文章编号:1671 - 6833(2017)04 - 0057 - 08

基于图的概念重现发现与预测

白 洋[1], 王志海[1], 孙艳歌[1,2]

(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: 概念漂移是数据流挖掘中具有挑战性的问题. 当概念漂移发生后,原有分类模型的分类正确率会显著下降,因此需要及时发现并调整模型以适应这些改变. 概念重现是概念漂移的特殊情况,然而已有的算法大多未能充分考虑这种状况. 为此,提出一种能够处理重现的概念检测方法. 试验结果表明,该方法能够以较低的延迟和较低的误报率检测到概念漂移,并且可以识别重现的概念,很大程度上提升了分类器的分类正确率.

关键词: 数据流;数据挖掘;概念漂移;漂移检测;概念重现

中图分类号: TP311 **文献标志码:** A doi:10.13705/j.issn.1671 - 6833.2017.01.021

0 引言

数据挖掘的基本问题是处理随时间增长的大量数据,待处理的海量数据都以高速有序的形式到达,此类数据称为数据流[1]. 在动态变化和不平稳的环境中,数据分布会随着时间改变从而产生概念漂移现象[2]. 在这里概念就是指分类问题中输入变量 X 和目标变量 y 之间的联合概率. 那么,概念漂移就是指输入变量和目标变量之间联合概率的变化[3],如下式所示:

$$P(X^{t_0}, y_1^{t_0}) \neq P(X^{t_1}, y_2^{t_1}). \quad (1)$$

式(1)定义了时间 t_0 和 t_1 之间的概念漂移. 其中可能是 $P(y)$ 发生了改变,也可能是 $P(X \mid y)$ 发生了改变. 现实生活中有许多关于概念漂移的例子. 例如,在垃圾邮件分类问题中,垃圾邮件的发送者为了避免自己发送的邮件被过滤掉,会对邮件中的一些关键字进行修改,从而躲避检测. 或者是邮件的接收者对于某一类邮件的态度的转变,某类邮件在某段时间被接收者认定为垃圾邮件,可能过了一段时间之后接收者突然转变对此类邮件的态度,从而变得愿意接收此类邮件. 又例如,当预测超市一周的销售额时,往往会根据广告投入和促销活动等因素进行预测,但是建立的预测模型不可能一直都保持较高的预测正确率,因为销售额可能会受假期等因素的影响. 影响销售行为的因素会随时出现,因而无法避免. 故而在处理有概念变化的数据时,要注意当数据流中出现变化时模型的自适应问题. 概念漂移检测能够及时捕捉到数据流的变化,并在发生变化后更新模型,使分类模型能够保持较高的分类正确率.

当出现概念漂移现象时,漂移后产生的概念可能是从未出现过的新概念,也可能是曾经出现过的概念[4]. 概念重现[5]是概念漂移的特殊情况,它是指一个新概念出现在数据流中,然后消失一段时间之后再次出现的现象. 在概念重现的情况下,以前出现过的概念会在将来再次出现,因此旧的分类模型可以用于将来的分类. 但是现有检测概念漂移的文献大多会忽略这种现象,它们总是把漂移后产生的概念当做是一个新概念,没有考虑这个概念是否在过去出现过.

针对这一问题,笔者首先提出一种概念漂移的检测方法(distribution-based detection method, DBDM),并在此基础上提出了一种检测概念重现的方法(recurrent detection and prediction, RDP),此方法检验当前是否发生了概念重现,并有效地提高了分类正确性.

收稿日期:2016 - 11 - 03;**修订日期:**2016 - 12 - 14

基金项目:国家自然科学基金资助项目(61572417,61572005);北京市自然科学基金资助项目(4142042);信阳师范学院青年骨干教师资助计划项目(2016GGJS-08).

通信作者:王志海(1963—),男,河南安阳人,北京交通大学教授,主要从事数据挖掘和机器学习等方面的研究,E-mail:14120375@bjtu.edu.cn.

1 相关工作

在数据流的研究领域中,对于概念漂移的处理可分为基于触发的方法和逐渐演化的方法[6].基于触发的方法利用明确的检测方法来指出概念变化,并根据检测的结果来决定是否更新分类模型.然而逐渐演化的方法不检测变化,这种方法通常会保持一组分类模型,并基于它们的性能定期进行更新.

Gama 等提出的 DDM 算法[7]通过监控当前模型的错误率来检测变化. Baena-Garcia 等发现 DDM 方法在检测渐变的概念漂移时效果不佳,为此提出了 EDDM 算法[8],提高了检测渐变概念漂移的效果. ADWIN 方法[9]中滑动一个固定的检测窗口,存放最近读入的数据,采用指数直方图的变形作为数据结构,检测窗口中的数据变化. Ross 等[10]提出了 EWMA(exponentially weighted moving average charts)算法,利用指数加权移动平均控制图(exponentially weighted moving average charts)监控错误率,当错误率超过一定阈值,则说明发生概念漂移. Sakthithasan 等[11]和 Pears 等[12]提出了 SeqDrift 检测算法,该方法采用蓄水池抽样来进行内存管理,很大程度上提高了在缓慢变化情况下的检测灵敏度.

SEA 算法[13]是最早利用从数据流中学习得到的分类器集成处理概念漂移的算法,根据简单多数投票做出预测. EB 算法[14]也是一种集成方法,它利用连续的数据块来创建分类器,根据每个分类器在最后一个数据块上的分类正确率来决定该分类器的权值,它也能处理概念重现的现象. OCBoost[15]是一种在线 boosting 算法,它将几种其他的在线 boosting 算法互相联系起来以获得更好的性能. DDD 算法[16]是基于分类器差异性的动态集成方法,算法中还使用了内部漂移检测来加速自适应. ADACC 算法[17]是一种主动处理概念漂移现象并且能够处理重现概念的集成方法. OAUE 算法[18]根据每个基分类器在连续的时间和空间中的错误率来权衡它们的性能. PDSRF 算法提出了一种基于随机森林算法的决策树集成算法来处理概念漂移数据流[19].

在最新的关于概念重现的文献中,Katakis 等[20]采用数据流聚类的方法构造和更新分类器的集成,当一批新实例到达,识别这批数据属于哪一个概念,应用对应的分类器来预测实例的类标; Gomes 等[21]提出一个基于情境感知的数据流学习系统,根据概念和上下文来添加或删除分类器; Abad 等[22]提出一个解决概念重现问题的框架,用于垃圾邮件的检测问题.

2 基于数据分布的漂移检测算法

笔者提出的基于数据分布的概念漂移检测算法通过监控不同时间窗口中的数据分布来检测是否有漂移产生,本章主要介绍这种监控数据分布的方法.

2.1 问题描述

在数据流环境中,每到来 M 个数据,就把这 M 个数据与之前的数据之间的分界点看做是一个检测点,并把这 M 个数据看做是一个数据块.遍历每一个检测点,检验检测点两侧的数据是否来自同一分布.如果发现它们来自不同的分布,就认为发生了漂移.用 μ_1 和 μ_2 分别表示检测点左右两侧的总体均值,检验假设 $H_0:\mu_1=\mu_2$, $H_1:\mu_1\neq\mu_2$.用 $\hat{\mu}_1$ 和 $\hat{\mu}_2$ 分别表示检测点左右两侧的样本均值,那么当 $\Pr(|\hat{\mu}_1-\hat{\mu}_2|\geqslant\varepsilon)>\delta$ 成立时,就拒绝 H_0,即认为发生了概念漂移.其中 $\delta\in(0,1)$ 是用户定义的显著水平,当 H_0 为真时拒绝 H_0 的概率不超过 δ, ε 是关于 δ 的函数.通常会设置两个 δ,分别为 δ_{warning} 和 δ_{drift}($\delta_{\text{warning}}<\delta_{\text{drift}}$),与之对应的是 $\varepsilon_{\text{warning}}$ 和 $\varepsilon_{\text{drift}}$.当均值差异超过 $\varepsilon_{\text{warning}}$ 时,就认为当前可能有漂移发生;当均值差异超过 $\varepsilon_{\text{drift}}$ 时,就认为发生了概念漂移.

2.2 基于 Bernstein 不等式的动态阈值设计

接下来要计算阈值 ε.通常刻画两个分布之间差异的不等式有 Hoeffding 不等式[23]、Chernoff 不等式[24]、Bernstein 不等式[25]等.其中 Hoeffding 不等式被广泛应用,但是在 Hoeffding 不等式中忽略了方差的作用,这样就使得方差较小时,得到的结果不够精确. Bernstein 公式被用于一些文献中[9,11-12],它将期望和方差联系了起来,因此利用 Bernstein 不等式计算得到的阈值更严谨.笔者利用它们的推导进行分析. Bernstein 不等式为:

$$\Pr\left(\left|\frac{1}{n}\sum_{i=1}^{n}X_i-EX\right|\geqslant\varepsilon\right)\leqslant 2\exp\left(\frac{-n\varepsilon^2}{2\sigma^2+\frac{2}{3}\varepsilon(c-a)}\right),\tag{2}$$

其中, $X_1,X_2,\cdots,X_n$ 是相互独立的随机变量, $X_i\in[a,c]$,由于要讨论的是分类问题,因此取 $a=0$, $c=1$; EX 表示总体均值; σ^2 表示方差.该不等式反映了随机变量偏离期望的概率的上界.要控制 H_0 为真时拒绝 H_0 的概率不超过 δ,只需令

$$\Pr[|\hat{\mu}_1-\hat{\mu}_2|\geqslant\varepsilon]=\delta.\tag{3}$$

根据布尔不等式,可以得到:

$$\Pr[\,|\hat{\mu}_1-\hat{\mu}_2|\geqslant\varepsilon]\leqslant\Pr[\,|\hat{\mu}_1-\mu|\geqslant k\varepsilon]+\Pr[\,|\hat{\mu}_2-\mu|\geqslant(1-k)\varepsilon]\,,\tag{4}$$

其中,k 表示检测点左侧数据占所有数据的比例. 在式(4) 的右边应用式(2)中的 Bernstein 不等式,并将 $a=0$,$c=1$ 代入,得到:

$$\Pr[\,|\hat{\mu}_1-\hat{\mu}_2|\geqslant\varepsilon]\leqslant 2\exp\left(\frac{-n_1(k\varepsilon)^2}{2\sigma_1^2+\frac{2}{3}k\varepsilon}\right)+2\exp\left(\frac{-n_2[(1-k)\varepsilon]^2}{2\sigma_2^2+\frac{2}{3}(1-k)\varepsilon}\right),\tag{5}$$

在右侧应用 Bernstein 不等式得到:

$$2\exp\left(\frac{-n_1(k\varepsilon)^2}{2\sigma_1^2+\frac{2}{3}k\varepsilon}\right)+2\exp\left(\frac{-n_2[(1-k)\varepsilon]^2}{2\sigma_2{}^2+\frac{2}{3}(1-k)\varepsilon}\right)\geqslant 4\exp\left(\frac{-n_1(k\varepsilon)^2}{2\sigma_1{}^2+\frac{2}{3}k\varepsilon}\right),\tag{6}$$

结合式(5) 和式(6),可得到:

$$\Pr[\,|\hat{\mu}_1-\hat{\mu}_2|\geqslant\varepsilon]\leqslant 4\exp\left(\frac{-n_1(k\varepsilon)^2}{2\sigma_1{}^2+\frac{2}{3}k\varepsilon}\right),\tag{7}$$

为了满足式(2),结合式(7),得到:

$$\delta\leqslant 4\exp\left(\frac{-n_1(k\varepsilon)^2}{2\sigma_1^2+\frac{2}{3}k\varepsilon}\right).\tag{8}$$

由于左右两侧数据量差异过大会造成对阈值计算的不准确,因此,在计算均值和方差时采用分层抽样的方法使两侧数据量相等. 按照一定的比例,在数据量较大的一侧从每个块中独立地抽取一定数量的数据,将各块取出的个体合在一起作为样本. 假设检测点左右两侧数据量的比值为 l $(0<l<1)$,那么就从数据量较大的一层按比例 l 抽取数据,即当块的大小为 m 时,从该块中抽取 $l\times m$ 个数据. 采用分层抽样不仅可以让左右两侧数据量保持一致,而且还能保证抽样得到的数据样本的结构和整体结构比较相近,使抽取的数据更具有代表性. 因此 k 值可以用 1/2 替换,由式(8)可以得到 ε 的表达式为:

$$\varepsilon=\frac{2}{3n_1}\left(\ln\frac{4}{\delta}+\sqrt{\left(\ln\frac{4}{\delta}\right)^2+18n_1\sigma_1^2\ln\frac{4}{\delta}}\right).\tag{9}$$

在 DBDM 中,检测到一个变化点之前要进行多次检验. 在多重检验问题中,随着检验次数的增加,错误地拒绝原假设的可能性就会增加. 现有的检测算法常常会利用 Bonferroni 校正法来避免这个问题,但是当检验次数较多时,Bonferroni 校正的效果较差. 因此,笔者对 Sidak 校正法进行一些修改,用修改后的校正法对显著性水平 δ 进行校正. 校正后的 δ'为:

$$\delta'=1-(1-\delta)^{\frac{1}{n}},\tag{10}$$

其中,n 表示假设检验的次数.

2.3　算法描述

本节主要介绍 DBDM 算法的检测过程,具体步骤如下所示.

1) Input:S:data stream
2) M:block size
3) δ_{warning},δ_{drift}
4) Output: true ‖ false
5) flagWarning = false;
6) for each instance in S do
7) numInstance ++ ;
8) ifnumInstance % $M=0$ then
9) for each check point $P_i$do
10) compute mean values μ_1 and μ_2;
11) compute drift and warning significant levels δ'_{drift} and δ'_{warning} using sidak correction;
12) compute drift and warning thresholds $\varepsilon_{\text{drift}}$ and $\varepsilon_{\text{warning}}$;
13) if $|\mu_1-\mu_2|>\varepsilon_{\text{drift}}$ then
14) delete data which are on the left side of P_i;
15) if $\mu_1-\mu_2<0$ then
16) return true;
17) break;
18) if $|\mu_1-\mu_2|>\varepsilon_{\text{warning}}$ then
19) flagWarning = true;
20) returnfalse.

3　概念重现检测与预测算法

笔者提出的 RDP 算法可以发现概念重现这种现象,该方法利用上文提出的 DBDM 检测算法获取当前数据流的状态,并根据当前状态做出不同的处理,最终可以检测到概念重现. 在检测概念重现时利用 k 近邻算法进行比较,并利用一个带权有向图来预测当前出现的概念最有可能是过去出现过的哪个概念的重现.

3.1　基于图表示的历史概念发现

检测概念重现的方法 RDP 的基本思想是:每当检测到数据流达到了 Warning 状态之后,保存一组实例,并开始训练一个新的分类器作为备用分类器. 当检测到 Drift 状态之后检验这组实例和之前保存的实例是否来自同一概念. 如果它们来自同一概念,就说明当前概念是一个重现的概念,那么就用之前存储的适用于这个概念的分类器进行分类;如果不是来自同一个概念,就说明当前概念是一个

未出现过的新概念,那么将备用分类器和代表此概念的一组实例存储下来.在检查两组实例是否来自同一概念时,在代表当前概念的实例中运用 k 近邻算法找出存储的一组实例中每一个实例的 k 个最近邻,如果这 k 个最近邻中的大多数都和该实例属于同一类别,那么就说明它们是来自同一概念.每次都在存储的实例中挑选一组最有可能的实例跟当前的一组实例进行比较,每个存储的概念被重用的可能性大小用一个图来表示.

在该方法中,把存储的所有概念整体看作是一个图,图中的每个顶点都代表一个概念,顶点中存储着与这个概念相关的一组实例和对应的分类器.每次检测到漂移之后,就把以漂移点之前的概念为弧尾、以漂移点之后的概念为弧头的弧的权值加1.并且在下次漂移发生时,优先挑选以当前概念为弧尾的弧中权值最大的那条弧指向的概念作为下一个备选概念.例如,在图1(a)中,如果当前的概念为概念1,一段时间后发生了漂移,由于以概念1为弧尾的所有弧中,权值最大的是以概念3为弧头的那条弧,接下来是以概念4为弧头的弧,最后是以概念2为弧头的弧.所以依次选择概念3、概念4、概念2作为漂移后的概念,如果不成立,再选择概念5,如果仍不成立,就说明漂移后产生了一个没有出现过的新概念.

因此,最后的结果有3种情况:

1)概念2、概念3和概念4之中的某一个为漂移之后的概念,假设为概念2,那么就将从概念1到概念2的弧的权值加1,如图1(b)所示.

2)概念5为漂移之后的概念,那么就在概念1和概念5之间加一条弧,弧的权值为1,如图1(c)所示.

3)漂移后产生了一个之前没有出现过的新概念6,那么就在图中加入一个顶点表示概念6,并在概念1和概念6之间加一条弧,权值为1,如图1(d)所示.

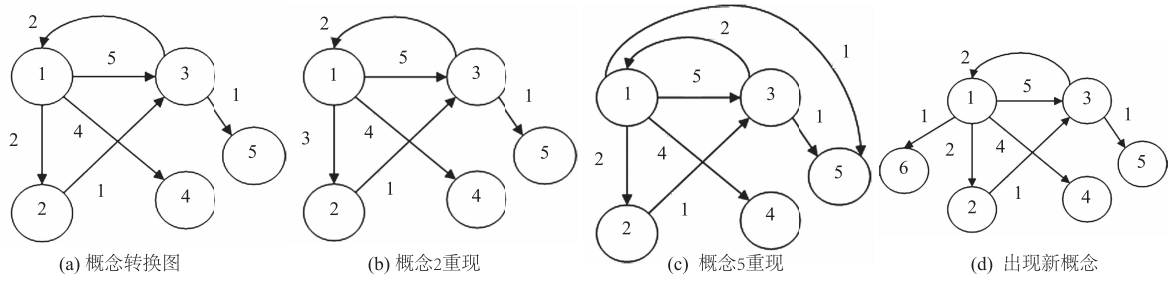

图1 概念图

Fig.1 The concept diagram

3.2 RDP算法描述

这一部分详细介绍该检测方法,其中 G 表示用来存储概念的图,它存储的是和过去出现过的概念相关的所有实例和对应的分类器;V 为图中的顶点,表示过去出现过的一个概念,它存储与这个概念对应的实例和分类器;vexnum 表示当前 G 中的顶点数;C_n 表示产生漂移后适用于新概念的分类器;B_n 是一组实例,它代表可能出现的新概念;p 表示漂移发生之前的概念在 G 中的序号.如下所示算法2描述RDP的具体过程.

01) Input: S: data stream
02) max: the maximum of nodes in G
03) Output: G
04) create a network G;
05) for each instance ins in Sdo
06) DBDM(ins);
07) if current state is Warningthen
08) save ins in B_n;
09) train C_n for later use;
10) elseif current state is Driftthen
11) for each arc in G whose tail is p
12) choose the arc with the maximum weight, it's head is node V_k;
13) if compare (B_n, V_k. instances) then
14) $C_n = V_k \cdot C$;
15) clear(B_n);
16) else
17) create a new node to store B_n and C_n;
18) ifvexnum > maxthen
19) delete one node in G;
20) insert new node into G;
21) C_n replace the current classifier;
22) else
23) clear B_n.

4 实验结果及分析

首先验证笔者提出的概念漂移检测方法在处理突变漂移时产生的误报较低,并且能及时检测

到渐变漂移.然后验证笔者提出的检验重现概念的方法能够检测到重复出现的概念,并且能提高分类正确率.实验环境是 Windows7 操作系统,3.3 GHz CPU,4G RAM.程序在 MOA 平台中实现,MOA 是一个开源的架构,它为数据流的在线学习提供实现算法和运行实验的软件环境[26].

4.1　概念漂移检测算法分析

这一部分的实验验证笔者提出的 DBDM 方法拥有较低的误报率.将 DBDM 方法分别和 DDM[7]、EDDM[8]、EWMA[10]进行比较.DBDM 方法中设置参数 $\delta_{\text{drift}}=0.1$,块大小设为 200;EWMA 中设置参数 $ARL_0=1\ 000$,$\lambda=0.2$.

首先比较当数据流处于稳定状态时,DDM、EDDM、EWMA 和 DBDM 的误报次数.实验中采用包含 200 000 个实例的服从伯努利分布的稳定的数据流,服从伯努利分布的数据流的均值分别设置为 0.05、0.1、0.3、0.5.每个实验重复 100 次,求出误报次数的均值,结果如图 2 所示.结果显示,在稳定的情况下,DDM 的平均误报次数一直都比较少,且一直在下降,由一开始的 1.89 次降低到最后的 0.19 次;EDDM 平均误报次数随着数据流均值的增长,从一开始的 35.56 次降低到了 9.38 次;EWMA 的平均误报次数最多,且在服从伯努利分布的数据流的均值达到 0.5 之前,一直维持较高的误报次数,但是当均值达到 0.5 时,误报次数突然下降到 0.11;DBDM 的误报次数一直都较少,且一直在下降,随着数据流均值的增长,误报次数从 3.39 次降低到了 0.03 次.在图 2 中,描述 DDM 和 DBDM 的误报次数的线条几乎重合,刚开始 DBDM 的误报次数高于 DDM,但是随着均值的上升,最终 DBDM 的误报次数降低到了 DDM 以下.

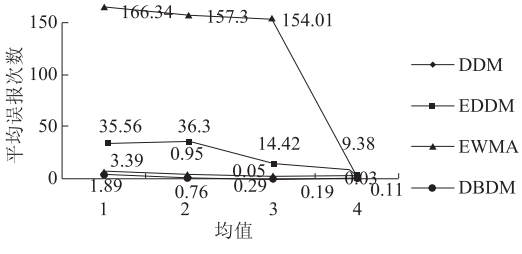

图 2　服从稳定的伯努利分布时的误报

Fig. 2　Average false positive counts on stationary Bernoulli distribution

然后比较在发生突变时检测方法的性能.由于检测到概念漂移后要更新分类器以适应变化后的情况.因此,当误报较高时,就会频繁地更新分类模型,这对资源是一种浪费,在本次实验中,验证在突变情况下 DBDM 的误报次数.用 MOA 数据流产生器产生服从伯努利分布的数据集,实例的总数为 200 000,前 100 000 个实例服从均值为 0.01 的稳定的伯努利分布,然后均值分别从 0.01 突然上升到 0.05、0.1、0.3、0.5.图 3 展示了每个检测算法的误报次数.结果显示 DBDM 在突变的情况下误报次数较少,且当均值增长值从 0.04 变化为 0.09 时,误报次数从 4.99 次降低到了 3.76 次,随后误报次数又呈上升趋势,但是上升的幅度不大;DDM 和 EDDM 的误报次数都随着均值变化幅度的增大而降低,其中 DDM 的误报次数从 11.93次降低到了 8.7 次,EDDM 的误报次数从 46 次降低到了 25.17 次,EDDM 的误报次数一直高于 DDM;EWMA 的误报次数最多,但是也随着变化幅度的增长而降低.

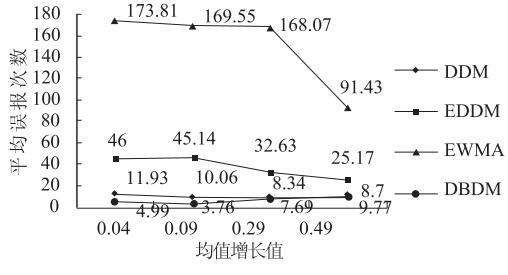

图 3　突变情况下的误报

Fig. 3　Average false positive on abrupt drift

最后比较概念发生渐变时,检测方法的性能.由于渐变的变化很缓慢,没有突变那么明显,因此检测渐变时主要看重的是检测时延,即检测到概念漂移的点与实际发生概念漂移的点之间的实例个数.在这个实验中,实例的总数是 1 000 000 个,前 998 000 个实例是稳定的,均值为 0.0100,从第 998 000 个实例之后,均值分别以 0.000 1、0.000 2、0.000 3 和 0.000 4 为斜率增长,检测时延如图 4 所示.EWMA 在渐变情况下的检测延迟较大,随着斜率的增长虽然检测延迟有所降低,但是降低的不明显;DBDM 的检测延迟一开始较高,然后随着斜率的增长,检测延迟的降低较明显.在实验中,DDM 和 EDDM 在很多情况下不能检测到

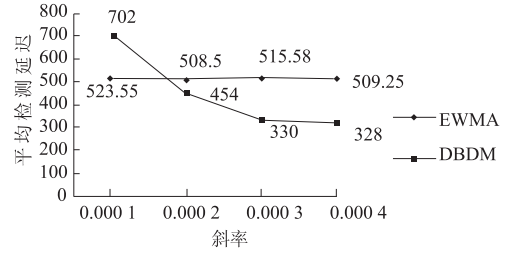

图 4　渐变情况下的检测延迟

Fig. 4　Detection delays on an gradual drift

真正的变化点,因此没有在图中显示它们的检测时延.

4.2 概念重现的发现与分类正确率的提升

在分类器对数据流进行分类时,由于数据流中存在概念漂移现象,分类正确率会受到影响.检测漂移就是为了能及时发现概念变化,然后更新分类模型,以达到提高分类正确率的目的.笔者提出的漂移检测方法能够以较小的延迟和较低的误报率检测到概念漂移,因此,它能够有效地提高分类正确率.接下来的实验就是为了验证这一点,并检验RDP是否能够检测到重现的概念.实验中的分类器采用NaiveBayes,利用4个数据集进行验证,最后将RDP与几种可从MOA中获得源码的处理概念漂移数据流的算法进行比较.

第1个是用随机树生成器生成的数据集,一共产生1 000 000个实例,该数据集包含10个属性,并且包含4个重现的漂移,它们均匀地分布在这1 000 000个实例之中.

第2个是邮件列表数据集,它是一个真实的数据集,包含概念重现的现象.它模拟来自不同主题的邮件信息数据流,连续把信息呈现给用户,然后用户根据自己的兴趣把这些邮件标记为"感兴趣"和"垃圾邮件".邮件列表数据集包含1 500个实例,913个属性,这些属性是在语料库中至少出现10次的单词,还包含2个类值[26].

第3个是由超平面生成器生成的数据,d维空间中的超平面是一个点x的集合,x满足$\sum_{i=1}^{d} w_i x_i = w_0$,其中$x_i$是$x$的第$i$个坐标.满足$\sum_{i=1}^{d} w_i x_i \geqslant w_0$的实例标记为正,满足$\sum_{i=1}^{d} w_i x_i < w_0$的实例标记为负.用超平面模拟随时间变化的概念是十分有用的,因为可以通过改变权值的相对大小来改变超平面的坐标和位置[27].在本次实验中,选择实例数为1 000 000个,属性数量为20,类值的数量为2,漂移的属性数为10,每个样本变化的幅度为0.1.

第4个是森林覆盖数据集,它是一个真实数据集,该数据集包含从美国林务局的资源信息系统的数据中得到的不同类型的森林的地理描述,数据集中包含55个属性,其中有10个数值型属性,44个名称型属性,7个类值,共581 012个实例.

图5显示对于随机树生成器生成的数据集,只利用NaiveBayes进行分类以及结合RDP进行分类这两种情况下的分类正确率.在这次实验中,一共检测到7次概念重现,从图5中可以看出,RDP提高了分类器的分类正确率.

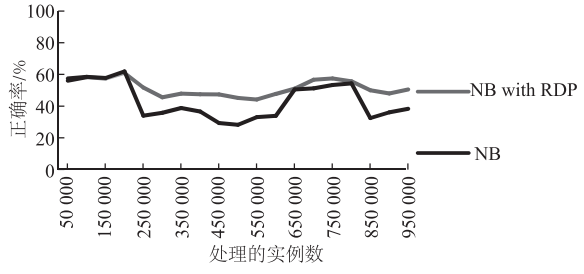

图5 在随机树生成器生成的数据流上的分类正确率

Fig. 5 Accuracy on random tree dataset

图6显示对邮件列表数据集进行分类的分类正确率,可以看到在应用了RDP算法后分类正确率有了明显提升.

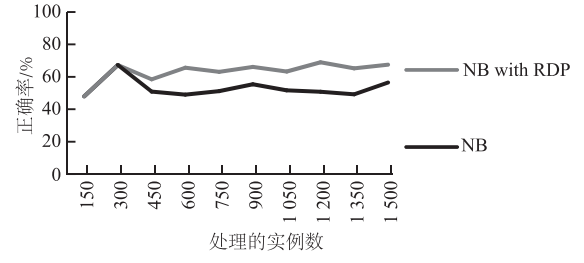

图6 在邮件列表数据集上的分类正确率

Fig. 6 Accuracy on email list dataset

图7显示,在只应用NaiveBayes对超平面数据集分类的情况下,分类正确率一直较低,在第250 000个实例到450 000个实例之间的这一段,分类正确率大幅度下降,说明发生了概念变化,当前的分类器已不适用.应用RDP之后,它可以检测到漂移的发生,及时更新分类器,因此可以保持较高的分类正确率,分类正确率几乎都在80%以上.

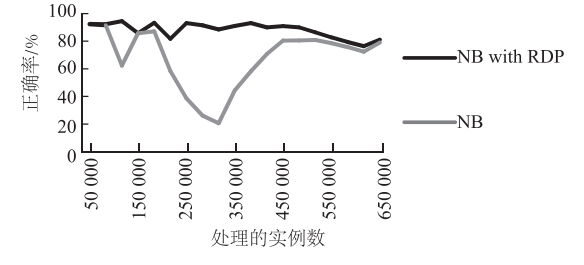

图7 在超平面生成器生成的数据流上的分类正确率

Fig. 7 Accuracy on hyperplanedataset

图8是对森林覆盖数据集进行分类的情况,从图中可以看出,RDP对分类正确率的随着时间产生显著性提升.

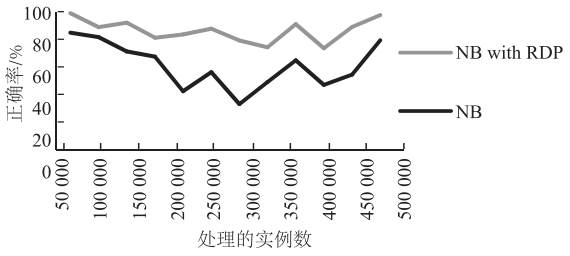

图 8　在森林覆盖数据集上的分类正确率

Fig. 8　Accuracy on forest cover dataset

最后将 RDP 算法和 4 种可从 MOA 中获得源码的经典的处理概念漂移数据流的算法 EB[14]、OCBoost[15]、ADACC[17]、OAUE[18]进行比较. 将这些算法与 NaiveBayes 分类器结合,在上述 4 个数据集上比较它们的分类正确率. 表 1 展示了实验结果数据,其中加粗的数字表示 5 种算法在该数据集上最高的分类正确率.

表 1　5 种算法在 4 个数据集上的分类正确率

Tab. 1　Accuracy of the four algorithms on the five datasets　%

数据集	EB	OCBoost	ADACC	OAUE	RDP
随机树数据集	50.52	23.42	34.42	49.20	**52.49**
邮件列表数据集	55.24	60.19	71.72	55.98	65.09
超平面数据集	88.44	87.79	81.59	88.40	**90.39**
森林覆盖数据集	77.23	46.58	**90.82**	81.53	87.10

从表 1 中可以看出,RDP 和 ADACC 分别在两个数据集上取得了最高的分类正确率. 在随机树数据集上,RDP 的分类正确率最高,而 OCBoost 的分类正确率最低;在邮件列表数据集上,ADACC 的分类正确率最高,EB 的分类正确率最低;在超平面数据集上,RDP 的分类正确率高于其他 4 个算法,而 ADACC 在此数据集上的表现最差;ADACC 在最后一个森林覆盖数据上的表现最好,而 OCBoost 在该数据集上的表现最差. RDP 除了在其中两个数据集上拥有最高的分类正确率之外,在另外两个数据集的表现也很不错,而 ADACC 在其中一个数据集上取得了最低的分类正确率,这说明 RDP 的整体性能优于 ADACC. 由此可以得出一个结论,RDP 能较好地适应概念漂移数据流,可以及时并且正确地检测到数据流中的概念漂移,有效地提高分类器的分类正确率.

5　结论

笔者提出了一种概念漂移检测算法,并在此基础上提出一种检测概念重现的方法,利用图存储历史概念,并预测可能重现的概念. 实验结果表明,笔者提出的方法能够有效地检测突变漂移和渐变漂移,并能够发现概念重现,有效地提高了分类正确率. 下一步希望找到更好的方法减小检测时的时间消耗,并且希望能够对分类正确率进行进一步提高.

参考文献:

[1] HENZINGER M R, RAGHAVAN P, RAJAGOPALAN S. Computing on data streams[R]. s. l.: Digital systems research center, 1998: 107 – 118.

[2] SCHIMMER J, GRANGER R. Incremental learning from noisy data[J]. Machine learning, 1986, 1(3): 317 – 354.

[3] GAMA J, ZLIOBAITE I, BIFET A, et al. A survey on concept drift adaptation[J]. ACM computing surveys (CSUR), 2014, 46(4): 44 – 80.

[4] WEBB G I, HYDE R, CAO H, et al. Characterizing concept drift[J]. Data mining and knowledge discovery, 2015, 30(4): 1 – 31.

[5] WIDMER G, KUBAT M. Learning in the presence of concept drift and hidden contexts[J]. Machine learning, 1996, 23(1): 69 – 101.

[6] WANG S, MINKU L L, GHEZZI D, et al. Concept drift detection for online class imbalance learning[C]// Neural networks (IJCNN), The 2013 international joint conference on. New York: IEEE, 2013: 1 – 10.

[7] GAMA J, MEDAS P, CASTILLO G, et al. Learning with drift detection[M]//Advances in artificial intelligence-SBIA 2004. Berlin: Springer, 2004: 286 – 295.

[8] BAENA-GARCIA M, CAMPO-AVILA J D, FIDALGO R, et al. Early drift detection method[C]//Theproceedings of the 4th ECML PKDD international workshop on knowledge discovery from data streams. CA: AAAI, 2006: 77 – 86.

[9] BIFET A, GAVALDA R. Learning from time-changing data with adaptive windowing[C]//The proceedings of SIAM international conference on data mining. Minneapolis: Amer Stat Assoc, 2007: 443 – 448.

[10] ROSS G J, ADAMS N M, TASOULIS D. K, et al. Exponentially weighted moving average charts for detecting concept drift[J]. Pattern recognition letters, 2012, 33(3): 191 – 198.

[11] SAKTHITHASAN S, PEARS R, KOH Y. One pass concept change detection for data streams[C]//Advances in knowledge discovery and data mining. Berlin: Springer, 2013: 461 – 472.

[12] PEARS R, SAKTHITHASAN S, KOH Y, et al. Detecting concept change in dynamic data streams[J]. Machine learning, 2014, 97(3): 259 – 293.

[13] STREET W N, KIM Y S. A streaming ensemble algorithm (SEA) for large-scale classification[C]//Theproceedings of the 7th ACM SIGKDD international conference on knowledge discovery and data mining. New York: ACM, 1993: 377 - 382.

[14] RAMAMURTHY S, BHATNAGAR R. Tracking recurrent concept drift in streaming data using ensemble classifiers[C]//The 6th international conference on machine learning and applications, ICMEA 2007. CA: IEEE, 2007: 404 - 409.

[15] PELOSSPF R, JONES M, VOVSHA I, et al. Online coordinate boosting[C]//The 12th international conference on computer vision workshops (ICCV Workshops). New York: IEEE, 2009: 1354 - 1361.

[16] MINKU L L, YAO X. DDD: A new ensemble approach for dealing with concept drift[J]. IEEE transactions on knowledge and data engineering, 2012, 24(4): 619 - 633.

[17] JABER G, CORNUEJOLS A, TARROUX P. A new online learning method for coping with recurring concepts: the ADACC system[C]//International conference on neural information processing. Berlin Heidelberg: Springer, 2013: 595 - 604.

[18] BRZEZINSKI D, STEFANOWSKI J. Combining block-based and online methods in learning ensembles from concept drifting data streams[J]. Information sciences, 2014, 265: 50 - 67.

[19] ZHUKOV A, SIDOROV D, FOLEY A. Random forest based approach for concept drift handling[J]. arXiv preprint, 2016: 1602.04435.

[20] KATAKIS I, TSOUMAKAS G, VLAHAVAS I. Tracking recurring contexts using ensemble classifiers: an application to email filtering[J]. Knowledge and information systems, 2010, 22(3): 371 - 391.

[21] GOMES J B, SOUSA P A C, MENASALVAS E, et al. Learning recurring concepts from data streams with a context-aware ensemble[J]. Intelligent data analysis, 2012, 16(5): 803 - 825.

[22] ABAD M A, GOMES J B, MENASALVAS E. Recurring concept detection for spam filtering[C]//The 17th international conference on information fusion (FUSION). New York: IEEE, 2014: 1 - 7.

[23] HOEFFDING W. Probability inequalities for sums of bounded random variables[J]. Journal of the american-statistical association, 1963, 58(301): 13 - 30.

[24] CHERNOFF H. A measure of asymptotic eciency of tests of a hypothesis based on the sum of observations[J]. Annals of mathematical statistics, 1952, 23(4): 493 - 507.

[25] PEEL T, ANTHOINE S, RALAIVOLA L. Empirical bernstein inequalities for U-statistics[C]//Advances in neural information processing systems. New York: Curran Associates, 2010: 1903 - 1911.

[26] BIFET A, HOLMES G, KIRKBY R., et al. MOA: massive online analysis[J]. Journal of machine learning research, 2010, 11: 1601 - 1604.

[27] HULTEN G, SPENCER L, DOMINGOS P. Mining time-changing data streams[C]//Theproceedings of the 7th ACM SIGMOD international conference on knowledge discovery and data mining. New York: ACM, 2001: 97 - 106.

Recurring Concept Detection and Prediction Based on the Graph

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

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

Abstract: Concept drift was a challenging problem in stream mining. When the concept drift occured, the accuracy of the original predictive model may decrease significantly. So it was necessary to put forward a feasible method to detect concept drift. Recurring concept is a special case of concept drift. However, most of existing algorithms have not taken full account of this case. This research proposed an approach to the recurring concept detection problem. Extensive experiment revealed that the method we proposed could detect not only the concept drift with relatively low delay and rate of false positive, but also the recurring concepts. Moreover, the accuracy of the classification would be greatly improved.

Key words: data stream; data mining; concept drift; drift detection; recurring concept