控制与决策
Control and Decision

面向分布式在线学习的共享数据方法

张宇， 刘威， 邵良杉

引用本文:
张宇, 刘威, 邵良杉. 面向分布式在线学习的共享数据方法[J]. 控制与决策, 2021, 36(8): 1871−1880.

在线阅读 View online: https://doi.org/10.13195/j.kzyjc.2019.1811

面向分布式在线学习的共享数据方法

张 宇[1,2†], 刘 威[1], 邵良杉[2]

(1. 辽宁工程技术大学 理学院，辽宁 阜新 123000；
2. 辽宁工程技术大学 管理科学研究中心，辽宁 葫芦岛 125105)

摘 要: 分布式数据流已成为现代数据驱动应用产生数据的主要形式,而局部节点的数据虽然独立存储,但彼此之间是相互关联的,因此如何高效地共享局部节点数据来构建全局学习器是分布式在线学习的关键问题. 针对此问题,提出一种分布式在线学习的数据共享解决方案,包括基于指数损失的半监督聚类方法和基于协方差矩阵与均值向量的数据共享方法,并证明重构数据集的累计绝对误差小于给定绝对误差界的概率下界. 实验表明:所提出的方法可以使节点间的共享数据量维持在一个较低的水平,同时保证基于重构数据训练得到的学习器具有很好的泛化学习能力.

关键词: 分布式数据流；全局学习器；在线学习；数据共享；半监督聚类；数据集重构

中图分类号: TP181 **文献标志码:** A

DOI: 10.13195/j.kzyjc.2019.1811 **开放科学(资源服务)标识码(OSID):**

引用格式: 张宇,刘威,邵良杉. 面向分布式在线学习的共享数据方法[J]. 控制与决策, 2021, 36(8): 1871-1880.

A sharing data approach oriented to distributed online learning

ZHANG Yu[1,2†], *LIU Wei*[1], *SHAO Liang-shan*[2]

(1. College of Science, Liaoning Technical University，Fuxin 123000，China；2. Research Centre in Management Science，Liaoning Technical University，Huludao 125105，China)

Abstract: Distributed data stream generated by current data-driven applications has become a main data representation. Although distributed data stream is captured from different data sources, they are correlated to a common event. Hence, the key issue of distributed online learning is how to build global learners by sharing data of local node. For this problem, this paper proposes a sharing data solution for distributed online learning, containing the semi-supervised clustering approach based on exponential loss and the sharing data approach based on covariance matrixes and mean vectors, and proves the cumulative absolute error between the rebuilding data set and the original data set is bounded on the given threshold under some probability. Experimental study demonstrates that the proposed approach has lower network traffic between nodes, and gets the learner having better generalization capability.

Keywords: distributed data stream；global learner；online learning；sharing data；semi-supervised clustering；rebuilding data set

0 引 言

现有的数据驱动应用产生的数据都是流式数据,被称为数据流(data stream). 数据流可视为随时间增长的数据序列,通常具有容量大、增速快[1]、概念漂移[2-4]和特征漂移[5-7]等特点. 近年来,随着物联网、社交网络等应用的普及,数据流又呈现出存储分散性的特点. 虽然各节点数据流的存储是相互独立的,但是它们表达的概念是相互关联的. 因此,分布式在线学习的关键问题是如何共享局部节点的数据来构建全局学习器. 由于隐私政策以及网络带宽等因素的限制,将各节点数据流传输到一个中心节点来存储和学习是不可行的,这也促生了数据流分布式学习解决方案的出现[8-10]. 然而,数据流的分布式学习同样面临节点间的数据共享问题,为了减少节点间的通讯代价,共享数据不能是实时传输的原始数据,而应为统计数据.

本文提出一种新的面向分布式在线学习的共享数据模式,其工作主要研究具有K个节点的分布式网络在线实时学习问题,并假定网络中的任意两个节点是相连(包括直接相连和间接相连)的. 在每个局

收稿日期: 2019-12-25；**修回日期:** 2020-03-12.
基金项目: 辽宁省教育厅项目(LJ2019QL016)；国家自然科学基金项目(71771111).
责任编委: 阳春华.
†通讯作者. E-mail: 185629623@qq.com.

部节点,利用已标记示例集合对依次到达节点的数据块进行聚类,计算每个簇的协方差矩阵和均值向量并将其作为统计数据共享给其他节点,同时接收其他节点共享的统计数据. 最后,基于所有节点的统计数据生成训练集来更新本地学习器. 本文提出方法具有以下几个优点. 1)在表示相同数量原始数据的条件下,协方差矩阵与均值向量占用的存储空间很小,因此可有效降低网络的通讯代价; 2)由于最终训练局部学习器的数据与原始数据具有相同的结构而并非统计数据,可以灵活选择构建局部学习器的方法.

本文研究内容主要包括: 1)提出基于指数损失的半监督聚类方法; 2)证明重构数据集与原始数据集的累计绝对误差小于给定绝对误差界的概率下界.

1 相关工作

随着分布式系统的发展,多节点独立存储但逻辑上关联的数据流成为一种重要的数据形态,即分布式数据流. 分布式数据流的在线学习给数据流学习提出了新的挑战,即如何共享局部节点的数据来构建全局学习器. 目前,局部节点共享数据的方式大致分为3类: 第1类是共享原始数据[11-14]. 文献[11]使用估计得到的概率密度函数从原始数据集筛选出发生概率大于阈值的示例集合,并将该子集作为共享数据; 文献[12-13]将等长的原始数据块作为共享数据,二者的区别为前者从中心节点共享给局部节点,而后者则正相反; 文献[14]使用MQTT协议将局部节点产生的数据实时地传输给中心节点. 此类方法主要将原始数据或筛选之后的原始数据子集共享给其他节点,全局学习器直接从原始数据学习得到,进而学习效果较好,但这类方法对网络资源和存储资源的需求较高,因此不适用于资源受限的分布式网络环境下的数据挖掘. 第2类是共享基于原始数据的统计信息,其中最为典型的是文献[15]提出的微簇结构. 这种微簇由一个5元组表示,包括示例数量、已标记示例的数量、示例的类别分布和簇心等统计信息,而其组成的微簇集合是局部节点之间共享的统计数据. 文献[16]使用了类似的3元组微簇结构,每个微簇由示例数量、簇心和半径等统计信息构成; 文献[17-18]运用了更为简单的微簇结构,每个微簇只包含簇心和半径两个统计信息,但二者计算微簇半径的方法不同. 与上述相似的微簇结构还有文献[19-20]. 相比共享原始数据,此类方法可有效降低节点间的共享数据量. 但是,当被表达概念较为复杂或易变时,需要相应调整微簇的数量,因此这类方法的共享数据量是不确定的,当共享的微簇数量较多时,又将导致网络通信负担加重. 第3类是共享局部学习器的参数,其中出现较早的是文献[21]提出的基于AdaBoost的分布式在线学习方法. 在每一轮迭代中,它将局部节点学习得到的弱学习器权重共享给中心节点以进一步计算此轮迭代的全局权重,与其相似的共享策略还有文献[22-23]. 不同于集中式共享策略,文献[24]将弱分类器的权重共享给其余的局部节点而并非中心节点; 另外,文献[25]使用待预测示例及其在本地节点计算得到的最近距离作为节点间的共享数据. 第3类方法可有效解决第2类方法面临的问题,但是此类方法需要在同步学习的条件下进行,通常会增加存储器容量的开销并且需要复杂的协同机制来维持节点间的同步学习,因此通常会降低学习效率和预测性能.

本文所提共享方法属于第2类,但与它们有两点不同: 1)本文使用协方差矩阵与均值向量作为共享数据而非微簇集合,由于协方差矩阵与均值向量的维数是一个可确定的常量,无论表达简单、复杂还是易变的概念,共享的数据量是一定的,所以可有效解决微簇结构所面临的共享数据量不确定问题; 2)本文采用局部节点相互共享统计数据的学习方式,这样可避免层次式学习框架引起的中心节点通信开销过大的问题; 3)由于最终训练局部学习器的数据与原始数据具有相同的结构而并非特殊的统计数据,现有的通用学习算法可直接用于构建局部学习器.

2 基于协方差矩阵与均值向量的数据共享方法

2.1 分布式学习框架

假设当前的分布式学习系统有K个节点,每个节点的数据流可视为随时间增长的数据集合,则分布式数据流可视为一组递增的数据集合,定义为$D = \{D_1, D_2, \ldots, D_K\}$. 现在,将节点$i$的数据流$D_i$划分成大小相等的数据块集合,记为$D = \{D_i^1, D_i^2, \ldots, D_i^t, \ldots\}$. 其中: $D_i^j = \{\langle \boldsymbol{x}_i^1, l_i^1\rangle, \langle \boldsymbol{x}_i^2, l_i^2\rangle, \ldots, \langle \boldsymbol{x}_i^n, l_i^n\rangle\}$, $\boldsymbol{x}_i^n$为到达节点i的第n个示例. 不失一般性,假设$\boldsymbol{x}_i^n$为p维随机向量,而l_i^n为$\boldsymbol{x}_i^n$的真实标签(true label),且$l_i^n \in \{\phi, L_1, L_2, \ldots, L_s\}$(如果$l_i^n = \phi$,则$\boldsymbol{x}_i^n$为未标记示例).

在本文中,各局部节点的分类任务是相互独立的,每个节点共享自己的统计数据,并使用其他节点共享的统计数据更新自己的学习器,接着使用更新后的学习器对最新到达节点的数据块进行分类. 具体过程如下:

1) 聚类: 在局部节点i,利用已标记数据集将数据块D_i^j转化为簇集C_i^j(注: 本文假设局部节点i存在可更新的已标记数据集,而已标记数据集的生成及更新不在本文的研究范围,通常可采用人工标记或主动学习的方法获取).

2) 共享: 基于C_i^j计算协方差矩阵集合M_i^j与均值向量集合v_i^j,并将其发送给其他节点,与此同时接收其他节点共享的协方差矩阵集合与均值向量集合.

3) 更新: 基于所有节点的协方差矩阵集合与均值向量集合生成训练数据集A_i^j,接着使用A_i^j更新本地学习器LL_i. LL_i由两个相同的学习器(记为L_1和L_2)组成,两个学习器周期性地交替执行预测,以此保证LL_i能够在线更新与预测. 更新LL_i的策略通常与使用的学习算法有关,例如: 集成学习算法与增量式学习算法的更新策略存在不同,不同的集成学习算法或者不同的增量式学习算法也存在不同的更新策略,这些更新策略不是本文的研究重点,因此没有做更深入的研究与探讨,但本文将在实验部分给出所使用的更新策略.

4) 预测: 使用更新后的学习器对最新到达的数据块进行分类.

图1演示了包含2个节点的分布式学习过程.

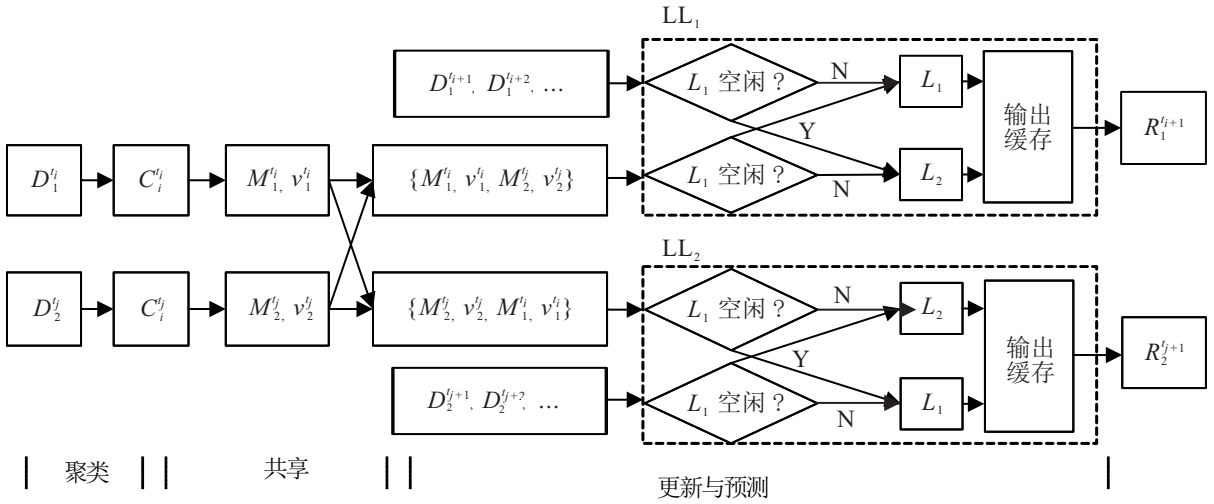

图1　两个节点的分布式学习演示系统

2.2　局部学习器间的数据共享

2.2.1　半监督聚类

假设节点i的第m个数据块D_i^m与已标记数据集B_i构成的混合标记数据集为T_i^j,T_i^j的划分过程即为生成s个簇的聚类过程. 如何利用已标记示例完成T_i^j的划分是半监督聚类(semi-supervised clustering)任务中的关键问题. 为了保证簇内散度最小化和簇内纯度最大化的目标,本文引入指数函数建立优化目标函数E.

$$E=\sum_{k=1}^{s}\sum_{\boldsymbol{x}_i^j\in C_k}L(\boldsymbol{x}_i^j,l_i^j,C_k,\boldsymbol{\mu}_k)\|\boldsymbol{x}_i^j-\boldsymbol{\mu}_k\|_2^2. \tag{1}$$

其中: μ_k为簇C_k的簇心,且有

$$L(\boldsymbol{x}_i^j,l_i^j,C_k,\boldsymbol{\mu}_k)=\exp(-\mathrm{sign}(l_i^j,g(C_k))\|\boldsymbol{x}_i^j-\boldsymbol{\mu}_k\|_1). \tag{2}$$

$\mathrm{sign}(l_i^j,g(C_k))$为指示函数,定义为

$$\mathrm{sign}(l_i^j,g(C_k))=\begin{cases}1, & l_i^j=g(C_k);\\ 0, & l_i^j=\mathrm{null};\\ -1, & \mathrm{otherwise}.\end{cases} \tag{3}$$

这里$g(C_k)$为簇C_k的类别标签.

进一步,设簇C_k中未标记示例所组成的集合为U_k,余下已标记示例组成的集合为R_k,则式(1)可进一步分解为

$$E=\sum_{k=1}^{s}\Big(\sum_{\boldsymbol{x}_i^j\in U_k}\|\boldsymbol{x}_i^j-\boldsymbol{\mu}_{\boldsymbol{k}}\|_2^2+\sum_{\boldsymbol{x}_i^r\in R_k}L(\boldsymbol{x}_i^r,l_i^r,C_k,\mu_k)\|\boldsymbol{x}_i^r-\boldsymbol{\mu}_k\|_2^2\Big). \tag{4}$$

从式(4)容易看出,惩罚函数L可以促使具有相同标记的示例聚类到同一簇,进而减少簇内不同类别标记示例的数量,提高簇的纯度. 最小化式(1)的求解过程与K-均值算法[26]类似,详细过程请见如下算法.

算法1　基于指数损失的半监督聚类.

输入: 原始数据集D_i^m,已标记数据集B_i,聚类簇数s,停止聚类的阈值a;

输出: 簇集W_i.

step 1: **do**

step 2: 从B_i的每个类别子集随机选取一个示例作为相应簇的初始簇心$\mu_i=\{\boldsymbol{\mu}_i^1,\boldsymbol{\mu}_i^2,\ldots,\boldsymbol{\mu}_i^s\}$.

step 3: $W_i = \{\ \}$.

step 4: **for all** $\boldsymbol{x}_i^j \in D_i^m$ **do**

step 5: 计算$\boldsymbol{x}_i^j$与各均值向量$\boldsymbol{\mu}_k$的距离$d_i^{jk} = L(\boldsymbol{x}_i^j, l_i^j, C_k, \boldsymbol{\mu}_k)\|\boldsymbol{x}_i^j - \boldsymbol{\mu}_k\|_2$.

step 6: 确定与$\boldsymbol{x}_i^j$距离最小簇的索引$I_k = \arg\min\limits_{1\leqslant k\leqslant s} d_i^{jk}$.

step 7: 将$\boldsymbol{x}_i^j$并入相应的簇$W_i^{I_k} = W_i^{I_k} \bigcup \{\boldsymbol{x}_i^j\}$.

step 8: **end for**

step 9: $\boldsymbol{\mu}_i' = \boldsymbol{\mu}_i$.

step 10: **for all** $\boldsymbol{\mu}_i^k \in \mu_i$ **do**

step 11: $\boldsymbol{\mu}_i^k = \dfrac{1}{|C_k|}\sum\limits_{\boldsymbol{x}_i^j \in C_k} \boldsymbol{x}_i^j$.

step 12: **end for**

step 13: **while**$\|\boldsymbol{\mu}_i' - \boldsymbol{\mu}_{\boldsymbol{i}}\| > a$.

2.2.2 共享统计数据

协方差矩阵与均值向量是表达数据集信息的两个重要数字特征. 数据集求解协方差矩阵与均值向量的过程可视为数据信息的压缩过程,如果将压缩后的信息作为节点间的共享数据,则同原始数据相比,可以大幅度降低节点间的通信量. 但是,压缩信息很难直接用于学习器的训练,因此需要在目标节点重构数据集. 由于任一数据集的协方差矩阵与均值向量的求解过程是不可逆的,基于协方差矩阵与均值向量重构原数据集是不可能的. 但是,如果重构的数据集与原数据集保持相同的协方差矩阵与均值向量,并且重构数据集的累计绝对误差控制在给定的阈值(保证算法学习正确性的最大阈值)范围内,则重构数据集在一定意义上更接近于原始数据集. 本文借鉴黎曼积分的思想,将无限容量的数据流依据在线到达的次序划分为等长的数据块,在满足一定累计误差的条件下,利用协方差矩阵与均值向量近似重构每个数据块. 由于每个数据块相对于无限容量的数据流,可以认为其足够小,进而通过对每个数据块的近似间接实现对总体的近似.

定理1 给定n个具有相同标签的示例:$\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n(\boldsymbol{x}_i = (x_{i1}, x_{i2}, \ldots, x_{ip})^{\mathrm{T}}, x_{ij} \in [a_{ij}, b_{ij}])$,$n$个相互独立且服从标准正态分布的随机向量:$\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_n(\boldsymbol{z}_i = (z_{i1}, z_{i2}, \ldots, z_{ip})^{\mathrm{T}}, z_{ij} \in [-u_{\alpha/2}, u_{\alpha/2}]$,$u_{\alpha/2}$为标准正态分布的上$\alpha/2$分位点),若$y_i = E(x_i) + Sz_i$,其中$S$满足$\Sigma = SS^{\mathrm{T}}$($\Sigma$为$n$个示例的协方差矩阵;$\lambda_j$、$\boldsymbol{q}_j = (q_{1j}, q_{2j}, \ldots, q_{pj})^{\mathrm{T}}$为$\Sigma$的第$j$个特征值及相应的特征向量),则对于任意给定的$\varepsilon_j > 0\,(j = 1, 2, \ldots, p)$,有

$$\Pr\Big(\Big|\sum_{i=1}^{n}(y_{ij} - x_{ij})\Big| < \varepsilon_j\Big) > 1 - 2\exp\Big(\frac{-2\varepsilon_j^2}{\sum\limits_{i=1}^{n}\Big(b_{ij} - a_{ij} + 2u_{\alpha/2}\sum\limits_{t=1}^{p}\lambda_t^{1/2}q_{tj}\Big)^2}\Big). \tag{5}$$

证明 设随机向量$\boldsymbol{X} = (X_1, X_2, \ldots, X_p), \boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n$为随机向量$\boldsymbol{X}$的$n$个观察示例. 令$\omega_j = (y_{1j} - x_{1j}, y_{2j} - x_{2j}, \ldots, y_{kj} - x_{kj}, y_{nj} - x_{nj}), \omega_j' = (y_{1j} - x_{1j}, y_{2j} - x_{2j}, \ldots, y_{kj}' - x_{kj}', y_{nj} - x_{nj})$,其中$x_{kj}' \neq x_{kj}$且$x_{kj}$、$x_{kj}' \in [a_{kj}, b_{kj}]$. 定义函数

$$f(\omega_j) = \sum_{i=1}^{n}(y_{ij} - x_{ij}), \tag{6}$$

则有

$$\begin{aligned}&|f(\omega_j) - f(\omega_j')| = \\ &\Big|\sum_{i=1}^{n}(y_{ij} - x_{ij}) - \\ &\Big(\sum_{i=1,i\neq k}^{n}(y_{ij} - x_{ij}) + (y_{kj}' - x_{kj}')\Big)\Big| = \\ &|(y_{kj} - x_{kj}) - (y_{kj}' - x_{kj}')|.\end{aligned} \tag{7}$$

由$y_{kj} = E(X_j) + S_j z_k$可得

$$\begin{aligned}&|f(\omega_j) - f(\omega_j')| = \\ &|E(X_j) + S_j z_k - x_{kj} - E(X_j') - S_j z_k' + x_{kj}'| = \\ &|(x_{kj} - x_{kj}') + S_j(z_k - z_k')| \leqslant \\ &|(x_{kj} - x_{kj}')| + |S_j(z_k - z_k')| \leqslant \\ &|(b_{kj} - a_{kj}) + 2u_{\alpha/2}S_j\boldsymbol{1}|.\end{aligned} \tag{8}$$

根据McDiarmid不等式,可得

$$\begin{aligned}&\Pr(|f(\omega_j) - E(f(\omega_j))| \geqslant \varepsilon_j) \leqslant \\ &2\exp\Big(\frac{-2\varepsilon_j^2}{\sum\limits_{i=1}^{n}(b_{ij} - a_{ij} + 2u_{\alpha/2}S_j\boldsymbol{1})^2}\Big).\end{aligned} \tag{9}$$

又由$|f(\omega_j) - E(f(\omega_j))| = \Big|\sum\limits_{i=1}^{n}(y_{ij} - x_{ij})\Big|, S_j\boldsymbol{1} = \sum\limits_{t=1}^{p}\lambda_t^{1/2}q_{tj}$可推出式(5). □

在上述定理成立的条件下,节点i只要将数据集的协方差矩阵集合与均值向量集合共享给节点j,并在节点j重新生成数据集,便间接实现了节点j共享节点i的原始数据信息. 接下来讨论如何完成这一过程.

假设节点i的最新到达的数据块为$D_i^k = \{\langle \boldsymbol{x}_i^1, l_i^1\rangle, \langle \boldsymbol{x}_i^2, l_i^2\rangle, \ldots, \langle \boldsymbol{x}_i^n, l_i^n\rangle\}$,且$D_i^k$被划分为$s$个簇: $C_1, C_2, \ldots, C_s$,其中C_i可视为一个$n_i \times p$的矩阵. 令Σ_{C_i}为C_i的协方差矩阵,则存在矩阵S_{C_i}使得$\Sigma_{C_i} = S_{C_i} S_{C_i}^{\mathrm{T}}$. 进一步,由定理1可得

$$\hat{C}_i = V_{C_i} + S_{C_i} Z. \tag{10}$$

其中: $V_{C_i} = \{\boldsymbol{\mu}_{C_i}, \boldsymbol{\mu}_{C_i}, \ldots, \boldsymbol{\mu}_{C_i}\}$, $\boldsymbol{\mu}_{C_i}$为C_i的均值向量, $Z = (\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_{n_i})$. 从式(10)很容易推得$\hat{C}_i$与$C_i$具有相同的协方差矩阵和均值向量. 因此,只需将节点i的数据块D_i^k的协方差矩阵集合与均值向量集合共享给节点j,在节点j便可以重新生成与数据块D_i^k具有相同信息结构的数据块$\hat{D}_i^k$.

2.2.3 节点间共享数据量分析

本文研究的分布式在线学习方式为异步学习方式,因此协方差矩阵与均值向量集合为节点之间的唯一共享数据,而无需其他的数据信息. 假设两个节点之间共享的协方差矩阵与均值向量集合为$M = \{M_1, \boldsymbol{v}_1, M_2, \boldsymbol{v}_2, \ldots, M_s, \boldsymbol{v}_s\}$. 其中: $M_i \in R^{p\times p}$, $\boldsymbol{v}_i \in R^p$. 如果M_i与$\boldsymbol{v}_i$中的每个分量占用r bytes,则M_i与$\boldsymbol{v}_i$共占用$r(p^2+2p)/2$ bytes,进而两个节点之间的通讯数据量为$sr(p^2+2p)/2$ bytes. 通常在某一个具体的学习任务中, s和p为一个常量,因此节点之间共享的数据量是一个确定的常量.

3 实验及结果分析

3.1 实验数据

本节实验选择1组人造数据集Hyperp和3组真实数据集Heterogeneity activity (Activity)、Forest cover type (Covertype)、KDDCUP1999 (Kdd)作为测试数据. 数据集Hyperp由数据生成器HyperPlane[27-28]构建,并在其中添加了渐进式概念漂移和反复出现的突发式概念漂移,而另外3组真实数据集同样具有明显的概念漂移特征,被广泛应用于数据流学习算法的检验[29-32]. 数据的详细信息请参见表1.

表1　数据集

数据集	维度	类型	大小	类别数量
Activity	7	numeric	13 062 475	7
Hyperp	10	numeric	1 000 000	2
Kdd	42	nominal, numeric	4 000 000	23
Covertype	55	numeric	581 012	7

3.2 对比方法

为了验证学习器在本文共享数据模式下的泛化学习有效性,本节实验选择原始数据共享模式(第1类共享方法)和微簇共享模式(第2类共享方法)作为参照对比方法. 第3类共享方法因学习器的不同而有所差异,同本文共享方法不属一类,不具有通用可比性,因此没有将其作为对比方法. 在每个局部节点,分别基于原始数据、微簇集合和协方差矩阵与均值向量集合构建3种集成学习器,记为: RDEL、MCEL和CVEL. 由于现有分类算法中只有最近邻算法[33]可以直接学习微簇,MCEL选择最近邻算法构建个体学习器. RDEL和CVEL的个体学习器的学习对象是与原始数据具有相同结构的数据集而非统计数据,因此构建个体学习器的分类算法的选择不受共享数据形式限制,现有的分类算法都可以作为候选. 为了构建泛化能力强和训练时间短的集成学习器,尽量选择预测精度高和训练时间短的分类算法作为个体学习器,但二者通常是矛盾的,因此在泛化能力与训练时间复杂度之间取得较好折中的算法是最佳选择. 本文使用多组数据集对几种常用的分类算法(包括KNN、BP神经网络、SVM、C4.5和朴素贝叶斯算法)进行测试,发现精度与训练和预测加权时间的比值最高为C4.5[34]算法,因此C4.5成为最后的选择.

由于每个节点都有可能出现概念漂移,需要周期性地更新节点学习器. 节点学习器由若干个个体学习器组成,而每个个体学习器所学到的假设空间是不同的,因此对当前数据块所表达概念类的覆盖程度也是不同的. 个体学习器在当前数据块的泛化错误率越高,其假设空间对当前数据块所表达概念类的覆盖程度越低;反之则越高. 鉴于此,节点学习器的更新策略为: 利用当前节点的最新已标记数据集计算每个个体学习器的泛化错误率,替换低于平均泛化错误率的个体学习器.

3.3 参数设置

本节实验的分布式环境由4个局部节点(distributed node)组成,每个节点部署一个本地学习器,一个测试集和一个已标记的初始训练集. 每次实验,将测试数据集随机划分为4个子集(交集为空)并分别存储在4个节点,每个节点的初始训练数据集由相应子集分层随机采样生成,训练集容量设置为1 000. 节点间的共享数据分为3种: 第1种为原始数据,其容量设置为2 000;第2种为协方差矩阵与均值向量集合,基于其重构的数据集容量通过式(5)计算得到,其中绝对误差界$\epsilon = n/100$ (n为重构数据集的容量),重构数据集与原始数据集累计绝对误差小于ϵ的概率下界$p_{\inf} = 0.75$;第3种为微簇集合,其容量为$30s$(s为共享数据集的类别标记数量,30为每个类别

的微簇数量). 为了更好地模拟数据随机到达的情况,每个局部节点的数据流速(每秒元组数)会随着时间不断变化,其变化范围设置为[200, 400].

RDEL、CVEL均设置16个个体学习器,其中每个个体学习器使用Bagging[35]方法训练得到. MCEL的构建借鉴了文献[16]中的方法,其个体学习器的数量等于训练集的类别标记数量乘以4,而其中每个个体学习器包含微簇的数量设置为30(注:原文将微簇数量设置为50,但是实验发现MCEL的预测速度较慢,于是对微簇数量进行了优化,发现微簇数量降至30时,平均泛化错误率几乎没有什么变化,所以将微簇数量设置为30),每个微簇包含的统计信息为微簇包含的样本数量n、微簇的均值向量μ、微簇中最远数据点与簇心的欧氏距离r.

3.4 实验结果与分析

本节实验选择了两种评价学习器泛化能力的度量指标:错误率(ER)和F_1. 其中:ER是分类错误的样本数占样本总数的比例;F_1是查准率与查全率的调和平均数,是查准率与查全率的综合度量指标.

$$F_1 = \frac{2 \times P \times R}{P + R}. \tag{11}$$

这里:P为查准率,R为查全率. 起始阶段,每个节点的学习器使用初始训练集构建. 从第1个到达节点的数据块开始,依次使用每个数据块对学习器进行测试,并记录测试结果,接着使用此数据块和其他节点的共享数据更新学习器. 本节每个图形的x轴代表数据流序列,而图2～图5、图6～图9、图10和图11的y轴分别代表ER、F_1、训练时间t和共享数据量(size of sharing data),其中每条曲线都是10次实验结果的平均值绘制的. 在图2～图9及图11中:虚线表示MCEL,实线表示RDEL,点线表示CVEL;横作标单位为chunk.

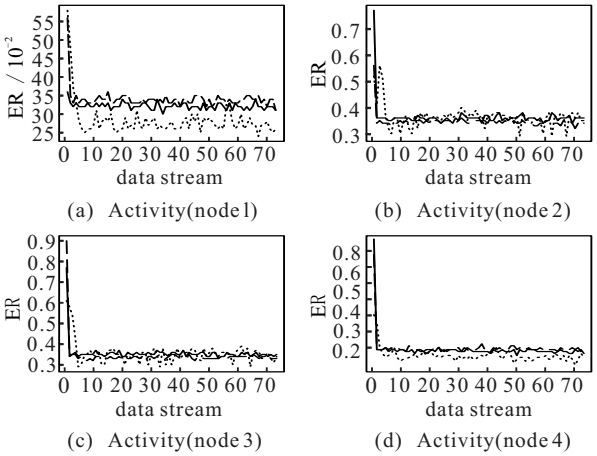

图2 所有节点数据流的ER(Activity)

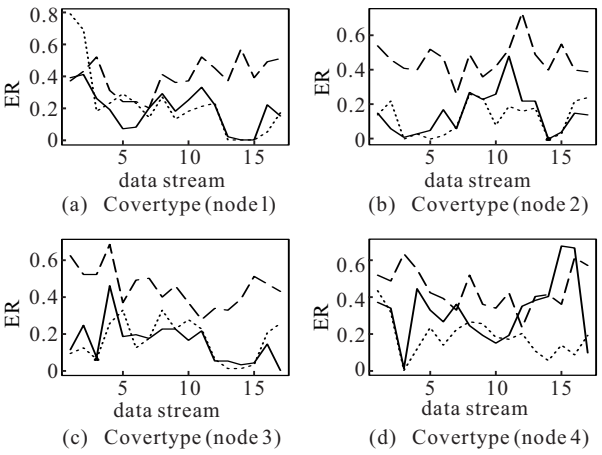

图3 所有节点数据流的ER(Covertype)

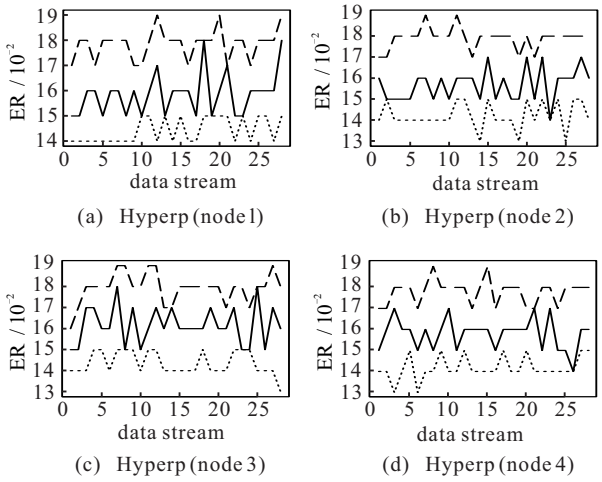

图4 所有节点数据流的ER(Hyperp)

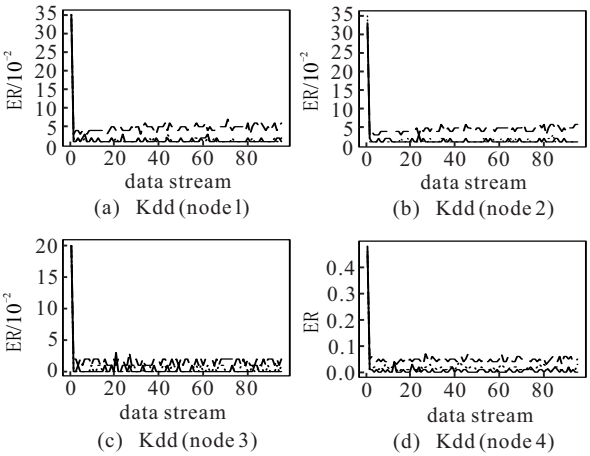

图5 所有节点数据流的ER(Kdd)

图2～图5显示了3种学习器在4个数据集上的ER测试结果. 观察Activity的测试结果可发现,从第15个数据块开始,CVEL在节点1和节点4的ER值要明显低于RDEL和MCEL,而在节点2和节点3,CVEL与MCEL的ER值相近,并略高于RDEL. 从Covertype的测试结果可发现,CVEL与RDEL在4个节点的ER值比较接近,并且低于MCEL. 观察Kdd的测试结果可发现,CVEL在前3个节点上的ER值与RDEL

接近,并低于MCEL,而在第4个节点上,RDEL的ER值低于CVEL. 从Hyperp的测试结果可发现,CVEL在4个节点的ER值与RDEL比较相近,并且略低于MCEL. 综合图2～图5的测试结果可以发现,CVEL与RDEL的ER值在绝大部分节点上是相近的,而且优于MCEL. 另外,3种学习器在Activity和Kdd的起始几个数据块出现了较高的ER,这主要因为初始已标记训练集是随机采样生成的,很可能其表达的概念类与数据块表达的概念类的交集很小,这将导致学习器学到的假设空间只能覆盖数据块表达的概念类的很小一部分,因此其ER值较高,但是随着已标记训练集的不断更新,学习器的ER值很快降低了.

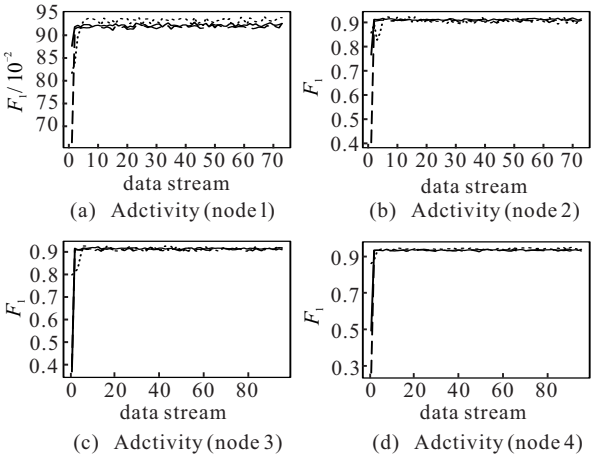

图 6　所有节点数据流的F_1 (Activity)

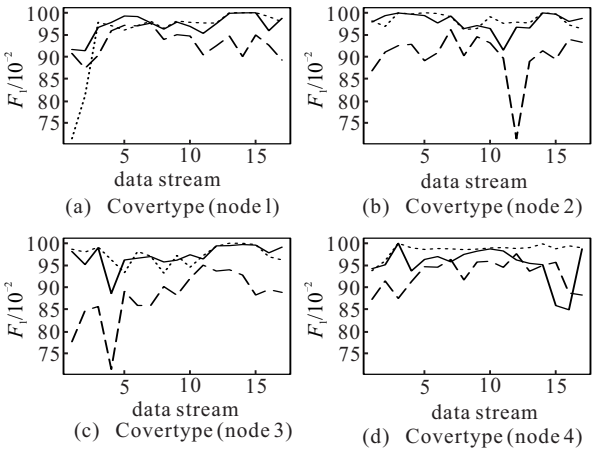

图 7　所有节点数据流的F_1 (Covertype)

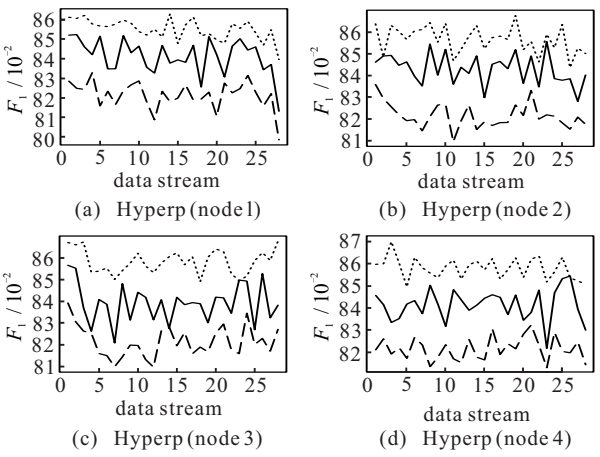

图 8　所有节点数据流的F_1 (Hyperp)

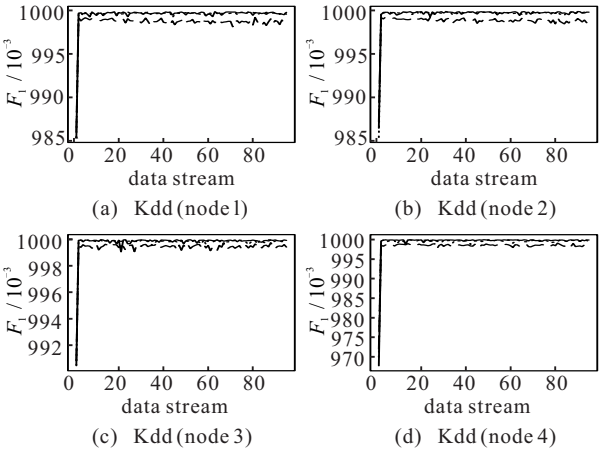

图 9　所有节点数据流的F_1 (Kdd)

图6～图9显示了3种学习器在4个数据集上的F_1测试结果. 观察Activity的测试结果可以发现,3种算法在后3个节点的F_1值比较接近,而在节点1,CVEL的F_1值略高于对比的两种算法. 从后3个数据集的测试结果可以发现,CVEL在4个节点的F_1值与RDEL相近,并高于MCEL. 另外,除了短暂的调整阶段(Activity的前13个数据块,Covertype的前3个数据块,Kdd的前2个数据块),CVEL与RDEL的F_1值平均变化幅度比较接近且小于MCEL. 综合图3的测试结果同样可以发现CVEL与RDEL的F_1值在绝大部分节点上是相近的,而且优于MCEL,这与图2的分析结果类似,进一步说明与MCEL相比,CVEL学到的假设空间更接近于RDEL.

在图10中:虚线表示MCEL,实线表示RDEL,点线表示CVEL,点划线表示CVELCP;单位为chunk.

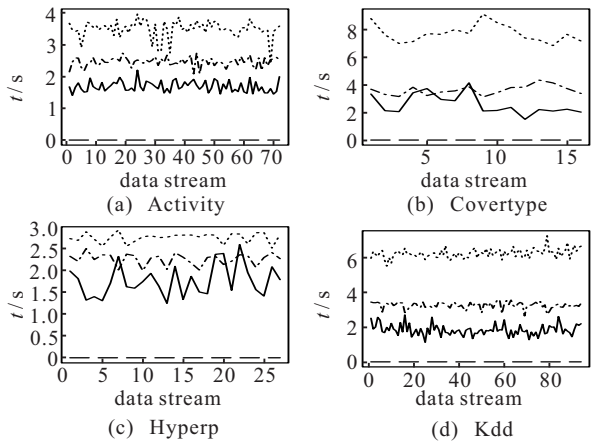

图 10　训练时间

图10显示了3种学习器在4个数据集上的训练时间测试结果. 由于MCEL的每个个体学习器使

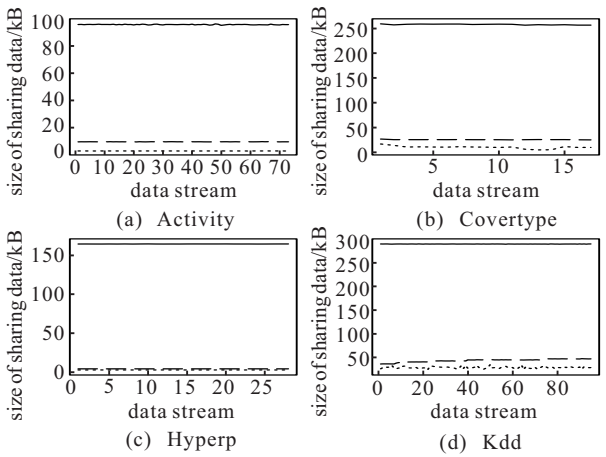

图 11 共享数据量

用最近邻算法构建,其训练时间开销为零. CVEL与RDEL使用了相同数量的个体学习器和同一训练策略,但最终CVEL的训练时间比RDEL高出一部分,这主要是因为RDEL的训练集即为共享的原始数据,可直接用于个体学习器的学习,而CVEL需要基于共享的协方差矩阵与均值向量集合重构训练集,才能进一步训练个体学习器,所以多出的时间开销为重构训练集的时间. 就二者总的训练时间差距而言,CVEL重构训练集的时间开销并不是很大. 另外,从训练集的重构过程可以看出,每次循环迭代的执行过程是相互独立的,所以本文引入多线程技术完成训练集的重构,并将基于多线程技术实现的CVEL记为CVEL(p). 观察图10可以发现CVEL(p)大幅缩短了训练集的重构时间,其训练时间平均缩短为原来的1/2,已经接近于RDEL,这说明多线程技术可以有效降低训练集的重构时间,促使训练集的重构进程对学习器训练更新的整个进程的影响降至最低.

现以Kdd数据集为例分析3种共享数据的传输数据量. 假设数据集中的每个分量占4 bytes,则原始数据集的共享数据量为$2\,000 \times 42 \times 4 = 336$ kB;微簇集合的共享数据量为$4 \times 30 \times (42 + 2) \times 23 = 121.4$ kB;协方差矩阵与均值向量集合的共享数据量为$4 \times 23 \times (41 \times 41/2 + 41) = 81.1$ kB. 从上述计算结果很容易看出协方差矩阵与均值向量集合的共享数据量最小,大约减少为原始数据集的6/25,微簇集合的2/3. 图11显示了3种共享方法在4个数据集的共享数据量测试结果. 从图11中可以发现,RDEL的共享数据量远高于CVEL和MCEL,而在Activity、Covertype和Kdd上,CVEL同MCEL存在较大差距,其共享数据量相比MCEL平均减少1/2. 另外,CVEL在Covertype和Kdd上的共享数据量出现了向下的波动,这主要是因为共享的协方差矩阵出现了较多的零元素,进而降低了整体的共享数据量.

表2列出了3种学习器在4个数据集上的统计结果. MCEL的泛化性能相对较差,其ER与F_1的平均值同RDEL和CVEL分别相差7.4%与2%和8.9%与2.6%,而CVEL与RDEL之间只有1.5%与0.6%的微弱差距. 另外,CVEL产生的共享数据量是最低的,其平均共享数据量约为RDEL的1/18,MCEL的1/2.

表 2 所有数据集的ER、F_1和共享数据量

数据集	MCEL			RDEL			CVEL		
	ER	F_1	size/kB	ER	F_1	size/kB	ER	F_1	size/kB
Activity	0.340 2	0.910 0	9.698 8	0.325 8	0.917 0	95.827 3	**0.315 6**	**0.919 7**	**2.960 5**
Covertype	0.441 7	0.913 7	25.943 9	0.205 8	0.969 3	258.417 1	**0.175 4**	**0.973 5**	**10.434 7**
Hyperp	0.178 7	0.821 2	3.662 1	0.158 8	0.840 8	163.767 5	**0.142 9**	**0.856 4**	**2.251 4**
Kdd	0.042 2	0.998 7	48.113 7	**0.011 6**	**0.999 6**	289.219 3	0.014 1	0.999 5	**29.065 8**
average	0.250 7	0.911 8	21.854 6	0.176 8	0.931 7	201.807 8	**0.162**	**0.937 3**	**11.178 8**

表2的统计数据说明CVEL与RDEL的泛化能力相近且具有最低的共享数据量,验证了协方差矩阵与均值向量作为共享数据的有效性.

综合上述实验结果分析得到CVEL与RDEL的泛化能力相当,即二者具有相似的假设空间. 由于CVEL与RDEL采用相同的集成学习方法和个体学习器算法,训练集是影响其假设空间的唯一因素. 然而,现在二者学到的假设空间存在较大交集,则二者的训练集所表达的概念类存在较大交集,说明基于协方差矩阵与均值向量重构的数据集在一定的误差范围内较好地还原了原始数据集的概念类. 另外,基于协方差矩阵与均值向量的共享数据量是最低的,因此可以有效降低节点间的数据传输量,缩短节点间的数据传输时间,进而加快节点学习器的训练更新进程,当概念漂移现象发生时,局部节点可以快速完成数据聚类、数据共享以及学习器更新的整个进程.

4 结 论

针对如何高效共享局部节点的数据建立全局学习器的关键问题,本文提出了一种面向分布式在线学习的数据共享方法. 设计了一种分布式学习框架,并在此框架下了提出了基于指数损失的半监督聚类算法和基于协方差矩阵与均值向量的共享数据模式,证明了重构数据集与原始数据集累计绝对误差小于给定绝对误差界的概率下界. 此共享模式不仅具有较低的网络通信量、计算量和内存开销,而且基于此共享模式构建的学习器可以异步学习与预测,在共享节点生成任意数量大小的训练集,因此可以灵活选择构建节点学习器的算法. 然而,如果数据集的维数与样本数满足条件$p \geqslant 2n/s - 2$(p为维数,n为样本数,s为类别数),则本文共享方法将失去优势,因为其转换后的协方差矩阵与均值向量集合存储的数据量将大于等于原始数据集. 因此,未来将进一步研究基于高维数据的共享数据模式及相应的在线学习方法.

参考文献(References)

[1] Domingos P, Hulten G. Mining high-speed data streams[C]. Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Boston: ACM Press, 2000: 71-80.

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

[3] Masud M M, Chen Q, Khan L, et al. Addressing concept-evolution in concept-drifting data streams[C]. IEEE International Conference on Data Mining. Sydney: IEEE Computer Society, 2010: 14-17.

[4] Minku White A P L L, Yao X. The impact of diversity on online ensemble learning in the presence of concept drift[J]. IEEE Transactions on Knowledge and Data Engineering, 2010, 22(5): 730-742.

[5] Masud M M, Chen Q, Khan L, et al. Classification and adaptive novel class detection of feature-evolving data streams[J]. IEEE Transactions on Knowledge and Data Engineering, 2013, 25(7): 1484-1497.

[6] Barddal J P, Gomes H M, Enembreck F, et al. A survey on feature drift adaptation: Definition, benchmark, challenges and future directions[J]. Journal of Systems and Software, 2017, 127(5): 278-294.

[7] Masud M M, Chen Q, Gao J, et al. Classification and novel class detection of data streams in a dynamic feature space[C]. European Conference on Machine Learning and Knowledge Discovery in Databases. Berlin: Springer-Verlag, 2010: 337-352.

[8] Wolff R, Bhaduri K, Kargupta H. A generic local algorithm for mining data streams in large distributed systems[J]. IEEE Transactions on Knowledge and Data Engineering, 2009, 21(4): 465-478.

[9] Tekin C, Yoon J, Van der Schaar M. Adaptive ensemble learning with confidence bounds[J]. IEEE Transactions on Signal Processing, 2017, 65(4): 888-903.

[10] Vanli N D, Sayin M O, Delibalta I, et al. Sequential nonlinear learning for distributed multiagent systems via extreme learning machines[J]. IEEE Transactions on Neural Networks and Learning Systems, 2017, 28(3): 546-558.

[11] Chen R, Sivakumar K, Kargupta H. Distributed web mining using bayesian networks from multiple data streams[C]. Proceedings of the 2011 IEEE International Conference on Data Mining. Washington: IEEE, 2002: 75-82.

[12] Ramirez-Gallego S, Krawczyk B, Garcia S, et al. Nearest neighbor classification for high-speed big data streams using spark[J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2017, 47(10): 2727-2739.

[13] Wang C K, Meng X F, Guo Q, et al. Automating characterization deployment in distributed data stream management systems[J]. IEEE Transactions on Knowledge and Data Engineering, 2017, 29(12): 2669-2681.

[14] Akbar A, Khan A, Carrez F, et al. Predictive analytics for complex IoT data streams[J]. IEEE Internet of Things Journal, 2017, 4(5): 1571-1582.

[15] Masud M M, Gao J, Khan L, et al. A practical approach to classify evolving data streams: Training with limited amount of labeled data[C]. The 8th IEEE International Conference on Data Mining. Pisa: IEEE, 2008: 929-924.

[16] AI-Khateeb T, Masud M M, Ai-Naami K M, et al. Recurring and novel class detection using class-based ensemble for evolving data stream[J]. IEEE Transactions on Knowledge and Data Engineering, 2016, 28(10): 2752-2764.

[17] Hahsler M, Bolaños M. Clustering data streams based on shared density between micro-clusters[J]. IEEE Transactions on Knowledge and Data Engineering, 2016, 28(6): 1449-1461.

[18] Fahy C, Yang S, Gongora M. Ant colony stream clustering: A fast density clustering algorithm for dynamic data streams[J]. IEEE Transactions on Cybernetics, 2019, 49(6): 2215-2228.

[19] Morales G D F, Bifet A. SAMOA: Scalable advanced massive online analysis[J]. Journal of Machine Learning Research, 2015, 16(1): 149-153.

[20] Basheer A, Sha K. Cluster-based quality-aware adaptive data compression for streaming data[J]. Journal of Data

and Information Quality, 2017, 9(1): 1-33.

[21] Fan W, Stolfo S J, Zhang J. The application of AdaBoost for distributed, scalable and on-line learning[C]. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM Press, 1999: 362-366.

[22] Chawla N V, Hall L O, Bowyer K W, et al. Learning ensembles from bites: A scalable and accurate approach[J]. Journal of Machine Learning Research, 2004, 5(4): 421-451.

[23] Folino G, Pizzuti C, Spezzano G. Training distributed GP ensemble with a selective algorithm based on clustering and pruning for pattern classification[J]. IEEE Transactions on Evolutionary Computation, 2008, 12(4): 458-468.

[24] Canzian L, Zhang Y, Van der Schaar M. Ensemble of distributed learners for online classification of dynamic data streams[J]. IEEE Transactions on Signal and Information Processing Over Networks, 2015, 1(3): 180-194.

[25] Shao J M, Huang F, Yang Q L, et al. Robust prototype-based learning on data streams[J]. IEEE Transactions on Knowledge and Data Engineering, 2018, 30(5): 978-991.

[26] Aloise D, Deshpande A, Hansen P, et al. NP-hardness of Euclidean sum-of-squares clustering[J]. Machine Learning, 2009, 75(2): 245-248.

[27] Street W N, Kim Y S. A streaming ensemble algorithm (SEA) for large-scale classification[C]. Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM Press, 2001: 377-382.

[28] 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, 2014, 25(1): 81-94.

[29] 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.

[30] Zhang P, Zhu X Q, Shi Y. Categorizing and mining concept drifting data streams[C]. ACM Sigkdd International Conference on Knowledge Discovery and Data Mining. Las Vegas: ACM Press, 2008: 812-820.

[31] Rutkowski L, Jaworski M, Pietruczuk L, et al. A new method for data stream mining based on the misclassification error[J]. IEEE Transactions on Neural Networks and Learning Systems, 2015, 26(5): 1048-1059.

[32] 张宇, 包研科, 邵良杉, 等. 面向分布式数据流大数据分类的多变量决策树[J]. 自动化学报, 2018, 44(6): 157-169.
(Zhang Y, Bao Y K, Shao L S, et al. A multivariate decision tree for big data classification oriented to distributed data streams[J]. Acta Automatica Sinica, 2018, 44(6): 157-169.)

[33] Cover T M, Hart P E. Nearest neighbor pattern classification[J]. IEEE Transactions on Information Theory, 1967, 13(1): 21-27.

[34] Quinlan J R. C4.5: Programs for machine learning[M]. San Mateo: Morgan Kaufmann Publishers Inc., 1992: 115-287.

[35] Breiman L. Bagging predictors[J]. Machine Learning, 1996, 24(2): 123-140.

作者简介

张宇(1981−), 男, 讲师, 从事机器学习、数据挖掘的研究, E-mail: 185629623@qq.com;

刘威(1977−), 男, 副教授, 博士, 从事机器学习、深度神经网络、复杂系统仿真等研究, E-mail: lv8218218@126.com;

邵良杉(1961−), 男, 教授, 博士, 从事数据挖掘、复杂管理信息系统等研究, E-mail: lntushao@163.com.

(责任编辑：闫 妍)