渐变式(gradual concept drift)[6].若在较短的时间内，数据流中数据分布突然地被另一个完全不同的数据分布取代，则称此时数据流中发生了突变式概念漂移.此类型的漂移通常在毫无征兆的情况下发生(如传感器突然发生故障)，会使准确率急剧下降甚至模型完全失效.渐变式概念漂移则是一种慢速率改变(如传感器逐渐失灵)，通常是经过较长一段时间后才能观察到，且概念漂移发生前后概念之间有或多或少的相似.在现实环境中，数据流中概念重复出现是普遍存在的.重现式概念漂移[5](recurring concept drift)是一种特殊类型的概念漂移，除了兼具上述两种漂移的特点外，某种概念会有规律或无规律地会重复出现，使得分类模型需要不断地重复训练以适应这种变化.例如，一年四季的用电量数据会随着季节周期变化；社交网络中某一话题可能在固定的时间(如节日或选举)周期出现.

概念漂移是数据流挖掘中的挑战问题，近年来，针对概念漂移问题国内外学者做了大量工作，主要分为基于实例选择、基于实例加权及集成学习3种方法[6].这些算法大多数只针对某一类型的概念漂移进行处理，并未充分考虑概念会重复出现的情况.对此类型概念漂移，要求模型能够使用历史数据，并且当重复概念发生时能够使用以往训练过的模型进行分类，从而避免重复训练.一个理想的分类模型应能增量式地学习并能适应多种类型的变化，因此设计出能应对多种类的概念漂移的分类算法具有重要的研究意义.集成方法通过在不同时段数据来训练个体分类器来保留历史概念，因此是一种有效的处理概念漂移的方法.本文主要关注如何构建面向数据分布规律随时间变化的数据流集成分类模型.

对于概念漂移，有两个问题亟待解决：一是如何快速准确地检测到概念漂移；二是检测到漂移后如何根据不同类型的变化来修正模型以适应这些变化.为此，本文设计并实现了一种能应对多种概念变化的自适应集成算法.主要贡献如下：

(Ⅰ)针对第一个问题，提出了基于信息熵的概念漂移检测方法.从信息熵的角度出发，通过Jensen-Shannon散度来度量新旧窗口之间数据分布的距离，不仅能检测出概念漂移，且能有效地识别重复概念.

(Ⅱ)针对第二个问题，设计了一种分类器池的机制，当检测到概念漂移后，若是新概念则加入分类器池，若是重复概念则重用已有分类器.

(Ⅲ)提出了一种可以同时检测概念漂移和利用重复概念的集成算法，并在人工合成和真实数据流上进行了实验，从分类准确率，运行时间以及抗噪性等多角度进行考察，验证了本文提出方法的可行性和有效性.

1 面向数据流的集成式分类研究背景

1.1 模型描述及相关概念

数据流可表示为 $S=\{s_1, s_2, \cdots, s_t, \cdots\}$，其中 $s_t=(x_t, y_t)$ 为 t 时刻($t=1, 2, \cdots, T$)的实例，$x_t \in \mathbb{R}^d$ 是特征向量，$y_t \in \{c_1, c_2, \cdots, c_k\}$ 是类值，k ($k>1$)为 S 中包含的类值数.根据源源不断到达的实例，增量式学习产生了一个分类器 f，用其对类别标记未知的实例预测类值.

若数据流中数据分布随着时间以某种方式发生变化，则称在该数据流中发生了概念漂移现象.从贝叶斯学习理论的角度来讲，在 t_0 到 t_1 时刻发生了概念漂移可定义为[7]

$$\exists X: P_{t_0}(X,y) \neq P_{t_1}(X,y).$$

式中，$P_{t_0}(X,y)$ 表示 t_0 时刻一组输入变量 X 与目标变量 y 的联合概率分布.

1.2　相关研究工作

目前的概念漂移检测方法大多是根据模型的分类错误率来检测数据分布的变化，如 Gama 等[9]提出的通过监控当前模型的错误率来检测变化的 DDM 算法(drift detection method)，但不能有效地检测到渐变式概念漂移.随后，Baena-García 等[10]提出了基于伯努利分布检测概念漂移方法(early drift detection method，EDDM)，在保证了能对突变式概念漂移的检测的同时，提高了算法对渐变概念漂移的检测效果.Nishida 等[11]提出 STEPD 算法，通过采集训练样本的分类准确率和对全部训练样本的分类准确率来检测概念漂移.Bifet 等[12]提出的自适应滑动窗口算法(adaptive windowing，ADWIN)，通过比较不同子窗口之间的错误率的均值的差异来判断是否发生概念漂移.Ross 等[13]提出了 ECDD (EWMA for concept drift detection)算法，利用指数加权移动平均控制图(EWMA)来监控错误率，若错误率超过一定阈值，则说明发生了概念漂移.

以上算法大多并未考虑概念会重复出现的问题.早在 1996 年 Widmer 就提出了概念会重复出现的问题，直到最近几年才得到学术界的关注.Widmer 等[5]提出 FLORA3 算法，将历史的概念描述保存起来，当概念重现时，保存的分类器被重新激活.Nishida 等提出了一种在线集成算法(adaptive classifier ensemble，ACE)来应对重复概念的出现.类似的方法有 Ramamurthy 等[14]提出的一种基于集成学习方法(ensemble building，EB).EB 算法在数据块序列上构建一组全局的分类器，该方法不会删除历史分类器，而是有选择地从全局分类器中挑选相关的分类器.Katakis 等[15]一种采用聚类找到新概念的，基于概念表示模型.Yang 等[16]把概念看作为 Markov 链中的状态，从概念变换的过程中学习概念转移的规律，并通过 Markov 模型描述概念转换，选择一个与当前概念最相似的概念.Gama 等[17]采用了两层分类器的模型，第一层根据当前概念训练分类器，而其中第二层则是根据已有的概念创建分类器.当检测到概念漂移发生时，则重用第二层的分类器.Gonçalves 等[18]提出了一个处理重复性概念漂移的通用框架(recurring concept drifts，RCD)通过多元非参数统计的方法来识别新旧概念是否来自同一分布.

2　基于信息熵的自适应集成分类算法

我们将首先介绍基于信息熵的概念检测算法，然后描述本文所提出的具有概念检测机制的自适应集成算法.

2.1　基于信息熵的概念检测算法

由于数据流具有无界的特点，因此只能选取数据流中的一部分数据来学习.数据流中滑动窗口(sliding window，SW)是指在数据流上设定一个区间，该区间包含数据流最新的数据，目的为了更好地获取当前数据的特征[19].比较不同子窗口之间数据分布的变化是一种常用的概念检测方法.

信息论中，相对熵(relative entropy)又称 Kullback-Leibler(KL)散度，是衡量相同事件空间 X 里两个概率分布相对差距的测度.两个概率分布 $p(x)$ 和 $q(x)$ 的相对熵定义为

$$\mathrm{KL}(p \,||\, q) = \sum_{x \in X} p(x) \log \frac{p(x)}{q(x)} \tag{1}$$

Kullback-Leibler 散度不满足对称性，因此并不是严格的距离概念.Jensen-Shannon 散度是一种基于 Kullback-Leibler 散度的距离度量，它解决了 Kullback-Leibler 散度的不对称问题.信息论中的 Jensen-Shannon 散度(JSD)能很好地表示两个数据分布之间的关系，因此本文提出一种基于 Jensen-Shannon 散度的概念检测算法，通过比较两个相邻窗口中数据分布是否存在显著性差异来检测概念漂移.Jensen-Shannon 散度的定义为

$$\mathrm{JSD}(P \,||\, Q) = \sum_{x \in \chi} \left(p(x) \log \frac{2p(x)}{p(x)+q(x)} + q(x) \log \frac{2q(x)}{p(x)+q(x)} \right) \tag{2}$$

基于两个窗口的检测模型如下：用 $W_1 = \{x_{t+1}, x_{t+2}, \cdots, x_{t+n}\}$ 和 $W_2 = \{x_{t+n+1}, .\cdots, x_{t+2n}\}$ 分别表示 t 时刻两个连续的大小相等的窗口，W_1 表示参考窗口，W_2 表示当前窗口.用 $\mathrm{JSD}(W_1 \,||\, W_2)$ 度量两个窗口之间分布的距离，当此值小于等于 10^{-5}(非

常接近于零)时,表示两个窗口的数据分布相同,即发现重复概念;当大于 10^{-5} 小于阈值 τ 时,认为两个窗口之间的分布无显著性差异;大于阈值则表明此时有概念漂移发生.阈值采用文献[20]中提出的Bootstrap的方法计算得到.由于窗口每次向前滑动一个实例,因此能及时检测到突变式概念漂移.伪代码如算法2.1所示.

算法 2.1 基于信息熵的概念检测算法

输入:数据流 S;窗口大小 n;
输出:change alarm
01 t=0;
02 $W_1=\{x_{t+1}, \cdots, x_{t+n}\}$;
03 $W_2=\{x_{t+n+1}, \cdots, x_{t+2n}\}$;
04 for 所有实例 $x_t \in S$ do
05 由公式(2)计算得到 $JSD(W_1||W_2)$;
06 采用 Bootstrap 计算得到阈值 τ;
07 if $JSD(W_1||W_2) > \tau$
08 then alarm a change detect at time t;
09 else if $JSD(W_1||W_2)==0$
10 then alarm a recurring concept;
11 end if;
12 end if;
13 t++;
14 go to 02;
15 end for.

2.2 基于信息熵的自适应集成分类算法

考虑到概念的重复性,识别历史概念的代价比创建新的概念模型小的多,因此有必要存储数据流中历史概念的基本信息.本文采用分类器池来保存历史概念,每个分类器表示一个概念,当检测到重复概念时,快速调出相关信息进行处理,减少不必要的重复训练.这需要增加一个内部的概念检测方法来增加算法对概念漂移的适应性,提出的具有概念漂移检测机制的自适应集成分类算法(ensemble with internal change detection,ECD).基于以上分析,本文提出的算法由以下两个功能模块组成:概念漂移检测模块和集成模块.只有在检测到新概念时才重建新的分类器并放入分类器池中,防止重复概念出现导致的重复训练,减少模型更新频率,提高模型实时分类能力和分类效果.

具体说来,用 $E=\{C_1, C_2, \cdots, C_k\}$ 来表示由 k 个分类器组成的分类器池,同时每个分类器还附带变量用于记录该分类器被重复使用的次数,$B=\{B_1, B_2, \cdots, B_k\}$ 表示存储相应的数据,C' 表示建立的新分类器.采用滑动窗口模型来维护最新的数据,对于源源不断到达的实例 (x_i, y_i),W_1 表示参考(旧)窗口,W_2 表示当前窗口.通过比较新旧两个窗口数据分布的距离来检测概念漂移,当检测到有概念漂移发生,就与分类器池中分类器的数据分布进行比较,若是新概念则新建一个新分类器加入到分类器池中,并把相应的数据放在缓存区;若是重复概念则重用已有分类器.分类器按照重复使用的频率从高到低排序,当分类器池中存放的分类器数达到最大值时,则替换最不经常使用的分类器.接着根据每个基分类器 $C_i(i=1, 2, \cdots, k)$ 在最新窗口中实例的分类错误率,按

$$\left.\begin{aligned} &\mathrm{Weight}(C_i)=\mathrm{MSE}_r-\mathrm{MSE}_{ij}\\ &\mathrm{MSE}_r=\sum_{yi} p(y_i)(1-p(y_i))^2\\ &\mathrm{MSE}_{ij}=\frac{1}{|W_2|}\sum_{(x_i,y_i)\in W_2}(1-f^i_{y_i}(x_i))^2 \end{aligned}\right\} \quad (4)$$

对其进行加权,加权投票方式对每个实例进行预测.式中,MSE_r 为随机预测分类器的均方差,MSE_{ij} 为基分类器 C_i 在当前窗口上预测的均方差,$f^i_{y_i}(x_i)$ 表示在分类器 C_i 中预测属性值为 x_i 的类值为 y_i 的概率,$p(y_i)$ 为 y_i 的先验概率.这种情况下,通过在分类器池中搜索代表当前概念的分模型,不但减少了计算代价,而且提高了对概念漂移的适应.伪代码如算法2.2所示.

算法 2.2 ECD 算法伪代码

输入:数据流 S;基分类器数目 k;
输出:包含 k 个分类器的集成分类器 E
01 begin
02 E←∅;
03 B←∅;
04 for 所有实例 $x_t \in S$ do
05 $W_2 \leftarrow W_2 \cup \{x_t\}$;
06 if 检测到概念漂移 then
07 根据最新窗口数据建立新分类器 C′;
08 else if 检测是重复概念 then
09 从分类器池中取出分类器用于分类,分类器的计数器加1;
10 if |E|<k then E←E ∪ C′,C′的计数器加1;
11 else 替换最不常用的分类器;
12 end if;
13 end if;
14 end if;
15 end for;

```
16      用 E 对 x_t 进行预测;
17    t++;
18 end.
```

3 实验

本文是在 CPU 为 2.8 GHZ,内存为 8 GB,操作系统为 Windows 7 的 PC 机上进行实验,所有算法均在大规模数据在线分析开源平台(massive online analysis,MOA)[21]下实现.

3.1 数据集描述

3.1.1 人工合成数据集

实验选取 3 个人工合成数据集对本文提出模型进行验证,如表 1 所示.

表 1 人工合成数据集合基本信息

Tab.1 Characteristic ofsynthetic datasets

名称	实例数	属性数	类值数	噪声	漂移数	类型
HyperPlane	1M	10	2	5%	1	渐变
SEA	1M	3	4	10%	3	突变、重现
Waveform	1M	40	3	无	0	无

用 MOA 中数据流生成器生成 3 种类型概念漂移:突变式、渐变式和重现式概念漂移.

HyperPlane 是使用最广泛的数据流数据集[2, 7, 12],该数据集通过改变数据样本属性的权值来模拟概念漂移现象.实验采用 MOA 中的数据流生成器 HyperplaneGenerator 生成实例改变概率为 0.001 的渐变式概念漂移数据集.

SEA 由 Street 于 2001 年提出[12],是经典的突变式概念漂移数据集.其基本结构为$<f_1, f_2, f_3, C>$,其中 f_1、f_2 及 f_3 为条件属性,C 为类属性,仅 f_1, f_2 及 C 相关.当实例属性满足 $f_1+f_2\leqslant\theta$ 时,属于第一类;否则属于第二类.实验首先采用数据流生成器 SEAGenerator 生成包含 3 次突变的数据集,分别在 250 K,500 K 和 750 K 处发生;然后采用能生成重复概念的数据流产生器 RecurrentConceptDriftStream 生成包含 3 个重复概念的数据集.

Waveform 数据集有 3 种基准波形(每一个基准波形由 21 个数值型属性组成),每一类别都是其中两种或 3 种的结合.实验采用数据流生成器 WaveformGenerator 生成具有 40 个数值型属性波形数据流数据集,其中包括 19 个不相关属性.

3.1.2 真实数据集

Emailing list(Elist)是包含突发概念漂移和重复概念的数据集[15],Spam filtering(Spam)则是包含渐变概念漂移的数据集,两个数据集都是以布尔型词袋模型表示.数据集均可在 http://mlkd.csd.auth.gr/concept_drift.html 下载,用 MOA 中的 ArffFileStream 生成器将静态的数据模拟生成数据流.

Elist 模拟了连续不断的来自不同领域的各种邮件信息,用户可以根据兴趣把这些邮件标记成为垃圾或者感兴趣.共包括 1 500 个实例和 913 个属性,数据被划分为 5 个阶段,通过用户的兴趣发生的变化来模拟概念漂移的出现.表 2 描述每个阶段中哪个类型被认为感兴趣或者垃圾邮件,其中(+)表示感兴趣,(−)表示一个垃圾邮件.用 C_1 表示用户只对医学类的邮件感兴趣,C_2 表示用户对航空和棒球感兴趣,则此数据流表示 C_1, C_2, C_1, C_2, C_1 概念序列.

表 2 Emailing list 数据集描述

Tab.2 Characteristic of Emailing list dataset

	1～300	300～600	600～900	900～1200	1200～1500
Medicine	+	−	+	−	+
Space	−	+	−	+	−
Baseball	−	+	−	+	−

Spam 包括 9 324 个实例和 500 个属性,每个实例表示一个邮件的信息,被分为两种类型:垃圾邮件(仅占 20%)和合法邮件.数据集中的垃圾邮件的特征随着时间缓慢变化.

3.2 参数设置与实验结果分析

实验采用文献 Prequential 评价策略[21],即每条实例先作为测试数据而后作为训练数据,这样准确率是增量更新的.采用这种评价策略不用扣留数据集来测试,从而保证最大化利用每个数据的信息,也保证准确率随时间具有平滑性.

3.2.1 参数对算法性能影响

在 SEA 数据集上进行测试,分别测试滑动窗口大小 n 取值为[500, 2000],以此验证窗口大小设置对算法性能的影响.由图 1 可以看出,一开始,随着窗口的扩大,构建分类器的数据增多,分类准确率也随之上升;随着窗口的继续增大,概念漂移的发现滞后,此时,分类器的分类准确率达到了瓶颈,因而平

均分类准确率略有降低.表 3 表明本文所提的方法受窗口大小设置影响很小,当窗口大小为 1 000 时,算法可以获得相对较高的分类准确率.

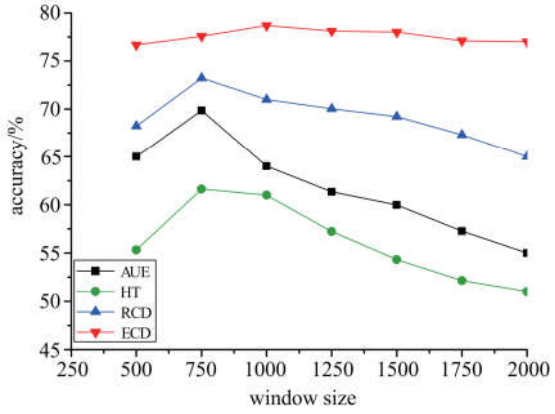

图 1 不同窗口大小的分类准确率比较

Fig.1 Accuracy in different window sizes

表 3 不同窗口大小下的分类准确率

Tab.3 Accuracy in different window sizes

Dataset	window size						
	500	750	1000	1250	1500	1750	2000
HyperPlane	71.69	72.31	72.23	72.01	71.61	72.01	71.25
SEA	76.68	77.59	78.69	78.12	78.02	77.12	77.01
Waveform	81.16	81.25	81.36	81.54	80.95	81.53	82.01
Elist	72.23	72.39	72.59	72.63	72.15	72.09	72.03
Spam	93.16	93.20	93.26	93.19	93.08	93.15	93.03

3.2.2 不同算法性能比较分析

本文算法与以下 3 个算法进行对比:Hoeffding Tree(HT)、RCD 和 accuracy update ensemble (AUE).HT 是一种从数据流中增量式生成决策树的单分类器算法,利用 Hoeffding 不等式理论,能够以一定的概率保证所构造决策树的精度.文献[2]提出 AUE 算法,在固定大小的数据块上不断建立分类器,采用非线性的方式对分类进行加权.

HT 和 AUE 在 MOA 下实现,RCD 可在 https://sites.google.com/site/moaextensions/获取.为了便于比较,分类器池中分类器数 $k=15$,进行比较的集成分类算法的基分类器均采用 Hoeffding Tree,选取与文献[22]相同的参数:叶子结点采用 adaptive Naïve Bayes 预测类值,其中 $n_{\min}=100$,置信度 $\delta=0.01$,$\tau=0.05$.分别从分类准确率和运行时间两个方面进行比较.

表 4 展示了算法在 5 个数据集上的分类准确率.总体来看,集成算法获得了比单分类器算法更高的分类准确率.HT 算法在包含概念漂移的数据集上均表现最差,这是由于其没有任何处理概念漂移机制,因此不适合概念漂移环境.在没有概念漂移数据集 Waveform 上,由于数据分布相对比较稳定,所有算法差别并不大,本文算法并没有明显优势.在渐变式数据集 HyperPlane 上,AUE 获得了最高准确率,其次是本文算法.究其原因,是由于 AUE 算法在最新数据块上不断训练生成分类器,能及时应对渐变式概念漂移.在包含重复概念的数据集 SEA 上,本文算法获得了最高的准确率,这是由于增加了重复概念漂移检测机制,能及时建立新的分类器来适应突然的概念变化.在真实数据集 Elist 上,本文算法表现最好.在 Spam 上表现最好的是 RCD,其次是本文算法.

表 4 不同算法分类准确率(%)

Tab.4 Comparison of classification accuracy(%)

Dataset	RCD	AUE	HT	ECD
HyperPlane	67.21 (3)	79.58 (1)	63.47 (4)	72.23 (2)
SEA	73.24 (2)	69.86 (3)	60.63 (4)	84.69 (1)
Waveform	82.27 (4)	83.13 (1)	82.79 (2)	82.36 (3)
Elist	65.36 (2)	55.07 (3)	51.45 (4)	72.59 (1)
Spam	90.53 (1)	89.35(3)	75.36 (4)	90.26 (2)
Average Rank	2.4	2.2	3.6	1.8

运行时间的对比如表 5 所示.通过比较发现,HT 的运行时间最短,其次是本文算法,而 AUE 时间消耗最长.这是由于本文算法是一种基于数据分布的检测算法,可以快速检测到概念漂移和识别重复概念,并且选择一个已有的模型,避免了重复训练,因此在时间上要有优势.单分类器算法 HT 虽然速度最快,但却在分类准确率上表现最差.

表 5 不同算法运行时间(s)

Tab.5 Comparison of time consumption

Dataset	RCD	AUE	HT	ECD
HyperPlane	14.40 (2)	153.08 (4)	13.47 (1)	15.35 (3)
SEA	4.56 (2)	54.05 (4)	6.41 (3)	3.87 (1)
Waveform	8.38 (1)	84.13 (4)	11.42 (3)	10.21 (2)
Elist	1.36 (3)	2.43 (4)	0.98 (1)	1.24 (2)
Spam	17.36 (4)	13.45 (3)	4.36 (1)	7.02 (2)
Average Rank	2.4	3.8	1.6	2.0

观察图 2 发现,在 SEA 数据集中发生了 3 次突