计算机研究与发展
Journal of Computer Research and Development
DOI:10.7544/issn1000-1239.2016.20160223
53(8):1781-1791,2016

基于在线迁移学习的重现概念漂移数据流分类

文益民[1,2,3] 唐诗淇[1] 冯 超[1] 高 凯[4]
[1](桂林电子科技大学计算机与信息安全学院 广西桂林 541004)
[2](广西可信软件重点实验室(桂林电子科技大学) 广西桂林 541004)
[3](广西信息科学实验中心(桂林电子科技大学) 广西桂林 541004)
[4](河北科技大学信息学院 石家庄 050018)
(ymwen2004@aliyun.com)

Online Transfer Learning for Mining Recurring Concept in Data Stream Classification

Wen Yimin[1,2,3], Tang Shiqi[1], Feng Chao[1], and Gao Kai[4]
[1](*School of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin, Guangxi* 541004)
[2](*Guangxi Key Laboratory of Trusted Software (Guilin University of Electronic Technology), Guilin, Guangxi* 541004)
[3](*Guangxi Experiment Center of Information Science (Guilin University of Electronic Technology), Guilin, Guangxi* 541004)
[4](*School of Information Science & Engineering, Hebei University of Science and Technology, Shijiazhuang* 050018)

Abstract At the age of big data, data stream classification is being applied to many fields, like spam filtering, market predicting, and weather forecasting, et al, in which recurring concept is an important character. Aiming to reduce the influence of negative transfer and improve the lag of detection of concept drift, RC-OTL is proposed for mining recurring concepts in data stream based on online transfer learning strategy. When a concept drift is detected, RC-OTL selects one current base classifier to store, and then computes the domain similarities between the current training samples and the stored classifiers, in order to select the most appropriate source classifier to combine with a new classifier for learning the upcoming samples, which results in knowledge transfer from the source domain to the target domain. In addition, RC-OTL can select appropriate classifier to classify when the current classification accuracy is detected below a given threshold before concept drift detection. The preliminary theory analysis explains why RC-OTL can reduce negative transfer effectively, and the experiment results further illustrates that RC-OTL can efficiently promote the cumulate accuracy of data stream classification, and faster adapt to the samples of new concept after concept drift takes place.

Key words concept drift; transfer learning; recurring concept; online learning; negative transfer

摘 要 随着大数据时代的到来,数据流分类被应用于诸多领域,如:垃圾邮件过滤、市场预测及天气预报等.重现概念是这些应用领域的重要特点之一.针对重现概念的学习与分类问题中的"负迁移"和概念漂移检测的滞后性,提出了一种基于在线迁移学习的重现概念漂移数据流分类算法——RC-OTL.

收稿日期:2016-03-21;**修回日期**:2016-06-04
基金项目:国家自然科学基金项目(61363029,U1501252);广西区自然科学基金项目(2014GXNSFAA118395);广西区科学研究与技术开发项目(桂科攻 14124005-2-1);广西信息科学中心项目(YB408)
This work was supported by the National Natural Science Foundation of China (61363029, U1501252), the Natural Science Foundation of Guangxi District (2014GXNSFAA118395), Guangxi Scientific Research and Technology Development Project (14124005-2-1), and the Program of Guangxi Experiment Center of Information Science (YB408).

RC-OTL 在检测到概念漂移时存储刚学习的一个基分类器,然后计算最近的样本与存储的各历史分类器之间的领域相似度,以选择最适合对后续样本进行学习的源分类器,从而改善从源领域到目标领域的知识迁移.另外,RC-OTL 还在概念漂移检测之前根据分类准确率选择合适的分类器对后续样本分类.初步的理论分析解释了 RC-OTL 为什么能有效克服"负迁移",实验结果进一步表明:RC-OTL 的确能有效提高分类准确率,并且在遭遇概念漂移后能更快地适应后续样本.

关键词 概念漂移;迁移学习;重现概念;在线学习;负迁移

中图法分类号 TP391

随着大数据时代的到来,数据流分类算法被应用于许多领域.比如:天气预报、信用卡欺诈分类、海啸与地震的预测、商场对顾客购买兴趣与偏好的掌控及产品质量检测等等.这些问题的共有特点是:不断产生的数据形成流;数据流没有终点;数据流中数据包含的概念随时可能产生变化.数据流中这种概念的变化被称为概念漂移[1].不同于传统的机器学习算法,概念漂移数据流分类基于一个动态的学习环境,每当概念发生变化后分类器必须进行调整以适应新到概念.特别是,有时某些概念会在数据流中重复出现.比如:天气随季节而变化、客户的购买兴趣会受季节变化以及流行潮流变化的影响、股市行情的变化等.这种重复出现的概念被称为重现概念.在概念漂移数据流分类中,如果事先存储学习过的概念,当概念重现时,就可以选择这些存储的历史概念对新到样本实施分类和学习,这必将大大减小分类器的学习花费,并提升分类准确率.

近些年来,在处理概念漂移数据流分类问题上取得了很多研究成果.Kuncheva 等人[2]、Tsymbal 等人[3]、王涛等人[4]、Zliobaite 等人[5]、Hoens 等人[6]、Gama 等人[7-8]及文益民等人[9]先后对概念漂移数据流分类进行了较深入的综述.CVFDT[10],AWE[11],WMA[12],DWM[13],KnnM-IB[14]等算法有效地提升了数据流分类的准确率.但是,这些分类算法都在检测到概念漂移后,分类器需重新学习新到概念,原来学习到的分类器被丢弃.在有重现概念的数据流中,这将导致分类器学习重现概念的花费较大,而且适应重现概念的速度会更慢.为克服此弱点,处理重现概念漂移的数据流分类算法,如 FLORA3[15],EB[16],EAE[17],RCD[18],CCP[19],MM-DD[20]及 MM-PRec[21]算法等,都在学习过程中存储学习过的概念的相关信息,如样本、分布特征或分类器等.当检测到重现概念后,通过检索存储的历史概念的信息,以选择适合重现概念的分类器以对后续样本进行分类和学习.但是,当后续样本与所选择的分类器不属于同一分布时会产生"负迁移"现象,导致分类准确率下降.另外,已有数据流分类算法一般都需收集到一定数量的新到样本后再实施概念漂移检测,然后才调整当前分类器,从而无法避免概念漂移检测的滞后性.若概念漂移发生在检测之前,这会导致当前分类器对新到样本的分类准确率降低.由于现有各种机器学习算法本质上都是基于一个静态学习环境,而以尽量保证学习系统之泛化能力为目标的寻优过程,以上这些问题给包含重现概念的数据流分类带来了很大的挑战.

针对上面提到的这 2 个问题,在 Zhao 等人[22]提出的基于在线迁移学习的概念漂移学习算法——CDOL 的启发下,本文提出了一种基于在线迁移学习的重现概念漂移数据流分类算法——RC-OTL.RC-OTL 做了 3 个方面的尝试:

1) CDOL 在检测到概念漂移后才会调整"源"分类器.由于概念漂移检测的滞后性,它可能会将刚学习过的概念作为源领域,从而导致"负迁移".不同于 CDOL,RC-OTL 在检测到概念漂移时,按照一定机制选择一个基分类器作为历史分类器存储,再从存储的各历史分类器中选择与新到样本"负相似度"最小的基分类器作为"源"分类器,将其与一个新建的基分类器组合成一个集成分类器,以对后续样本进行在线学习,从而实现从某个历史领域到新领域的知识迁移.

2) 证明了在"负迁移"较大的情形下 HomOTL-I[22]能很快减弱"负迁移"的影响.这能解释 CDOL 和 RC-OTL 为什么能有效克服"负迁移".

3) 提出一个当前分类器的调整算法以减少在检测到概念漂移之前发生概念漂移导致的分类准确率下降的程度.

1 相关工作

近年来,数据流中的概念重现问题引起了研究

者的密切关注. Widmer 等人[15]较早地注意到了数据流分类中的概念重现问题. 他们提出的 FLORA3 使用 3 种基于"特征-值"对的描述项来表示和存储概念.

Ramamurthy 等人[16]提出了 EB 算法. EB 每获得一个数据块 D 及各样本的类别后,判断分类器全局集合 G 中是否有分类器能较准确地对 D 分类. 若找不到这样的分类器,则从 G 中选择若干分类器构成集成分类器,并判断该集成分类器是否也能较准确地对 D 分类,否则利用 D 训练一个新分类器加入 G. 这样,当概念重现时就可以从 G 中选择到合适的分类器对后续样本分类. Jackowski 等人[17]提出了 EAE 算法. EAE 使用存储的分类器构成集成分类器对新到样本分类. 当分类准确率发生变化时,EAE 就把用新到样本训练的分类器存储. 若进一步检测到概念漂移,EAE 就利用进化算法更新集成分类器,以使新的集成分类器更适合对后续样本分类. Gonçalves 等人[18]提出了一种处理重现概念漂移的算法框架 RCD. RCD 使用 DDM[23]检测概念漂移. 当处于警告状态时,算法会另建一个备用分类器 C_n 学习新到样本,同时将相应的新到样本存入缓存 b_n;当检测到概念漂移时,再采用多元非参统计方法判断 b_n 中的样本是属于已学习概念还是新概念. 若是新概念,则将 C_n 存储,并将当前分类器调整为 C_n;若属于已学习概念,则利用 b_n 从历史分类器中选择最合适的分类器作为当前分类器. 若在后续学习中没有检测到概念漂移,则删除 C_n 和 b_n. Gomes 等人[24]提出了一种基于语境信息的重现概念漂移数据流分类算法. 该算法与 RCD 基本相似. 不同之处在于:当检测到概念漂移时,计算 C_n 与存储的历史分类器的相似度,以判断新到概念是否是重现概念. 若非,则将 C_n 存储,C_n 作为当前分类器;若是,则利用历史分类器构造一个集成分类器作为当前分类器. Katakis 等人[19]提出了 CCP 算法,CCP 用概念向量表达和存储多个概念. 概念向量由样本特征与类别间的概率关系组成. 一个数据块对应一个概念向量,计算不同数据块对应的概念向量之间的距离,可判断是否发生概念漂移或进一步判断是否是重现概念;如果是新概念,则将对应的概念向量存储. 另外,通过概念向量的增量聚类可实现属于同一概念的概念向量的合并.

与以上思路有些不同,Gama 等人[20]、Angel 等人[21]分别提出了基于元学习器与基学习器的分层框架. 这 2 种分层框架的基本特点在于:使用基学习器对样本进行分类和学习;使用元学习器对概念漂移产生的情境,对应每个概念的样本及分类器的分类情况等进行学习. 通过对概念间相似性的判断,以选择重现概念对应的分类器重用. 他们认为使用元学习器比跟踪新到样本的分布或分类准确率能更准确地检测到概念漂移.

迁移学习[25]是机器学习的热门研究领域,已经在推荐系统、文本分类等领域[26-27]取得很好的效果. 它的基本思想是:当目标领域的知识获取十分困难时,可以将相关源领域的知识迁移到目标领域中辅助分类器学习目标领域的知识. Zhao 等人[22]较早地将迁移学习应用于概念漂移数据流分类,提出了算法 CDOL. 在该算法中,当前分类器由 2 个基分类器带权组合而成,其中基分类器 $\boldsymbol{w}_t$ 对应新到概念,另一个 $\boldsymbol{w}_s$ 对应"源"概念. 使用当前分类器对新到样本分类,获得样本类别后,按照迁移学习的思路更新当前分类器. CDOL 的特点在于:检测到概念漂移后,$\boldsymbol{w}_s$ 被替换成 $\boldsymbol{w}_s$ 与 $\boldsymbol{w}_t$ 中的权值更大者. CDOL 的不足之处在于:由于概念漂移检测的滞后性,概念漂移产生时,它可能将刚学习过的旧概念作为源领域,这将导致较大"负迁移". 同时,CDOL 不存储学习过的历史概念,不适应有重现概念的数据流分类. 本文受此启发提出了基于在线迁移学习的重现概念漂移数据流分类算法——RC-OTL.

2 RC-OTL 算法

2.1 问题定义与迁移学习

定义数据流 DS 为按时间到达的 T 个样本,也就是有:

$$DS=\{(\boldsymbol{x}_t,y_t)\mid t=1,2,\cdots,T\},$$

其中,$(\boldsymbol{x}_t,y_t)\in X\times Y,\boldsymbol{X}\in\mathbb{R}^m,Y=\{-1,+1\}$.

为了实现对有重现概念数据流的学习与分类,本文借鉴了 Zhao 等人[22]提出的在线迁移学习算法 HomOTL-I. 根据 HomOTL-I,在学习 DS 的过程中,要以一个学习过的历史概念 C_{old} 对应的线性分类器 $y=\text{sgn}(\boldsymbol{w}_{\text{old}}^{\text{T}}\boldsymbol{x})$(其中 $\boldsymbol{w}_{\text{old}}$ 为权值向量)为基础实施迁移学习. 本文中在不产生歧义的条件下,为方便起见,用权值向量表示用该权值向量构成的线性分类器.

设新到概念为 C_{new},将其作为目标领域. 为将从历史概念 C_{old} 上学习到的知识迁移到对新到概念的学习中,将历史概念 C_{old} 设为源领域,记 $\boldsymbol{w}_s=\boldsymbol{w}_{\text{old}}$,并新建分类器 $\boldsymbol{w}_t=0$,将 $\boldsymbol{w}_s$ 和 $\boldsymbol{w}_t$ 作为基分类器加权

组合为一个集成分类器 f,作为当前分类器对新到样本进行分类和学习.新到样本存入缓存 PS 中.f 的定义如下:

$$f: y=\mathrm{sgn}(\alpha_1 \Pi(\boldsymbol{w}_s^{\mathrm{T}}\boldsymbol{x})+\alpha_2 \Pi(\boldsymbol{w}_t^{\mathrm{T}}\boldsymbol{x})-\frac{1}{2}), \quad (1)$$

其中 α_1 和 α_2 为基分类器的权重系数.当新建分类器 f 时初始化 α_1 和 α_2 为 1/2,$\Pi(z)$为一个压缩函数[22].

当前分类器 f 每获取到一个样本 $\boldsymbol{x}_t$ 及其真实类别 y_t 后,使用($\boldsymbol{x}_t, y_t$)对分类器采用 HomOTL-I 的更新策略进行更新,更新步骤如下:

1) 计算损失函数 $l_t=\max(0,1-y_t\boldsymbol{w}_t^{\mathrm{T}}\boldsymbol{x}_t)$;

2) 如果 $l_t>0$,则利用损失函数 l_t 对 $\boldsymbol{w}_t$ 进行更新,即 $\boldsymbol{w}_{t+1}=\boldsymbol{w}_t+\tau y_t\boldsymbol{x}_t$,其中 $\tau=\min(C, l_t/\|\boldsymbol{x}_t\|^2)$;

3) α_1 和 α_2 的更新规则为

$$\alpha_{1,t+1}=\frac{\alpha_{1,t}s_t(\boldsymbol{w}_s)}{\alpha_{1,t}s_t(\boldsymbol{w}_s)+\alpha_{2,t}s_t(\boldsymbol{w}_t)},$$
$$\alpha_{2,t+1}=\frac{\alpha_{2,t}s_t(\boldsymbol{w}_t)}{\alpha_{1,t}s_t(\boldsymbol{w}_s)+\alpha_{2,t}s_t(\boldsymbol{w}_t)}, \quad (2)$$

其中,$s_t(w)=\exp\{-\eta l^*(\Pi(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_t), \Pi(y_t))\}$,$\eta=0.5$,$\forall \boldsymbol{w}\in\mathbb{R}^m$,$l^*(z,y)=(z-y)^2$,$t$ 为新到样本序号.

2.2 重现概念的检测

在存在重现概念漂移的问题中,需要判断新到概念是否发生概念漂移.若是概念漂移,则需进一步判断是学习过的历史概念还是一个新概念.与 CDOL 类似,RC-OTL 采取每隔 p 个样本就进行一次概念漂移检测的策略.概念漂移检测的方法可采取已有的各类概念漂移检测算法.本文采取了 Zhao 等人[22]提出的 OWA 算法.若没有检测到概念漂移,则按照 HomOTL-I 算法继续进行在线学习.

若检测到了概念漂移,RC-OTL 首先需将当前分类器 f 的基分类器 $\boldsymbol{w}_t$ 根据如下规则加入 HS:设 α_1 和 α_2 为当前分类器 f 中对应基分类器 $\boldsymbol{w}_s$ 和 $\boldsymbol{w}_t$ 的权重系数.若 $\alpha_1<\alpha_2$,则将 $\boldsymbol{w}_t$ 存入 HS.然后,RC-OTL 需要根据新到样本集 PS 从 HS 中选择最合适的基分类器作为"源"分类器,以便实现知识迁移.为从 HS 中选择这样的分类器,本文引入洪佳明等人[28]提出的一种当目标领域标记样本很少的情况下度量 2 个领域相似性的指标——负相似度.他认为 2 个领域的负相似度越小,则这 2 个领域越相似,其定义如下:

设 $P_S(X,Y)$和 $P_T(X,Y)$分别为源领域和目标领域的概率分布.设 ρ 是一个小于 1 的正数,v 为一正数.若存在分类器 F,使得 F 在 $P_T(X,Y)$上的泛化误差小于 ρ,且能以 $1-v$ 的概率推断,对于$(x,y)\sim P_S(X,Y)$有 $F(x)=y$ 成立,则称 P_S 以 $1-v$ 的概率 ρ 弱相似于 P_T,称 v 为负相似度,记为 $v(F,P_S,P_T)$.负相似度的优点在于:在已获取的属于新到概念的样本很少的情况下就能选取合适的历史分类器.

设 $HS=\{\boldsymbol{w}_{s1},\boldsymbol{w}_{s2},\boldsymbol{w}_{s3},\cdots,\boldsymbol{w}_{sn},\cdots\}$,参照以上定义,RC-OTL 提出按照式(3)计算 HS 中各分类器的权值向量 $\boldsymbol{w}_{sn}$ 与新到样本集 PS 的负相似度:

$$v_n=\sum_{i=1}^{m}\max(0,1-y_i\boldsymbol{w}_{sn}^{\mathrm{T}}\boldsymbol{x}_i), \quad (3)$$

其中,$(\boldsymbol{x}_t,y_t)\in PS$,$m$ 为新到样本的数量.选择 HS 中与新到样本集 PS 负相似度最小的权值向量 $\boldsymbol{w}_{si}$ 构造线性分类器.一般情况下,它很可能是与后续样本最领域相似的分类器.然后新建分类器 $\boldsymbol{w}_t=\boldsymbol{0}$,按照式(1)建立当前分类器 f,以对后续样本实施分类和学习.这样能使得 f 对后续样本的分类准确率较高.

2.3 负迁移的克服

定理 1. 设$(\boldsymbol{x}_1,y_1),(\boldsymbol{x}_2,y_2),\cdots,(\boldsymbol{x}_t,y_t),\cdots,(\boldsymbol{x}_T,y_T)$为 T 个样本的序列,采用 HomOTL-I[22]算法进行学习.设 $l^*_{\boldsymbol{w}_s,t}=l^*(\Pi(\boldsymbol{w}_s^{\mathrm{T}}\boldsymbol{x}_t),\Pi(y_t))$为"源"概念对应的基分类器 $\boldsymbol{w}_s$ 在第 t 个样本上的损失函数;$l^*_{\boldsymbol{w}_t,t}=l^*(\Pi(\boldsymbol{w}_t^{\mathrm{T}}\boldsymbol{x}_t),\Pi(y_t))$为"目标"概念对应的基分类器 $\boldsymbol{w}_t$ 在第 t 个样本上的损失函数.当 $\sum_{i=1}^{t-1}l^*_{\boldsymbol{w}_s,i}>\sum_{i=1}^{t-1}l^*_{\boldsymbol{w}_t,i}$,$t=2,3,\cdots,T$ 时,存在 $\Delta>0$,$k>0$,使得 $\alpha_{1,t}<ke^{-\Delta t}$.

证明. 当 $t=2,3,\cdots,T$ 时,根据式(2)递推可知:

$$\frac{\alpha_{2,t}}{\alpha_{1,t}}=\frac{\exp(-\eta\sum_{i=1}^{t-1}l^*_{\boldsymbol{w}_t,i})}{\exp(-\eta\sum_{i=1}^{t-1}l^*_{\boldsymbol{w}_s,i})}=\exp(\eta\sum_{i=1}^{t-1}l^*_{\boldsymbol{w}_s,i}-\eta\sum_{i=1}^{t-1}l^*_{\boldsymbol{w}_t,i}),$$

因为$\frac{\alpha_{2,t}}{\alpha_{1,t}}=\frac{1-\alpha_{1,t}}{\alpha_{1,t}}$, $\sum_{i=1}^{t-1}l^*_{\boldsymbol{w}_s,i}>\sum_{i=1}^{t-1}l^*_{\boldsymbol{w}_t,i}$,可得:

$$\alpha_{1,t}=\frac{1}{1+\exp(\eta\sum_{i=1}^{t-1}l^*_{\boldsymbol{w}_s,i}-\eta\sum_{i=1}^{t-1}l^*_{\boldsymbol{w}_t,i})}<\frac{1}{2}\exp\Big(\frac{-\eta}{2}(\sum_{i=1}^{t-1}l^*_{\boldsymbol{w}_s,i}-\sum_{i=1}^{t-1}l^*_{\boldsymbol{w}_t,i})\Big).$$

由定义知 $0\leqslant\Pi(x)\leqslant1$,所以有 $0\leqslant l^*_{\boldsymbol{w}_s,t}\leqslant1$,从而得到 $0\leqslant\sum_{i=1}^{t-1}l^*_{\boldsymbol{w}_s,i}\leqslant t-1$.

由于事先假定 $\sum_{i=1}^{t-1} l^*_{w_s,i} > \sum_{i=1}^{t-1} l^*_{w_i,i}$,所以有 $0 < \sum_{i=1}^{t-1} l^*_{w_s,i} - \sum_{i=1}^{t-1} l^*_{w_i,i} \leqslant \sum_{i=1}^{t-1} l^*_{w_s,i} \leqslant t-1$. 故存在 $0 < \delta_t \leqslant 1$,有 $\sum_{i=1}^{t-1} l^*_{w_s,i} - \sum_{i=1}^{t-1} l^*_{w_i,i} = \delta_t \times (t-1)$,使得:

$$\alpha_{1,t} < \frac{1}{2}\exp\left(\frac{-\eta\delta_t}{2}(t-1)\right).$$

令 $\delta = \min\{\delta_2, \delta_3, \cdots, \delta_T\}$,则有:

$$\alpha_{1,t} < \frac{1}{2}\exp\left(\frac{-\eta\delta}{2}(t-1)\right).$$

再令 $\Delta = \frac{\delta\eta}{2}$, $k = \frac{\exp(\frac{\delta\eta}{2})}{2}$,从而有 $\alpha_{1,t} < ke^{-\Delta t}$.

证毕.

从定理 1 可以看出:"源"分类器对于目标领域存在较大负迁移时,HomOTL-I 算法将使"源"分类器的权重系数呈指数级衰减. 因此,相比于 RCD,由于 RC-OTL 以 HomOTL-I 算法为基本学习算法. 在对概念漂移数据流分类时,RC-OTL 能较有效地克服负迁移的影响. 相比于同样以 HomOTL-I 算法为基本学习算法的 CDOL,RC-OTL 会选择负迁移更小的历史分类器,从而能更快地适应新概念.

2.4 当前分类器的调整

RC-OTL 还根据当前分类器对每个窗口前半段样本的分类错误率 e,判断算法是否需要进入当前分类器调整阶段. 算法定义 e 如下:

$$e = \frac{2 \times Nerr}{p},$$

其中,$Nerr$ 为当前分类器对前半个窗口样本的分类错误数量,p 为当前窗口的长度.

预先给定一个阈值 $Fbuffer\_err$,当 $e > Fbuffer\_err$ 时,则认为在当前窗口中疑似发生概念漂移. 此时需将当前分类器 f 中的基分类器 $\boldsymbol{w}_t$ 与 HS 中的每一个历史分类器 $\boldsymbol{w}_{sn}$ 各建立一个集成分类器:

$$f_n: y = \mathrm{sgn}(\alpha_{1n}\Pi(\boldsymbol{w}_{sn}^{\mathrm{T}}\boldsymbol{x}_t) + \alpha_{2n}\Pi(\boldsymbol{w}_t^{\mathrm{T}}\boldsymbol{x}_t) - \frac{1}{2}),$$

并将得到的所有集成分类器及当前分类器 f 都存入集合 $POOL$ 中,进入当前分类器调整阶段.

在调整阶段中,每获取到一个样本,通过 BufferSelect 算法从 $POOL$ 中根据负相似度选择一个集成分类器对新到样本进行分类. BufferSelect 算法的详细描述如算法 1. 当获到样本的真实类别后,按照 HomOTL-I 的更新策略更新 $POOL$ 中所有的集成分类器.

算法 1. BufferSelect.

输入:分类器池 $POOL$、新到样本的集合 PS;

输出:f.

① $V = \varnothing$;

② for each w_{sn} in HS

③ $\quad v_n = \sum_{i=1}^{m} \max(0, 1 - y_i \boldsymbol{w}_{sn}^{\mathrm{T}} \boldsymbol{x}_i)$, $(\boldsymbol{x}_i, y_i) \in PS$;

④ $\quad V.add(v_n)$;

⑤ endfor

⑥ $\bar{v} = \min(V)$; /* 选取记录的负相似度中最小的负相似度 */

⑦ for $i = 1$ to $length(HS)$

⑧ $\quad$ if $v_n == \bar{v}$

⑨ $\quad\quad f = POOL[n]$; /* 选取当前分类器 */

⑩ $\quad\quad$ break;

⑪ $\quad$ endif

⑫ endfor

⑬ return f.

2.5 RC-OTL 算法描述

RC-OTL 的详细描述如算法 2 所示. 行①~③是算法的初始化阶段;行⑦~⑫是对当前分类器进行调整;行⑭~⑲是在没有检测到概念漂移情况下的在线分类与学习;行㉑~㉞是为进入当前分类器调整阶段进行准备. 在这个阶段中,由于要对当前分类器进行实时调整,需将进入调整阶段前的当前分类器存储:$f_{\mathrm{memory}} = f$. 若最终没有发生概念漂移,则需恢复进入调整阶段前的当前分类器;行㊱~㊾是进行分类器存储与分类器选择以实施迁移学习.

算法 2. RC-OTL.

输入:数据流 DS、窗口大小 p、分类器选择阈值 $Fbuffer\_err$;

① $\boldsymbol{w}_s = \mathbf{0}; \boldsymbol{w}_t = \mathbf{0}; bufferflag = 0; Nerr = 0;$ $f_{\mathrm{memory}} = 0; POOL = \varnothing$; /* 初始化阶段 */

② $PS = \varnothing; HS = \varnothing$; /* 储存新到样本和存储历史分类器 */

③ $f: y = \mathrm{sgn}(\boldsymbol{w}_t^{\mathrm{T}}\boldsymbol{x})$; /* 初始化当前分类器 f */

④ for $t = 1$ to T do

⑤ $\quad$ receive instance $x_t \in X$;

⑥ $\quad$ if $bufferflag > 0$ then /* 进入当前分类器调整阶段 */

⑦ $\quad\quad f = bufferSelect(POOL, PS)$; /* 使用 $bufferSelect$ 调整当前分类器 */

⑧ $\quad\quad y = f.classify(\boldsymbol{x}_t)$; /* 使用当前分类器对样本 $\boldsymbol{x}_t$ 分类 */

⑨ receive the correct label: $y_t \in \{-1,+1\}$;
⑩ for each f_i in $POOL$
⑪ $f_i.update(\boldsymbol{x}_t, y_t, y)$ /* 按照式(2)的在线学习规则对 f_i 进行更新 */
⑫ endfor
⑬ else /* 进入正常分类阶段 */
⑭ $y = f.classify(\boldsymbol{x}_t)$; /* 使用当前分类器对样本 $\boldsymbol{x}_t$ 分类 */
⑮ receive the correct label: $y_t \in \{-1,+1\}$
⑯ if $y \neq y_t$ then
⑰ $Nerr = Nerr + 1$; /* 当前窗口错误数加 1 */
⑱ endif
⑲ $f.update(\boldsymbol{x}_t, y_t, y)$; /* 当前分类器在线学习样本 $\boldsymbol{x}_t$ */
⑳ endif
㉑ if $\mathrm{mod}(t, p/2) == 0$ then/* 检测是否将进入当前分类器调整阶段 */
㉒ $e = 2 \times Nerr/p$; /* 计算当前分类器对前半个窗口样本的分类错误率 e */
㉓ if $e > Fbuffer\_err$ then
㉔ $buffereflag = 1$; /* 设置标记,启动当前分类器调整准备工作 */
㉕ for each $\boldsymbol{w}_{sn}$ in HS
㉖ if $\boldsymbol{w}_s \neq \boldsymbol{w}_{sn}$ then
㉗ $f_n = create(\boldsymbol{w}_{sn}, \boldsymbol{w}_t)$; /* 当前 $\boldsymbol{w}_{sn}$ 与 HS 的其他 $\boldsymbol{w}_t$ 按照式(1)的方式组成集成分类器 */
㉘ $POOL.add(f_n)$; /* 将分类器存入分类器缓存池 */
㉙ endif
㉚ endfor
㉛ $f_{\mathrm{memory}} = f$; /* 保存进入调整阶段前的当前分类器 */
㉜ $POOL.add(f)$;
㉝ endif
㉞ endif
/* 当前分类器调整阶段准备工作结束 */
㉟ $PS.add((\boldsymbol{x}_t, y_t))$;
/* 缓存新到样本$(\boldsymbol{x}_t, y_t)$ */
㊱ if $\mathrm{mod}(t, p) == 0$ then /* 每隔 p 个样本检测一次概念漂移 */
㊲ $level = OWA(PS)$; /* 使用 OWA 算法检测是否发生概念漂移,发生返回 true,否则返回 flase */
㊳ if $level ==$ true then
/* 进入漂移阶段 */
㊴ $HS.add(\boldsymbol{w}_t)$; /* 将 $\boldsymbol{w}_t$ 按照规则存入 HS */
㊵ $V = \varnothing$; /* 用于记录 HS 中每个历史分类器与新到样本集 PS 的负相似度 */
㊶ for each $\boldsymbol{w}_{sn}$ in HS
㊷ $v_n = \sum_{i=1}^{p} \max(0, -y_i \boldsymbol{w}_{sn}^{\mathrm{T}} \boldsymbol{x}_i)$, $(\boldsymbol{x}_i, y_i) \in PS$;
㊸ $V.add(v_n)$;
㊹ endfor
㊺ $\bar{v} = \min(V)$; /* 最小负相似度 */
㊻ for each $\boldsymbol{w}_{sn}$ in HS /* 选取 HS 中与新到样本集 PS 最相似的历史分类器 */
㊼ if $v_n == \bar{v}$ then
㊽ $\boldsymbol{w}_s = \boldsymbol{w}_{sn}$;
㊾ endif
㊿ endfor
(51) 新建分类器 $\boldsymbol{w}_t = \boldsymbol{0}$;
(52) $f = create(\boldsymbol{w}_s, \boldsymbol{w}_t)$;
/* 建立当前分类器 */
(53) else
/* 未发生概念漂移进入正常阶段 */
(54) if $buffereflag == 1$ then
(55) $f = f_{\mathrm{memory}}$; /* 恢复进入调整阶段前的当前分类器 */
(56) endif
(57) endif
(58) $buffereflag = 0$, $POOL = \varnothing$, $PS = \varnothing$, $Nerr = 0$, $f_{\mathrm{memory}} = 0$;
(59) endif
(60) endfor

3 实验结果与分析

3.1 实验数据与实验方法

为了验证 RC-OTL 算法的有效性,本文使用了 5 个数据集[①]:email_list, MITface, usps, usenet1 和 usenet2. 其中,email_list, usenet1 和 usenet2 都是基于

① http://mlkd.csd.authgr/concept_drift.html

UCI 的 newsgroups20 数据集,描述了在一段时间内用户对医药、太空和篮球的兴趣的变化,其中"+"代表感兴趣,"−"代表不感兴趣. MITface 来自 MIT 的人脸数据库. Usps 来自美国邮政手写数字数据集. email_list, MITface, usps, usenet1 和 usenet2 的特征维数分别为 913, 361, 256, 99 和 99. 这些数据集的具体描述如表 1～5 所示. 我们将在这些数据集上对比概念漂移数据流分类算法 PA-I[29], CODL, RCD 及本文提出的 RC-OTL, RC-OTL-I(不对当前分类器进行调整)算法的累积分类准确率,累积分类准确率的变化以及实时分类准确率(实时分类准确率每 30 个样本计算一次)的变化. 其中,PA-I 是一个在线学习算法,它不进行概念漂移检测. PA-I 和 CODL 均使用 Zhao 等人提供的源程序①. CODL, RCD, RC-OTL 和 RC-OTL-I 均使用 PA-I 作为基本学习算法.

Table1 email_list Data Set

表 1 email_list 数据集

Example ID	Medicine	Space	Baseball
1～300	+	−	−
301～600	−	+	+
601～900	+	−	−
901～1 200	−	+	+
1 201～1 500	+	−	−

Table 2 MITface Data Set

表 2 MITface 数据集

Example ID	face	Non-face
1～1 500	+	−
1 501～3 000	−	+
3 001～4 500	+	−
4 501～6 000	−	+
6 001～6 977	+	−

Table 3 usps Data Set

表 3 usps 数据集

Example ID	Images Including the digits 1, 2, and 3	
	1	2,3
1～400	+	−
401～1 200	−	+
1 201～2 400	+	−
2 401～2 930	−	+

① http://www.stevenhoi.org/OTL/

Table4 usenet1 Data Set

表 4 usenet1 数据集

Example ID	Medicine	Space	Baseball
1～300	+	−	−
301～600	−	+	+
601～900	+	−	−
901～1 200	−	+	+
1 201～1 500	+	−	−

Table 5 usenet2 Data Set

表 5 usenet2 数据集

Example ID	Medicine	Space	Baseball
1～300	+	−	−
301～600	−	+	−
601～900	−	−	+
901～1 200	−	+	−
1 201～1 500	+	−	−

实验中采用与 Zhao 等人[22]相同的参数:惩罚系数 $C=5$,窗口大小 $p=30$. 将数据集中样本投影到高维平面采用的核函数为高斯径向函数,其参数 $\sigma=8$. RCD 使用了 R 语言扩展包[30]中提供基于 KNN 的多元非参统计测试方法检测新到概念是否为历史概念,其中最近邻数 $K=3$,相似参数 $p\_value=0.01$,同时 RCD 最多储存的历史分类器数为 15. 另外,设置 $Fbuffer\_err=7$.

为了比较各个算法的分类准确率和在发生概念漂移时分类准确率的变化,实验给出了各算法在每个数据集上的累积分类准确率和实时分类准确率. 实验数据集分别被按照概念随机打乱,实验重复 20 次. Zhao 等人提供了重复 20 次实验所需的数据集①. 实验结果为 20 次实验的平均值. 实验结果分别由表 6、图 1 及图 2 描述.

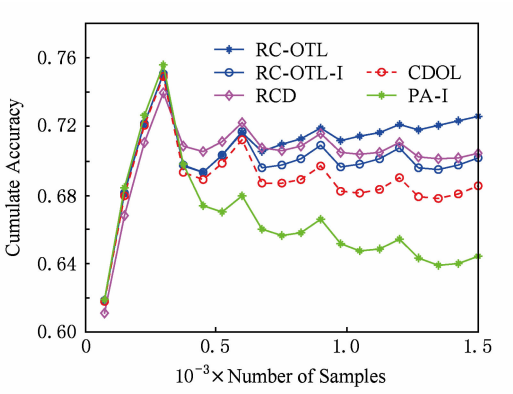
3.2 实验结果分析

表 6 和图 1 提供了各个算法的累积分类准确率情况. 在图 1 中,每个数据集均只提供了 20 个时间点的累积分类准确率. 从表 6 可以看出,RC-OTL 的累积分类准确率在上述每个数据集上均优于其他算法. 图 1 进一步表明:RC-OTL 的累积分类准确率与其他算法的差距随着样本的增加而不断增大. RC-OTL-I 的累积分类准确率比 RC-OTL 稍低,但多数情况下也优于 CDOL 和 RCD. 这说明:RC-OTL 所采取的利用负相似度选择最适合于新到样本的基分类器的策略有效地减少了"负迁移",从而有效地提高了累积分类准确率.

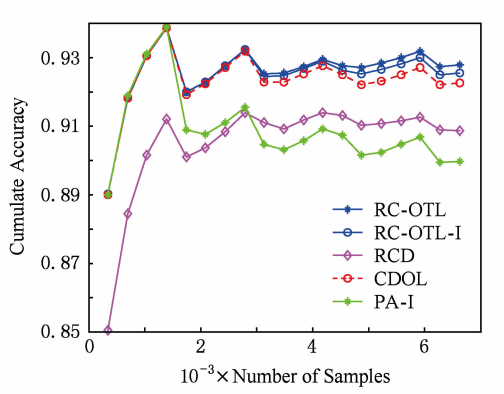
Table 6 Cumulate Classification Accuracy

表 6 累积分类准确率 %

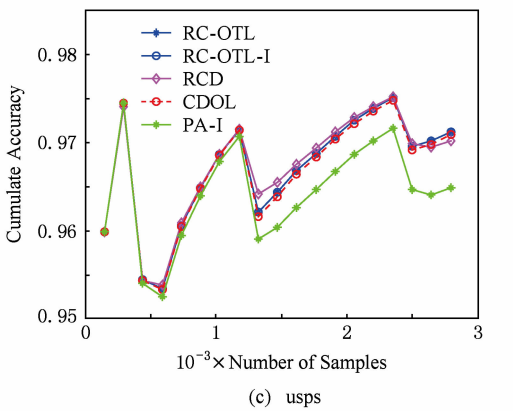
Algorithm	email_list	MITface	usps	usenet1	usenet2
PA-I	64.4667±0.7849	90.1397 ± 0.1413	96.5734±0.2289	59.1167±0.7807	59.1533±0.8414
CDOL	68.5467±1.6738	92.3749 ± 0.2854	97.1792±0.2188	60.3467±1.6469	64.7900±1.3295
RC-OTL	**72.5800±2.5222**	**92.8658± 0.5969**	**97.2167±0.2143**	**64.3467±2.4429**	**66.2100±1.3543**
RC-OTL-I	70.2000±2.0554	92.6580± 0.4626	97.2167±0.2143	62.0100±1.6680	66.1000±1.7139
RCD	70.4633±1.5277	90.9030 ± 1.0347	97.1092±0.2561	58.3367±0.8691	58.0467±1.0330



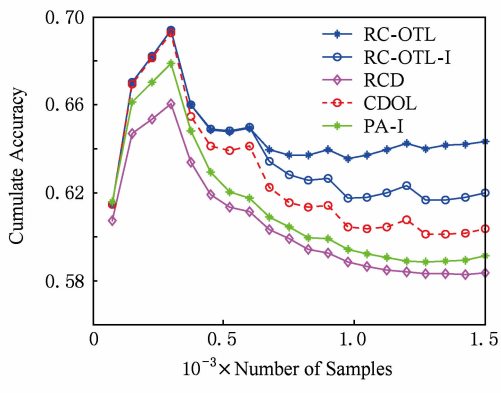
(a) email_list



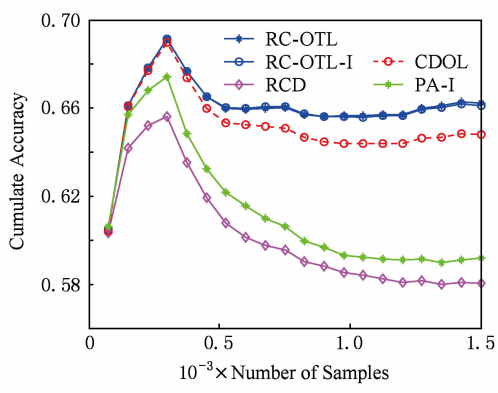
(b) MITface



(c) usps



(d) usenet1



(e) usenet2

Fig. 1 The variation of cumulate classification accuracy.

图 1 累积分类准确率变化图

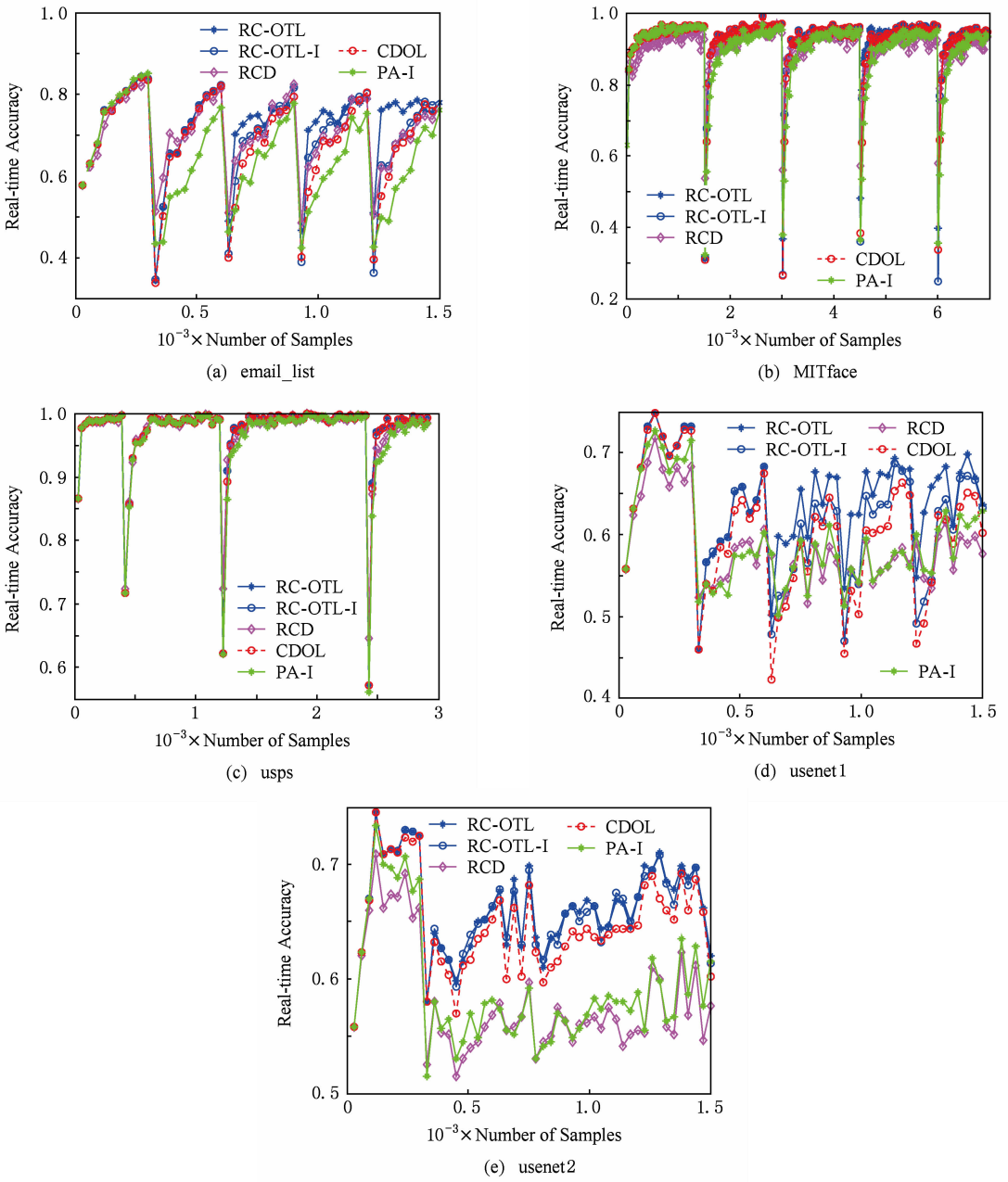


Fig. 2 The variation of real-time classification accuracy.

图 2 实时分类准确率变化图

图 2 描述了各算法在数据流分类与学习过程中实时分类准确率的变化情况. 从图 2 可以看出,与 CDOL 和 RCD 相比,RC-OTL 和 RC-OTL-I 能在遭遇概念漂移后显示出对新到样本更高的实时分类准确率,这说明 RC-OTL 和 RC-OTL-I 能更快地适应新到样本. 这表明:概念检测后选择负相似度最小的分类器为在线迁移学习提供了更好的基础. 更进一步地,由于 RC-OTL 的权重调整机制,能迅速减少由于"负迁移"对当前分类器造成的影响,而 RCD 则难以做到. 因此,RC-OTL 能更快地适应重现概念. 另外,从图 2 还可以看到:在多数情况下,当第 1 次概念漂移发生后,RC-OTL 在以后各次概念漂移发生时它的分类准确率的下降程度都比第 1 次概念漂移发生时的分类准确率的下降程度要低. 这说明:若概念漂移发生在概念漂移检测之前,RC-OTL 会利用负相似度调整当前分类器,使分类准确率不至于下降得太多.

4 总 结

本文针对重现概念的学习与分类问题中的"负迁移"和概念漂移检测的滞后性提出了一种基于

在线迁移学习的重现概念漂移数据流分类算法——RC-OTL. RC-OTL 存储学习过的历史分类器,计算新到样本与存储的历史分类器之间的负相似度,以选择最适合对后续样本进行分类和学习的基分类器,从而能更好地实现从源领域到目标领域的知识迁移. 初步的理论分析表明了 RC-OTL 的合理性. 实验结果进一步验证了 RC-OTL 的确能有效地提高分类准确率,并且在遭遇概念漂移后,能够很快地适应新到样本,显示了更高的实时分类准确率.

目前,本文只采用 PA 作为基分类器的学习算法,下一步将给出基于其他学习算法的 RC-OTL 算法,并探讨不同概念漂移检测方法对算法性能的影响以及存储的历史分类器的合并与淘汰问题.

参 考 文 献

[1] Schlimmer J, Granger R. Incremental learning from noisy data [J]. Machine Learning, 1986, 1(3): 317-354

[2] Kuncheva L I. Classifier ensembles for changing environments [C] //Proc of the 5th Workshop on Multiple Classifier Systems. Berlin: Springer, 2004: 1-15

[3] Tsymbal A. The problem of concept drift: Definitions and related work, TCD-CS-2004-15 [R]. Dublin: Department of Computer Science, Trinity College, University of Dublin, 2004

[4] Wang Tao, Li Zhoujun, Yan Yuejin, et al. A survey of classification of data streams [J]. Journal of Computer Research and Development, 2007, 44(11): 1809-1815 (in Chinese)
(王涛, 李舟军, 颜跃进, 等. 数据流挖掘分类技术综述[J]. 计算机研究与发展, 2007, 44(11): 1809-1815)

[5] Zliobaite I. Learning under concept drift: An overview, abs/1010.4784 [R]. Vilnius: Vilnius University, 2009

[6] Hoens T R, Polikar R, Chawla N V. Learning from streaming data with concept drift and imbalance: an overview [J]. Progress in Artificial Intelligence, 2012, 1(1): 1-13

[7] Gama J. A survey on learning from data streams: Current and future trends [J]. Progress in Artificial Intelligence, 2012, 1(1): 45-55

[8] Gama J, Zliobaite I, Bifet A, et al. A survey on concept drift adaption [J]. ACM Computing Surveys, 2014, 46(4): 1-37

[9] Wen Yimin, Qiang Baohua, Fan Zhigang. A survey of the classification of data streams with concept drift [J]. CAAI Trans on Intelligent Systems, 2013, 8(2): 96-104 (in Chinese)
(文益民, 强保华, 范志刚. 概念漂移数据流分类研究综述[J]. 智能系统学报, 2013, 8(2): 96-104

[10] Hulten G, Spencer L, Domingos P. Mining time-changing data streams [C] //Proc of the 7th ACM SIGKDD Int Conf on Knowledge Discovery and Data Mining. New York: ACM, 2001: 97-106

[11] Wang H, Fan W, Yu P S, et al. Mining concept-drifting data streams using ensemble classifiers [C] //Proc of the 9th ACM SIGKDD Int Conf on Knowledge Discovery and Data Mining. New York: ACM, 2003: 226-235

[12] Blum A. Empirical support for winnow and weighted-majority algorithms: Results on a calendar scheduling domain [J]. Machine Learning, 1997, 26(1): 5-23

[13] 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

[14] Guo Gongde, Li Nan, Chen Lifei. Concept drift detection for data streams based on mixture model [J]. Journal of Computer Research and Development, 2014, 51(4): 731-742 (in Chinese)
(郭躬德, 李南, 陈黎飞. 一种基于混合模型的数据流概念漂移检测算法[J]. 计算机研究与发展, 2014, 51(4): 731-742)

[15] Widmer G, Kubat M. Learning in the presence of concept drift and hidden contexts [J]. Machine Learning, 1996, 23(1): 69-101

[16] Ramamurthy S, Bhatnagar R. Tracking recurrent concept drift in streaming data using ensemble classifiers [C] //Proc of the 6th Int Conf on Machine Learning and Applications. Piscataway, NJ: IEEE, 2007: 404-409

[17] Jackowski K. Fixed-size ensemble classifier system evolutionarily adapted to a recurring context with an unlimited pool of classifiers [J]. Pattern Analysis Applications, 2014, 17(4): 709-724

[18] Gonçalves P M, Barros R S. RCD: A recurring concept drift framework [J]. Pattern Recognition Letters, 2013, 34(9): 1018-1025

[19] 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

[20] Gama J, Kosina P. Learning about the learning process [C] //Proc of the 10th Int Conf on Advances in Intelligent Data Analysis X. Berlin: Springer, 2011: 162-172

[21] Angel A M, Bartolo G J, Ernestina M. Predicting recurring concepts on data-streams by means of a meta-model and fuzzy similarity function [J]. Expert Systems with Applications, 2016, 46(3): 87-105

[22] Zhao Peilin, Hoi S C H, Wang Jialei, et al. Online transfer learning [J]. Artificial Intelligence, 2014, 216(16): 76-102

[23] Gama J, Medas P, Castillo G, et al. Learning with drift detection [C] //Proc of the 7th Brazilian Symp on Artificial Intelligence. Berlin: Springer, 2004: 286-295

[24] Gomes J B, Menasalvas E, Sousa P A C. Learning recurring concepts from data streams with a context-aware ensemble [C] //Proc of ACM Symp on Applied Computing. New York: ACM, 2011: 994-999

[25] Pan S J, Yang Qiang. A survey on transfer learning [J]. IEEE Trans on Knowledge and Data Engineering, 2010, 22 (10): 1345-1359

[26] Pan Weike, Zhong Hao, Xu Congfu, et al. Ming adaptive bayesian personalized ranking for heterogeneous implicit feedbacks [J]. Knowledge-Based Systems, 2015, 73 (1): 173-180

[27] Pan Weike, Yang Qiang. Transfer learning in heterogeneous collaborative filtering domains [J]. Artificial Intelligence, 2013, 197(4): 39-55

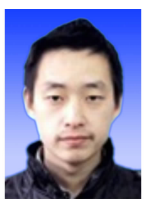
[28] Hong Jiaming, Yin Jian, Huang Yun, et al. TrSVM: A transfer learning algorithm using domain similarity [J]. Journal of Computer Research and Development, 2011, 48 (10): 1823-1830 (in Chinese)
(洪佳明, 印鉴, 黄云, 等. TrSVM: 一种基于领域相似性的迁移学习算法[J]. 计算机研究与发展, 2011, 48(10): 1823-1830)

[29] Crammer K, Dekel O, Keshet J, et al. Online passive-aggressive algorithms [J]. The Journal of Machine Learning Research, 2006, 7(3): 551-585

[30] Chen L, Dai P, Dou W. Mtsknn: Multivariate two-sample tests based on k-nearest-neighbors [CP/OL]. [2016-05-20]. http://cran.rproject.org/web/packages/MTSKNN/index.html

Wen Yimin, born in 1969. PhD. Professor and master supervisor in Guilin University of Electronic Technology. Senior member of China Computer Federation. His main research interests include machine learning, data mining, recommendation systems, big data analysis and online education.

Tang Shiqi, born in 1990. Master candidate in Guilin University of Electronic Technology. His main research interests include machine learning and data mining (tttpgs@163.com).



Feng Chao, born in 1989. Master from Guilin University of Electronic Technology. His main research interests include machine learning and data mining (henryfung01@126.com).

Gao Kai, born in 1968. PhD. Professor and master supervisor in Hebei University of Science and Technology. His main research interests include big data search and mining, natural language processing, Information retrieval, and social computing (gaokai@hebust.edu.cn).