基于 Kappa 系数的数据流分类算法

徐树良[1] 王俊红[1,2]
(山西大学计算机与信息技术学院 太原 030006)[1]
(计算智能与中文信息处理教育部重点实验室 太原 030006)[2]

摘 要 数据流挖掘已经成为数据挖掘领域一个热门的研究方向,由于数据流中概念漂移现象的存在,使得传统的分类算法无法直接应用于数据流中。为了能有效地应对数据流中的概念漂移,提出了一种基于 Kappa 系数的数据流分类算法。该算法采用集成式分类技术,以 Kappa 系数度量系统的分类性能,根据 Kappa 系数来动态地调整分类器,当发生概念漂移时,系统能利用已有的知识很快删除不符合要求的分类器来适应新概念。实验结果表明,相对于实验中参与比较的 BWE,AE 和 AWE 算法,该算法不但具有较好的分类性能,而且在一定程度上能较为有效地降低时间开销。

关键词 数据流,概念漂移,分类,Kappa 系数
中图法分类号 TP181 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2016.12.031

Data Stream Classification Algorithm Based on Kappa Coefficient

XU Shu-liang[1] WANG Jun-hong[1,2]
(School of Computer & Information Technology, Shanxi University, Taiyuan 030006, China)[1]
(Key Laboratory of Computational Intelligence and Chinese Information Processing of Ministry of Education, Taiyuan 030006, China)[2]

Abstract Data streams mining has become one of hot topics in the area of data mining. Because of the existence of concept drift, it is impossible for conventional classification algorithms to be directly applied in data streams environment. In order to deal with the concept changes in data streams, an algorithm based on Kappa coefficient was proposed. The approach uses ensemble classification techniques and a weighted voting strategy to decide the labels of test sets, in addition, the approach employs Kappa coefficient to measure the performance of classification system. When the performance of classifiers decreases significantly, an alarm about concept drift will be made and the algorithm will apply prior knowledge to delete inaccurate classifiers to adapt to new concept. The experimental results shows that, comparing with the contrast algorithms in the experiments: BWE, AE and AWE, the new approach can not only possess better performance for classification, but also efficiently decrease time cost.

Keywords Data streams, Concept drift, Classification, Kappa coefficient

1 引言

随着信息社会的发展,如电话通信、电子商务、股票交易、舆情监测、网络流量监控等许多领域产生了大量的数据流,这些数据与传统的静态数据有着较大区别,往往具有数量无限、连续到达和概念漂移等特性,使得传统的数据挖掘方法面临着巨大的挑战[1]。

近年来,数据流问题获得了学者们的重视,特别是包含概念漂移的分类问题得到了广泛的研究[2]。Shaker 等人提出了一种基于实例的数据流分类算法 IBLStream[3],此方法与 KNN 算法类似,通过保存相关事例,新到来的数据选择距离最近的若干个事例,依据它们的类标签作出决策,在分类的过程中计算分类结果的错误率,选出最小的错误率,并结合错误率的方差来进行概念漂移检测。针对概念反复出现的概念漂移问题,Gama 等人[4]提出了一种元学习理论的概念漂移检测算法,该算法检测概念变化的方法与 IBLStream 类似,当一个分类器的预测准确率超过规定的阈值时,则激活该分类器,将该分类器加入到对未知数据的分类系统中。Bifet 等人提出了一种自适应窗口的 Hoeffding 算法 HWF-ADWIN[5],该方法对于到来的数据利用 Hoeffding 不等式,结合最大和第二大信息增益对应的属性来对叶节点进行划分,同时当检测到分类器的准确率显著发生改变时,创建一棵备选子树,若备选子树的准确率高于当前决策子树,则用备选子树去替换当前子树,以此来应对数据中的概念漂移。针对具有大量未标签数据和少量有标签数据的情况,Wu 等人提出了一种随机决策树模型的 CDRT 算法[6],此方法使用半监督学习技术,采用聚类的思想来对未标记的数据进行标记,在分类过程中构建决策树,用信息增益来确定分裂属性,使用子树替换的策

到稿日期:2015-10-21 返修日期:2016-03-25 本文受国家自然科学基金(61202018)资助。
徐树良(1989-),男,硕士生,主要研究方向为机器学习,E-mail:xushulianghao@126.com;王俊红(1979-),女,博士,副教授,硕士生导师,主要研究方向为数据挖掘与机器学习,E-mail:wjhwjh@sxu.edu.cn(通信作者)。

略来应对概念漂移，此算法能够有效地区分概念漂移和噪声。针对部分标记的数据流分类问题，Gama 等人提出了一种基于双衰减因子的概念漂移检测算法[7]，该算法设置两个衰减因子，根据分类结果进行序列错误评估，获得相关参数后利用 Page-Hinkley 测试进行概念漂移检测。Brzezinski 提出了一种应对不平衡分类问题的 AUC 算法[8]，该算法采用 ROC 曲线的下方面积来衡量算法的性能，同时为了降低算法的时空消耗，采用了滑动窗口机制；为了能有效地查找和移除过时的节点，AUC 算法采用了红黑树结构来保存数据，当系统的数据量达到上限时，淘汰最早的数据，同时加入新数据。Rutkowski 等人提出了一种基于 McDiarmid 不等式的 McDiarmid Tree 算法[9]，此方法与 VFDT 算法[10]类似，通过计算最大信息增益和第二大信息增益来确定分裂属性，两者之差的阈值由 McDiarmid 界来确定，该算法的分类属性既可以用信息增益来度量，也可使用基尼系数来度量，由于决策树的叶节点保存了各类数据点的统计信息，使得算法处理数据的速度在一定程度上得到了加快。Dectert 提出了一种基于批处理的数据流分类算法 BWE[11]，该算法依据分类结果赋予分类器相应的权值，利用加权投票机制来做出决策，通过记录历史数据分类结果的准确率和方差来检测概念漂移。Zhang 等人提出了一种聚合集成式分类算法 AE[12]，该算法在数据流的水平方向和垂直方向上同时训练分类器，在水平方向上利用不同的数据块训练相同的分类算法，在垂直方向上利用同一数据块来训练不同的分类算法，最终采用多数投票的方式对分类结果做出决策。Wang 等人提出了一种基于 Top-k 机制的 AWE 算法[13]，该算法在滑动窗口进入新数据块时，利用新数据块来训练一个新分类器并计算出其权值，然后从已有的分类器中选出 k 个权值最高的分类器来对未知数据进行分类，分类的结果也采用加权投票机制来决定。

为了应对数据流中的概念漂移问题，本文提出了一种基于 Kappa 系数的数据流分类算法（Data Stream Classification Algorithm Based on Kappa Coefficient，DSCK），该算法在分类的过程中计算各个数据块的分类结果的 Kappa 系数，利用 Kappa 系数来检测数据流中的概念是否发生改变。当数据流中的概念发生了变化，系统会依据已有的知识，及时将所有不符合要求的分类器淘汰。相对于实验中的对比算法，本文算法不但能获得较高的准确率，而且在时间开销上也有一定程度的降低，取得了较好的结果。

本文第 2 节描述了相关概念，对数据块和概念漂移进行了解释；第 3 节介绍了基于 Kappa 系数的数据流分类算法，详细地介绍了 DSCK 算法的执行步骤；第 4 节进行实验分析，用人工数据集和真实数据集来验证算法的性能；最后为结束语部分，对全文进行总结，并展望未来的工作。

2 相关概念

设$\{\cdots,d_{t-1},d_t,d_{t+1},\cdots\}$为系统产生的数据流，$d_t$ 为 t 时刻产生的数据，对于每一个数据，$d_t=\langle \boldsymbol{x},y_t\rangle$，$i=1,2,\cdots$，其中：$\boldsymbol{x}=\langle x_1,x_2,x_3,\cdots,x_n\rangle$为 d_t 在各个属性上的取值，y_t 为该数据对应的类标签，并且有 $y_t\in \boldsymbol{y}$。

在数据流环境下，往往面对的是海量数据，存储数据所需的空间远远超出了内存的容量，为了能有效地处理海量数据，一般采用滑动窗口机制，即将数据流分成一个个数据块，每时刻只允许一个或若干个数据块进入内存，只有处理完当前窗口内的数据块才允许下一个窗口的数据块进入内存。

定义 1 设在 Δt 时间内，若分类器系统对滑动窗口内的数据分类结果的错误率始终处于较低水平，则称在此段时间内该数据的概念是稳定的。即：$P(err\text{-}best<\varepsilon)\geqslant 1-\alpha$，其中：$err$ 为当前分类器对数据分类的错误率，$best$ 为理想最优性能的分类器对该数据的分类错误率，α 为显著性水平。

定义 2 在 t 时刻，对滑动窗口内的数据进行训练获得的概念为 M，经过 Δt 时间后，再次对数据进行训练获得概念为 N，如果 $M\neq N$，则称数据流在 Δt 时间内发生了概念漂移。若数据流中的概念的变化是在较短时间内完成的，则称此种类型的概念漂移为突变式概念漂移；若数据流中的概念的变化是在较长时间内完成的，则称此种类型的概念漂移为渐进式概念漂移[14]。

当数据流中的概念发生改变后，系统分类器的错误率会以较大幅度上升，此时需要采取一定的策略来更新分类器系统，以适应数据流中的新概念。

3 基于 Kappa 系数的数据流分类算法

3.1 DSCK 算法的基本原理

在诊断实验中，研究者通常需要考察不同诊断方法在诊断结果上是否具有一致性，为了能有效地量度出这种一致性，Cohen 提出了 Kappa 系数。Kappa 系数是一种度量测量结果一致性的统计量，其 κ 值的计算方法如下[15]：

$$\kappa=\frac{p_0-p_e}{1-p_e} \tag{1}$$

其中，p_0 表示观测结果的一致率，p_e 为偶然达到的一致性比率，即两次检验结果由于偶然机会所达到的一致性。

在数据流环境中，如果数据的概念保持相对稳定，则随着不断训练，系统分类器的性能会逐渐提高并最终趋于稳定，此时分类器系统对各个数据块分类结果的准确率基本相同，若分类器对数据块的分类性能发生了显著性下降，则可以认为当前数据块的概念发生了改变。为了能有效地利用 Kappa 系数来度量分类器分类性能的稳定性，本文对 Kappa 系数相关参数进行了重新定义，如式(2)所示：

$$\kappa=\frac{\bar{p}-p_i}{1-p_i} \tag{2}$$

其中，p_i 为分类器系统对当前数据块 B_i 分类结果的准确率；$\bar{p}$为分类器系统中各个子分类器对距上一次概念漂移之间的每个数据块分类结果准确率的平均值，其按式(3)计算：

$$\bar{p}=\frac{1}{n}\sum_{j=1}^{n}\max(p_j) \tag{3}$$

其中，$p_j=\{p_{j1},p_{j2},\cdots,p_{jk}\}$为分类器系统中 k 个子分类器对数据块 B_j 分类准确率的集合，n 为当前数据块 B_j 与上一次概念漂移之间数据块的数目。

由式(2)可以看出，Kappa 系数反映了当前分类器与之前概念稳定时分类器性能之间的差异性，其值越小，表明当前分类器的性能越好，因此数据流中的概念处于稳定的可能性越大；反之，则数据流中概念处于变化的可能性也就越大。在 Kappa 系数的计算中，使用$\bar{p}$来评价之前概念稳定时分类器的性能，从而避免了仅仅比较相邻两个数据块中的分类器性能而产生的一系列偶然性误差。

在实际应用中，κ 的阈值选取是一个很大的问题，如果阈值选取过小会使算法对数据的概念的变化过于敏感，如果阈值选取过大会使算法对概念的变化反应过于迟钝，这两种情况都会降低算法的性能。为了能够选取合适的阈值，本文引入如下定理。

定理 1[16] 设 $X_i(i=1,2,3,\cdots,n)$ 为服从二项分布的随机变量，且有 $X_i=0$ 或 1，其中 $S=\sum_{i=1}^{n}X_i$，$E(S)=\mu$，对于任意 $0<\varepsilon<1$，都有如下不等式成立：

$$P(S>(1+\varepsilon)\mu)=P(|S-\varepsilon|>\varepsilon\mu)\leqslant e^{-\frac{\varepsilon^2\mu}{3}} \tag{4}$$

由定理 1 可知，由服从 0-1 分布的随机变量和组成的随机变量的观测值与期望值十分接近，观测值大于期望值的概率呈指数形式衰减。

定理 2 设 $\kappa_i=\frac{\bar{p}-p_i}{1-p_i}$ 为根据式(2)和数据块 B_i 计算出的 Kappa 系数，在数据流中概念稳定的条件下，有：$\kappa_i\leqslant\frac{1}{1-p_i}\sqrt{\frac{-3\bar{p}\ln\alpha}{n}}$，其中，$\alpha$ 为显著性水平，n 为试验观测的次数。

证明：由定理 2 可得：

$$P(|S-\mu|\leqslant\varepsilon\mu)=P(|\frac{S}{n}-\frac{\mu}{n}|\leqslant\frac{\mu\varepsilon}{n})\geqslant1-e^{-\frac{\varepsilon^2\mu}{3}} \tag{5}$$

由式(5)可知，当数据流中的概念稳定时，分类结果准确率的观测值与其期望值相差很小，有：

$$1-\alpha=1-e^{-\frac{\varepsilon^2\mu}{3}} \tag{6}$$

所以解式(6)得：

$$\varepsilon=\sqrt{\frac{-3\ln\alpha}{\mu}}=\sqrt{\frac{-3\ln\alpha}{n\bar{p}}} \tag{7}$$

在 $1-\alpha$ 的置信水平下，有：$\bar{p}-p_i\leqslant\mu\frac{\varepsilon}{n}=\bar{p}\varepsilon$，即：

$$\kappa_i=\frac{\bar{p}-p_i}{1-p_i}\leqslant\frac{1}{1-p_i}\sqrt{\frac{-3\bar{p}\ln\alpha}{n}} \tag{8}$$

通过计算每个数据块的 κ 值，可以利用定理 2 来检测数据流的概念是否发生改变，如果改变，则认为系统分类器的性能出现了显著变化，发生了概念漂移。

发生概念漂移后，算法需要依据一定的策略来对分类器系统进行更新，在更新的过程中，需要迅速淘汰掉分类性能较差的子分类器，来确保算法能够以尽可能快的速度收敛到新概念。此时，对于每一个子分类器，将式(2)中的 p_i 更换成当前子分类器 j 对数据块 B_i 分类结果的准确率 p_{ij}，计算出 κ_{ij}，如果 κ_{ij} 不满足式(8)，则认为分类器 j 不再符合当前数据流中的概念，需删除。

3.2 DSCK 算法的执行过程

由以上知识可知，DSCK 算法的执行过程如下所示。

算法 1 DSCK

输入：数据流 S，滑动窗口大小 winsize，分类器系统 ensemble=NULL，保存系统分类器对数据块分类结果的数组 acc={0}，k 为系统分类器数目的上限

输出：分类器系统 ensemble

```
While S!=NULL
{
  读取 winsize 条数据形成数据块 Bi;
  If size(ensemble)==0
  {
    用 Bi 训练一个新分类器 Cj，Cj. weight=1，ensemble=ensemble∪Cj;
  }
  Else
  {
    用 ensemble 对 Bi 进行分类，采用加权投票的方式作出决策获得 pi，acc=acc∪pi，计算出 p̄，κi;
    For each ensemble(m)∈ensemble
    {
      根据 ensemble(m)对 Bi 的分类结果计算准确率 pmi 和 κmi;
      ensemble(m). weight=log((1+κmi)/(κmi+σ))(其中 σ 为很小的数常数);
    }
    If κi>1/(1-pi)*sqrt(-3 p̄ln α / n) //发生概念漂移
    {
      删除系统中所有不满足式(8)的子分类器;
      用 Bi 训练一个新分类器 Cj，Cj. weight=p̄;
      If size(ensemble)≥k
      {
        删除一个权值最小的分类器;
      }
    ensemble=ensemble∪Cj;
    acc=NULL;
    }//end if 概念漂移
    用 Bi 去训练 ensemble 中每一个分类器;
  }//end else
}//end while
```

在 DSCK 算法中，以 Kappa 系数作为度量分类器性能是否稳定的依据，当数据流中的概念稳定时，各个分类器的分类性能基本一致，因此此时计算出的 κ 值应该非常小，符合式(8)的约束。为了能够及时删除无效的分类器，DSCK 算法利用各个单分类器的 Kappa 系数来判断子分类器的性能是否与之前的最优分类器相一致，与最优分类器相比，性能出现明显下降的子分类器会被系统立即删除，以保证算法能够以较快的速度收敛到新概念。

3.3 算法分析

在 DSCK 算法中使用了 Kappa 系数来度量分类器的性能，根据文献[17]可知，Kappa 统计量标准误差的近似值为：

$$\delta_i=\sqrt{\frac{p_i(1-p_i)}{N(1-\bar{p})^2}} \tag{9}$$

其中，N 为测试样本的大小。

在实际应用中，根据噪声学习理论[18,19]可知，当样本数 N 满足：

$$N\geqslant\frac{2}{\varepsilon^2(1-2\eta)^2}\ln(\frac{2n}{1-\alpha}) \tag{10}$$

则目标假设 H_i 与目标函数 H^* 不相符的概率为：

$$P(d(H_i,H^*)\geqslant\varepsilon)\leqslant1-\alpha \tag{11}$$

其中，ε 为分类错误率的上限，η 为噪声率的上限，n 为可能假设的总数，α 为显著性水平。设 β 为式(10)取等号时的参数，即：

$$N=\frac{2\beta}{\varepsilon^2(1-2\eta)^2}\ln(\frac{2n}{1-\alpha}) \tag{12}$$

解式(12)可得：

$$\varepsilon=\sqrt{\frac{2\beta}{(1-2\eta)^2N}\ln\left(\frac{2n}{1-\alpha}\right)} \tag{13}$$

结合式(9)和式(13),则 DSCK 算法的分类结果错误率的上限为:

$$error_{\max}=1-\left(1-\sqrt{\frac{p_i(1-p_i)}{N(1-\bar{p})^2}}\right)\times\left(1-\sqrt{\frac{2\beta}{(1-2\eta)^2N}\ln\left(\frac{2n}{1-\alpha}\right)}\right) \tag{14}$$

4 实验分析

为了验证 DSCK 算法的性能,选用 BWE[11],AE[12] 及 AWE[13] 作为对比算法;数据集选用 MOA 环境[20] 产生 *Hyperplane*, *LEDGeneratorDrift*(记为:*LED*), *WaveformGeneratorDrift*(记为:*Waveform*)和 *RandomRBFGeneratorDrif*(记为:*RBF*)4 个包含概念漂移的人工数据集,以及 *Shuttle* 和 *Page Blocks*(记为:*Page*)两个选自 UCI 的真实数据集。实验环境为:Windows7 操作系统,Intel Core 2.94G 双核 CPU,4GB 内存,算法程序由 Matlab R2013a 实现。在实验中,若无特殊说明,子分类器数目上限 $k=4$,显著性水平 $\alpha=0.05$,$\sigma=10^{-7}$,AE 算法的基分类器采用 C4.5 算法和 CART 算法实现,AWE 算法的基本分类器采用 C4.5 算法实现,DSCK 算法的基本分类器采用 CART 算法实现。

4.1 数据集描述

Hyperplane 数据集:其是一个人工数据,含有 22 维,50000 条数据。一个 d 维超平面样本 X 满足如下的数学表达式:$\sum_{i=1}^{d}a_ix_i=a_0$。当数据满足 $\sum_{i=1}^{d}a_ix_i\geqslant a_0$ 时,其类标记为 1;否则其类标记为 2,其中:$a_0=\frac{1}{2}\sum_{i=1}^{d}a_i$。

LED 数据集:其包含 25 个属性,其中 17 个属性为无关属性,在该数据集中前 7 个属性为漂移属性维,共包含 50000 个事例及 10 个不同的类标签。

Waveform 数据集:其包含 22 个属性,前 21 个属性在实数范围内取值,共含有 3 个不同的类标签及 50000 条事例,该数据集含有 5%的噪声。

RBF 数据集:其该数据集是一个由随机产生的数据点组成的数据集,数据中包含 11 个属性,前 10 个属性在实数范围内取值,共含有 50000 个事例及 5 个不同的类标签。

Shuttle 数据集:包含 10 个属性,前 9 个属性均在实数范围内取值,在所有事例中包含 7 个不同类标签,共含有 43500 个事例。

Page 数据集:该数据集中的事例来自 54 个不同的文档。每一个观察都涉及一个区块。数据含有 11 个维,所有属性都是数字,前 3 个维的取值均为整数,后 7 个维在实数范围内取值,最后一个维的取值代表数据的类标签。该数据集共含有 5473 个事例及 5 个不同的类标签。

4.2 实验过程和结果分析

为了研究实验算法的分类性能,选取 *Hyperplane*,*LED*, *Waveform*,*RBF*,*Shuttle* 及 *Page* 等 6 个数据集作为实验数据,分别运行 DSCK,BWE,AE 及 AWE 算法,除了在 *page* 数据集上 *winsize*=600,其余数据集的 *winsize*=2000,在 DSCK 算法中的参数 n 取 2×10^4,最终获得的实验结果如表 1、表 2 所列。

表 1 实验算法在不同数据集上的平均准确率

	BWE	AE	AWE	DSCK
Hyperplane	0.6368	**0.7552**	0.6935	0.7315
LED	0.1932	0.7297	0.3110	**0.8501**
Waveform	0.6525	0.7506	0.6609	**0.7894**
RBF	0.2939	0.6897	0.4067	**0.8293**
Shuttle	0.8691	0.9671	0.8691	**0.9981**
Page	0.92083	0.91583	0.9200	**0.9271**

表 2 算法测试数据集所需的时间(单位:s)

	BWE	AE	AWE	DSCK
Hyperplane	568.904	142.045	928.777	**26.023**
LED	5.883	**2.289**	5.008	15.820
Waveform	1876.724	307.794	2102.179	**98.940**
RBF	361.781	168.706	1111.152	**58.681**
Shuttle	602.333	128.618	999.731	**3.559**
Page	26.041	10.594	34.474	**0.525**

由表 1 和表 2 的数据分析可得,在绝大多数的数据集上,DSCK 算法不但准确率最高,而且所消耗的时间也最少,算法的性能最佳。在 *Hyperplane* 数据集上,DSCK 算法的准确率略低于 AE 算法,但从实验数据来看,AE 算法的准确率只比 DSCK 算法高 0.0237,其消耗的时间却是 DSCK 算法的 5.46 倍,因此从综合性能来看,DSCK 算法的性能要优于 AE 算法。

从表 1 来看,在 LED 和 RBF 数据集上,BWE 和 AWE 算法的准确率与 DSCK 和 AE 算法的差距十分明显。出现这种现象主要与 BWE 和 AWE 算法处理概念漂移的策略有关。在 DSCK 算法中,当发生概念漂移时,算法能够将所有分类性能较差的子分类器一次性删除;在 AE 算法中,算法每遇到一个数据块就会删除之前的分类器,在水平和垂直方向上重新训练子分类器;以上两种算法都能够保证不符合要求的子分类器以很快速度被删除,不会参与到对新数据的决策中。而在 AWE 和 BWE 算法中,算法处理概念漂移时采取逐步淘汰的策略,即每一次训练只能淘汰一个不符合要求的子分类器,当系统中有多个分类器不符合当前概念时,淘汰掉这些分类器需要较长的训练过程,在完全淘汰之前,这些性能较差的分类器仍然会继续参与对新数据分类的决策,当概念的变化速度快于分类器的淘汰速度时,BWE 和 AWE 算法的分类准确率就会始终处于较低水平。

从表 1 中的 *Shuttle* 和 *Page* 数据集来看,尽管两个数据集也存在概念漂移,各个算法的准确率却都比较高,与 *Hyperplane* 和 *RBF* 等数据集相比,存在明显不同。在 *Shuttle* 数据上,各个类的数据分布较为均匀,即同一数据块内各类数据的数量大致相同;在 *Page* 数据集中,同一数据块内同一类的数据分布较为集中;以上两种情况都使得训练集和测试集的数据分布较为一致,所以训练出来的分类模型对测试数据具有较好的预测能力。

从表 1 的 *Hyperplane* 和 *Waveform* 数据集来看,各个算法的准确率都不太高,每一种算法的准确率都未能达到 0.8,这主要与数据集中数据的特性相关。在 *Hyperplane* 和 *waveform* 数据集中,数据产生了渐进式概念漂移,即每个数据块之间,数据的概念存在一定的差异,因此通过训练集训练出来的分类模型并不完全符合测试集的数据模型,所以各个算法的准确率都不太高。在这两个数据集上尽管数据块之间的概念存在一定的差异,但这种差异的变化是相对较小的,还远没有达到突变式概念漂移中概念变化的剧烈程度,因此

BWE 和 AWE 算法的准确率没有像在 *LED* 和 *RBF* 中那样处于较低水平。

为了进一步研究滑动窗口的大小对算法分类性能的影响，选用 *Hyperplane*，*Waveform*，*RBF* 和 *LED* 4 个数据集作为实验数据，*winsize* 分别取不同的值运行 DSCK 算法，最终获得的实验结果如表 3 和图 1 所示。

表 3　DSCK 在不同 *winsize* 取值下的准确率

winsize	Hyperplane	Waveform	RBF	LED
300	0.7450	0.7676	0.8267	0.8440
500	0.7540	0.7714	0.8270	0.8292
700	**0.7571**	**0.7901**	0.8267	**0.8554**
1000	0.7433	0.7867	**0.8369**	0.8514
1500	0.7346	0.7898	0.8321	0.8544
2000	0.7315	0.7894	0.8291	0.8501

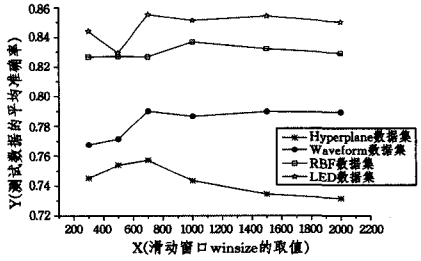

图 1　不同 *winsize* 取值下的 DSCK 准确率

由表 3 的数据和图 1 中曲线的基本变化趋势分析可得，随着 *winsize* 的增加，在绝大部分数据集下 DSCK 算法的准确率会先增加后下降。对于 DSCK 算法，当 *winsize* 取值较小时，随着其值的增大，分类器获得训练的数据增多，训练出来的分类模型更加符合测试数据的实际情况，因而准确率会逐渐提高。当 *winsize* 达到一定值后，再增大滑动窗口会使得窗口内包含多个概念的数据，同时窗口过大也降低了算法对数据流中概念变化的敏感性，训练出来的模型与测试数据的差异逐渐增大，因而准确率会逐步下降。

为了研究 k 值的取值对 DSCK 算法的影响，选用 *Hyperplane*，*LED*，*Waveform* 和 *RBF* 4 个数据集作为实验数据，在 k 分别取不同的值且 *winsize*＝2000 时，运行 DSCK 算法的最终结果如表 4 和图 2 所示。在不同 k 值下 DSCK 的时间开销如表 5 所列。

表 4　不同 k 值下 DSCK 的平均准确率

k 值	2	3	4	5	6
Hyperplane	0.6730	0.6956	0.7315	0.7423	0.7427
LED	0.8244	0.8218	0.8501	0.8516	0.8515
Waveform	0.7558	0.7693	0.7894	0.7915	0.7964
RBF	0.8086	0.8206	0.8293	0.8346	0.8359
Shuttle	0.9971	0.9974	0.9981	0.9980	0.9980
Page	0.7815	0.8456	0.8531	0.8527	0.8542

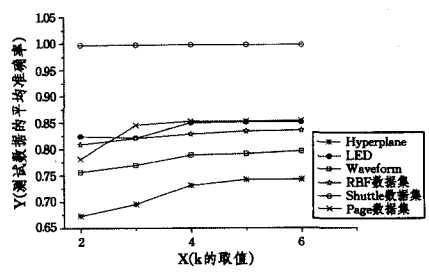

图 2　不同 k 值下 DSCK 的平均准确率的变化趋势

表 5　在不同 k 值下 DSCK 的时间开销(s)

k 值	2	3	4	5	6
Hyperplane	27.3583	32.2447	38.4486	49.4619	64.5322
LED	5.0589	8.3370	15.7729	17.9922	20.4956
Waveform	22.018	75.4539	107.7511	133.7969	155.3021
RBF	30.1970	44.8041	63.3659	86.0360	98.7768
Shuttle	1.6205	2.8674	3.8967	4.6768	5.5693
Page	1.2177	1.5033	1.7201	1.8310	2.1742

由表 4 的数据分析可得，随着 k 值的增加，DSCK 算法的准确率也提高。出现这种变化趋势主要是因为随着 k 值的增加，参与决策的分类器的数目也增多，因此经过集成分类器系统获得的决策结果能够减小由单个分类器自身和偶然性误差对分类结果的影响，所以子分类器数目越多的分类器系统能够获得更高的准确率。

从图 2 中数据的变化趋势分析可得，当准确率增加到一定程度后，增加 k 值使得 DSCK 算法准确率提高的幅度十分微小，基本趋于平稳。出现这种变化趋势是因为当子分类器达到一定数目后，分类器系统的可靠性已经达到了较高程度，其性能已经趋于稳定，再增加 k 值对分类器性能的影响十分有限。从表 5 可知，当 k 取值较大时，系统需要花费较多的时间来训练分类器和处理各个子分类器的决策结果，算法的时间消耗会大幅提高，此时算法的综合性能反而会下降，所以在实际应用中需要综合考虑各方面因素来选取 k 值。

结束语　本文提出了一种基于 Kappa 系数的数据流分类算法 DSCK，在该算法中使用了 Kappa 系数来度量当前分类器与之前的分类器性能的一致性，从而检测是否发生概念漂移，发生概念漂移后，DSCK 算法所采用的措施能够保证算法以很快的速度收敛到新概念。实验结果表明，该算法在时间开销和准确率上取得了较好的效果。然而从表 1 和图 2 的实验数据来看，DSCK 算法对概念变化迅速的数据集较为有效，对于概念变化缓慢的 *Hyperplane* 数据集和 *Waveform* 数据集上的测试效果，与已有的数据流分类算法相比，准确率上并没有特别大的优势，因此如何提高 DSCK 算法以应对概念变化缓慢数据集的能力将是下一步研究的重点。

参 考 文 献

[1] Bifet A, Holmes G, Pfahringer B. Leveraging Bagging for Evolving Data Streams [M]// Machine Learning and Knowledge Discovery in Databases. Springer Berlin Heidelberg, 2010: 135-150

[2] Lemaire V, Salperwyck C, Bondu A. A survey on Supervised Classification on Data Streams [J]. Business Intelligence, 2015, 205: 88-125

[3] Sharker A, Hullermeier E. IBLStreams: a system for instance-based classification and regression on data streams [J]. Evolving System, 2012, 3(4): 235-249

[4] Gama J, Kosina P. Recurrent concepts in data streams classification [J]. Knowledge and Information Systems, 2014, 40(3): 489-507

[5] Bifet A, Gavalda R. Adaptive Learning from Evolving Data Streams [C]// Proceedings of 8th International Symposium on Intelligent Data Analysis. Heidelberg: Springer, 2009: 249-260

[6] Wu Xin-dong, Li Pei-pei, Hu Xue-gang. Learning from concept drifting data streams with unlabeled data [J]. Neurocomputing, 2012, 92(3): 145-155

[7] Gama J, Sebastiao R, Rodrigues P P. On evaluating stream

learning algorithms [J]. Machine Learning, 2013, 90: 317-346

[8] Brzezinski D, Stefanowski J. Prequential AUC for Classifier Evaluation and Drift Detection in Evolving Data Streams [C]// Third International Workshop NFMCP 2014 Held in Conjunction with ECML(PKDD 2014). Heidelberg: Springer, 2015: 87-101

[9] Rutkowski L, Pietruczuk L, Duda P, et al. Decision trees for mining data streams based on the McDiarmid's bound [J]. IEEE Transactions on Knowledge and Data Engineering, 2013, 25(6): 1272-1279

[10] Domingos P, Hulten G. Mining High-Speed Data Streams [C]// Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM, 2000: 71-80

[11] Magdalena. Batch Weighted Ensemble for Mining Data Streams with Concept Drift [C]// 9th International Symposium(ISMIS 2011). Heidelberg: Springer, 2011: 290-299

[12] Zhang Peng, Zhu Xing-quan, Shi Yong, et al. An Aggregate Ensemble for Mining Concept Drifting Data Streams with Noise [C]// 13th Pacific-Asia Conference, PAKDD 2009. Heidelberg: Springer, 2009: 1021-1029

[13] Wang Hai-xun, Fan Wei, 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. New York: ACM, 2003: 226-235

[14] Wozniak M, Kasprzak A, Cal P. Weighted Aging Classifier Ensemble for the Incremental Drifted Data Streams [C]// 10th International Conference, FQAS 2013. Heidelberg: Springer, 2013: 579-588

[15] Zhou Ji-xiang, Mao Shi-song. Statistical Methods for Quality Management [M]. Beijing: China Statics Press, 2008: 433-440(in Chinese)
周纪芗, 茆诗松. 质量管理统计方法[M]. 北京: 中国统计出版社, 2008: 433-440

[16] Wang Tao, Liu Ming-ju, Li De-ming. A Sstrong Chernoff Bounds Derived from Equitable Colorings of Graphs [J]. Journal of Mathematics, 2014, 34(6): 1015-1024

[17] Zliobaite I, Bifet A, Read J, et al. Evaluation methods and decision theory for classification of streaming data with temporal dependence [J]. Machine Learning, 2015, 98(3): 455-482

[18] Zhang Chen-guang, Zhang Yan. Semi-Supervised Learning [M]. Beijing: China Agriculture Scientech Press, 2013: 31-33(in Chinese)
张晨光, 张燕. 半监督学习[M]. 北京: 中国农业科学技术出版社, 2013: 31-33

[19] Li Pei-pei, Wu Xin-dong, Hu Xue-gang, et al. Learning concept-drifting data streams with random ensemble decision trees [J]. Neurocomputing, 2015, 166(c): 68-83

[20] Bofet A, Holmes G, Kirkby R, et al. MOA: Massive Online Analysis [J]. The Journal of Machine Learning Research, 2010, 11(2): 1601-1604

(上接第 167 页)

[4] Liu B, Fu Y, Yao Z, et al. Learning geographical preferences for point-of-interest recommendation[C]// Acm Sigkdd International Conference on Knowledge Discovery & Data Mining. ACM, 2013: 1043-1051

[5] Zhou D, Wang B, Rahimi S M, et al. A Study of Recommending Locations on Location-Based Social Network by Collaborative Filtering[M]// Advances in Artificial Intelligence. Springer Berlin Heidelberg, 2012: 255-266

[6] Hu B, Ester M. Social Topic Modeling for Point-of-Interest Recommendation in Location-Based Social Networks[C]// 2014 IEEE International Conference on Data Mining (ICDM). IEEE Computer Society, 2014: 845-850

[7] Jiang S, Qian X, Shen J, et al. Author Topic Model based Collaborative Filtering for Personalized POI Recommendation[J]. IEEE Transactions on Multimedia, 2015, 17(6): 907-918

[8] Liu B, Xiong H. Point-of-Interest Recommendation in Location Based Social Networks with Topic and Location Awareness[C]// SDM. 2013: 396-404

[9] Yin H, Zhou X, Shao Y, et al. Joint Modeling of User Check-in Behaviors for Point-of-Interest Recommendation[C]// Proceedings of the 24th ACM International on Conference on Information and Knowledge Management. ACM, 2015: 1631-1640

[10] Gao H, Tang J, Hu X, et al. Content-aware point of interest recommendation on location-based social networks[C]// Proceedings of the 29th AAAI Conference on Artificial Intelligence. 2015

[11] LI Gui, CHEN Sheng-hong, HAN Zi-yang, et al. Location-aware Recommendation Based on Collaborative Filtering[J]. Computer Science, 2014, 41(11A): 340-346(in Chinese)
李贵, 陈盛红, 韩子阳, 等. 基于协同过滤的位置感知推荐[J]. 计算机科学, 2014, 41(11A): 340-346

[12] Gao H, Tang J, Hu X, et al. Exploring temporal effects for location recommendation on location-based social networks[C]// Proceedings of the 7th ACM Conference on Recommender Systems. ACM, 2013: 93-100

[13] Wang Zhen-zhen, He Ming, Du Yong-ping. Text Similarity Computing Based on Topic Model LDA [J]. Computer Science, 2013, 40(12): 229-232(in Chinese)
王振振, 何明, 杜永萍. 基于 LDA 主题模型的文本相似度计算[J]. 计算机科学, 2013, 40(12): 229-232

[14] Zhou Er-chong, Huang Jia-jin, Xu Xin-xin. A Point-of-Interest Recommendation Method Based on User Check-in Behaviors in Online Social Networks[J]. Computer Science, 2015, 42(10): 232-234(in Chinese)
周而重, 黄佳进, 徐欣欣. 一种基于用户网络签到行为的位置推荐方法[J]. 计算机科学, 2015, 42(10): 232-234

[15] Zheng Jiong, Shi Gang. Recommender Algorithm Based on Dynamical Trust Relationship between Users[J]. Computer Science, 2015, 42(9): 230-234(in Chinese)
郑炅, 石刚. 基于用户间动态信任关系的推荐算法研究[J]. 计算机科学, 2015, 42(9): 230-234