低代价的数据流分类算法①

李 南

(福建农林大学 计算机与信息学院, 福州 350002)

摘 要: 现有数据流分类算法大多使用有监督学习, 而标记高速数据流上的样本需要很大的代价, 因此缺乏实用性. 针对以上问题, 提出了一种低代价的数据流分类算法 2SDC. 新算法利用少量已标记类别的样本和大量未标记样本来训练和更新分类模型, 并且动态监测数据流上可能发生的概念漂移. 真实数据流上的实验表明, 2SDC算法不仅具有和当前有监督学习分类算法相当的分类精度, 并且能够自适应数据流上的概念漂移.

关键词: 概念漂移; 数据流; 分类; 低代价; 监督学习

Low-Cost Algorithm for Stream Data Classification

LI Nan

(College of Computer and Information Science, Fujian Agriculture and Forestry University, Fuzhou 350002, China)

Abstract: Existing classification algorithms for data stream are mainly based on supervised learning, while manual labeling instances arriving continuously at a high speed requires much effort. A low-cost learning algorithm for stream data classification named 2SDC is proposed to solve the problem mentioned above. With few labeled instances and a large number of unlabeled instances, 2SDC trains the classification model and then updates it. The proposed algorithm can also detect the potential concept drift of the data stream and adjust the classification model to the current concept. Experimental results show that the accuracy of 2SDC is comparable to that of state-of-the-art supervised algorithm.

Key words: concept drift; data stream; classification; low-cost; supervised learning

数据流分类现已成为数据挖掘领域的一个研究热点, 涉及的实际应用包括网络入侵检测以及垃圾邮件过滤等. 数据流上的样本持续、快速地到来, 使得传统的一次性静态学习的数据挖掘算法无法适用. 此外, 数据流上隐含的知识也有可能随着时间的推移而发生变化, 出现概念漂移[1]现象. 例如, 某些特定商品(如冷饮)的销量随着季节会呈现周期性的变化, 大家在网上关注的热点话题会不断变化等. 如何捕获数据流上的当前概念也为处理数据流分类问题带来了新的挑战.

目前对数据流进行分类, 单一分类模型和集成分类模型是学者们主要采用的两种方式. 单一分类模型首先在训练样本集合上建立初始分类模型, 然后为了拟合当前数据流样本的分布情况, 用随后到来的样本对现有模型进行增量式更新. 然而, 这种模型通常结构复杂, 并且需要频繁地对模型进行繁琐的更新. 在将数据流上的样本按照到来的时间划分为大小相同的数据块后, 集成分类模型用新数据块上建立的基分类器, 替换当前模型中分类性能较差或者已经过时的基分类器. 由于基分类器的训练速度通常比单一模型的更新速度快[2], 因此集成分类模型更适合对高速持续产生的数据流进行分类.

本文提出一种采用集成分类模型方式的低代价的数据流分类算法(Semi-supervised learning algorithm for Stream Data Classification, 简称 2SDC). 无论是采用单一分类模型还是集成分类模型, 现有绝大部分数据流上的分类算法重点只关注模型的分类效果, 进而假设所有待分类样本一旦被分类其类别信息即可立刻获得,

① 基金项目:福建省自然科学基金(2013J01216,2016J01280)

收稿时间:2016-03-29;收到修改稿时间:2016-06-01 [doi:10.15888/j.cnki.csa.005556]

然后马上利用这些类别信息对模型进行更新以最大限度地提高模型的分类精度, 这无疑人为忽略了标记数据流上样本所需的高昂代价. 本文的创新主要在于采用半监督学习的思想来降低更新现有模型所需要的有类别标记样本的使用量, 因此更符合实际应用中的要求. 此外, 算法主动检测概念漂移的发生, 这也加快了算法对数据流上当前概念的适应速度.

1 2SDC算法基分类器

现有绝大部分数据流上的分类算法使用全部样本的真实类别来拟合样本的当前分布情况, 而标记高速数据流上的所有样本需要很高的代价. 为了减少拟合数据当前分布所需要样本的数量, 2SDC 算法首先使用类似于半监督学习的聚类算法, 将给定数据块划分为若干个簇. 然后, 挑选中其中有代表性的簇, 保存每个簇的中心、半径、所在子空间以及相应的类别作为集成分类模型的基分类器, 进而用其来拟合数据流上当前样本的分布情况.

1.1 无监督学习的 K-means 聚类算法

设固定大小的数据块 D 由 N 条样本组成, 即 $D=\{(x_1,y_1),(x_2,y_2),...(x_N,y_N)\}$. 其中, $x_n=(x_{n1},x_{n2},...x_{nM})$ 表示由 M 维属性构成的第 n 条样本, $y_n\in\{0,1,2,...,L\}$ 表示 x_n 的类别. 如果 $y_n=0$, 表示该样本的类别未被标记.

要将数据块 D 中的样本划分为 K 个簇 $\{C_1,C_2,...,C_K\}$, 无监督的 K-means 聚类算法的目标是最小化所有样本到各自簇中心的距离之和, 即最小化目标函数: $O_{K-means}=\sum_{k=1}^{K}\sum_{x\in C_k}dis(x,c_k)$, 其中, c_k 表示第 k 个簇的中心, $dis(x,c_k)$ 表示样本 x 与 c_k 的相异度.

1.2 2SDC 算法基分类器构建

为了降低标记样本所需要的代价, 训练集中应该只有少部分样本已标记类别. 此时, 聚类的目标应是在最小化所有样本到各自簇中心的距离之和的同时, 使得已标记类别的来自同一类的样本尽可能的在相同的簇中, 即最小化目标函数:

$$O_{SK-means}=\sum_{k=1}^{K}\sum_{x\in C_k}dis(x,c_k)+\sum_{k=1}^{K}v_k imp_k \quad (1)$$

公式(1)中的 imp_k 表示第 k 个簇的“不纯净”程度, 算法中使用信息熵来衡量, 即 $imp_k=Ent_k=\sum_{l=1}^{L}(-p_l^k\log(p_l^k))$. p_l^k 表示第 k 个簇中属于第 l 类样本的先验概率, 即 $p_l^k=\frac{|C_k(l)|}{|C_k|}$, 其中 $|C_k|$ 表示被划分到第 k 个簇的样本个数, $|C_k(l)|$ 表示 C_k 中属于第 l 类的样本个数. 显然, 如果被划分到 C_k 中的已标记类别的样本均来自同一个类, 那么 $imp_k=0$, 此时 imp_k 最小. 相反, 若第 k 个簇中的已标记样本均匀地来自各个类别, 那么 $imp_k=\log(L)$, 此时 imp_k 最大.

公式(1)中的 v_k 表示第 k 个簇的权值. 为了平衡“簇间相异度”(公式(1)中的前半部分)和“簇内纯净度”(公式(1)中的后半部分)的影响, 取 $v_k=\frac{1}{\log(L)}\sum_{x\in C_k}dis(x,c_k)$. 综合以上考虑, 2SDC 算法中使用的聚类算法的目标函数为:

$$O_{SK-means}\sum_{k=1}^{K}\sum_{x\in C_k}dis(x,c_k)+\sum_{k=1}^{K}\sum_{x\in C_k}\frac{1}{\log(L)}dis(x,c_k)Ent_k \quad (2)$$

这样, 给定样本 x 以及簇中心 $\{c_1,c_2...,c_K\}$, 当前条件下的最优聚类为:

① 如果 x 是未标记类别的样本,

$$x\in C_k, if\forall j\in\{1,2,...,K\}, dis(x,c_k)\le dis(x,c_j) \quad (3)$$

② 如果 x 是已标记类别的样本,

$$x\in C_k, where\ \ c_k=\arg\min_k dis(x,c_k)\cdot(1+\frac{1}{\log(L)}Ent_k) \quad (4)$$

此外, 数据空间中往往存在与特定类别无关或者次要的属性[3]. 因此在聚类前, 需要先将样本投影到各个簇对应的子空间上, 然后再考虑其与各个簇中心的相异度. 这样也能减少聚类算法的迭代次数, 加快算法的收敛速度. 因此, 给定当前样本 x_i, x_i 和第 k 个簇的中心 c_k 的相异度 $dis(x_i,c_k)$ 应该采用加权的欧式距离来计算, 即 $dis(x_i,c_k)=\sum_{m=1}^{M}w_{km}(x_{im}-c_{km})$. 算法中使用一个对角矩阵 $W_k=\begin{pmatrix}w_{k1}&&&\\&w_{k2}&&\\&&\cdots&\\&&&w_{kM}\end{pmatrix}$ 表示簇 C_k 中各个属性的权重(即所在的子空间), 其满足 $\sum_{m=1}^{M}w_{km}=1$ 并且 $\forall m=1,2,...,M, w_{km}\ge 0$. 值得注意的是, 即使是为来自同一类别的样本所建立的聚类, 也有可能在不同的子空间上. 以文本数据流分类为例, 来自

“体育”类别的样本可能来自 “电子竞技”或者“传统项目”, 而这两个部分每个属性的权重显然应该是不一样的. 因此, 2SDC 算法中为每个簇独立地设置一个属性权重.

在开始第一次聚类前, 首先初始化 $w_{km}=\frac{1}{M}, k=1,2,...,K, m=1,2,...,M$. 在一次聚类后, 如果被分到簇 C_k 中的样本投影到属性 m 上越密集, 那么说明该属性对于簇 C_k 就越重要, 需要在下一次聚类中增大 w_{km} 的值. 因此, 在第 s 次迭代后, 设置 $\Delta w_{km}^{s}=\left(\sum_{j=1}^{M}\left(\frac{\sum_{x_i\in C_k}[(x_{im}-c_{km})+\delta]}{\sum_{x_i\in C_k}[(x_{ij}-c_{kj})+\delta]}\right)^2\right)^{-1}$, 其中 δ 是为了防止分母为 0 而设置的一个很小的数值, 取 $\delta=10^{-6}$. 这样, 第 $s+1$ 次聚类中, 簇 C_k 中属性 m 的权值 w_{km}^{s+1} 可以表示为:

$$w_{km}^{s+1}=w_{km}^{s}+\Delta w_{km}^{s} \tag{5}$$

值得注意的是, 由于 $\sum_{m=1}^{M}w_{km}=\sum_{m=1}^{M}w_{km}^{s}+\sum_{m=1}^{M}\Delta w_{km}^{s}=2$, 不满足限制条件 $\sum_{m=1}^{M}w_{km}=1$, 因此令 $w_{km}^{s+1}=\frac{1}{2}(w_{km}^{s}+\Delta w_{km}^{s})$.

此外, 如果各个簇的初始中心是随机选取的, 这会对算法最终的结果造成较大的影响. 因此, 为了保证模型分类效果, 2SDC 算法选取初始中心的方法为: 设为每个类别样本建立的簇的个数为 num (即选取 $K=num\cdot L$), 那么在数据块 D 中已标记的属于第 l 类的样本里, 依次选取 num 条最不相似的样本作为初始的簇中心. 综上所述, 2SDC 算法基分类器构建过程如算法 1.

算法 1: 基分类器构建

输 入: N 条训练样本(包含已标记类别样本和未标记类别样本), 簇的个数 K($K=num\cdot L$), 终止参数 $\varepsilon=10^{-6}$

输 出: K 个簇

Begin

Step1. 选取 K 条最不相似的已标记类别的样本作为各个簇的初始中心, 用矩阵 C 表示.

Step2. 初始化各个簇每个属性的权重, 用矩阵 W 表示.

Step3. 初始化迭代次数 $h=1$.

Step4. 根据公式(3)或公式(4), 依次将 N 条样本分配到各个簇中. $h++$.

Step5. 更新每个簇的中心.

Step6. 根据公式(5), 更新各个簇每个属性的权重.

Step7. 如果 $\left\|C^h-C^{h+1}\right\|\le\varepsilon$ 并且 $\left\|W^h-W^{h+1}\right\|\le\varepsilon$,算法停止. 否则, 随机重新排列 N 条样本的顺序, 转向 Step4.

End

如果每次划分都只是将当前样本划分到当前最优的聚类中, 那么聚类结果和样本的排列顺序紧密相关. 然而, 尝试所有的排列顺序以获得最优解是不现实的. 因此算法 1 中在每次聚类后, 重新排列 N 条样本的顺序, 以接近最优解. 文献[4]中证明这种方式是收敛的.

在将数据块 D 划分为 K 个簇后, 2SDC 算法将进行筛选, 选取其中有代表性的簇, 以四元组 $cluster=\{center,radius,weight,label\}$ 的形式保存下来, 用来拟合 D 上各类别样本的分布情况. 其中, $center$ 表示 $cluster$ 的中心, $radius$ 表示半径, $weight$ 表示所在的子空间, $label$ 表示类别. 具体过程如算法 2.

算法 2: 簇筛选算法

输 入: K 个簇, 簇保存阈值 $\varepsilon=0.05$, D 中有标记类别样本的比例 $labPer$

输 出: 2SDC 算法的一个基分类器 $Base\{cluster_1,cluster_2,...,cluster_{K'}\}$

Begin

Step1. 在被划分到 K 个簇的已标记样本中, 依次选取各个类别标记样本最多的簇, 加入集合 S.

Step2. 在剩下的 K-L 个簇中, 删除已标记各类别样本之和过少(小于 $\varepsilon\cdot labPer\cdot N$)的簇, 将剩下簇的加入 S.

Step3. 对于 S 中的簇, 构建相应的 $cluster$. 其中, 用 x_i 表示簇 C_k 中距离中心最远的样本, 那么保存簇 C_k 的中心 c_k 作为 $center_k$, 属性权重 W_k 作为 $weight_k$, 已标记类别样本里数量最多的类标签作为 $label_k$, $radius_k=dis(x_i,center_k)$.

End

2SDC 算法基于普遍的聚类假设[5]: “属于同一聚类的样本很可能具有同样的类别”来对未知样本进行

有效分类. 给定待分类样本 x 和一个基分类器 *Base*, 依次计算 x 和 *Base* 中所有 *cluster* 中心的加权距离. 如果 $dis(x,center_k) \le radius_k$ (即 x 被 $cluster_k$ 覆盖), 那么根据聚类假设, x 和建立 $cluster_k$ 的样本形成一个聚类, 因此将 x 分类为 $label_k$. 如果 x 被多个 *cluster* 覆盖, 那么选取其中类别数量最多的, 作为 x 类别的估计值. 如果 x 不被任何 *cluster* 覆盖或者数量最多的类别不是唯一的, 那么说明 x 是难处理样本, *Base* 不对其进行判断.

1.3 集成分类器构造

2SDC 算法集成分类器 E 由 p 个基分类器和一个仲裁分类器构成. 其中仲裁分类器使用增量朴素 Bayes 分类器. 给定待分类样本 x, 分类过程如算法 3.

算法 3: 2SDC 算法分类

输　入: 集成分类模型 E, 待分类样本 x

输　出: x 类别的估计值 y

Begin

Step1. 依次使用基分类器对 x 进行分类.

Step2. 设各基分类器对 x 类别的估计值分别为 $Y=\{y_1,y_2,...,y_p\}$. 如果 Y 中出现次数最多的类别不唯一, 或者各基分类器均不对 x 进行判断, 那么使用仲裁分类器对 x 进行分类. 否则, 返回 Y 中出现次数最多的类别.

End

每当新的数据块到来, 首先利用算法 3, 使用现有分类模型 E 对样本进行分类. 然后, 训练一个新的基分类器. 如果 E 中现有的基分类器个数不超过 p, 那么保存新的基分类器. 否则, 将原来基分类器中建立时间最早的替换出来. 最后, 用新数据块上的有标记类别的样本对仲裁分类器进行增量更新. 这样, 2SDC 算法具有对 *cluster* 覆盖的样本高分类精度的同时, 对不被任何 *cluster* 覆盖的样本的分类精度也有了保证.

2 概念漂移检测

现有集成分类算法大多没有概念漂移检测机制, 被动的使用新基分类器逐步替换过时基分类器的方式来适应概念漂移, 这会导致概念漂移发生后需要较长时间才能将过时的基分类器全部替换, 表现为模型的分类精度会出现较长时间的持续下降. 2SDC 算法采用概念漂移检测机制, 当检测到概念漂移发生时, 立刻抛弃现有整个已经过时的集成分类模型, 因此能够更快地适应数据流上的最新概念.

文献[6]认为概念漂移产生的来源于样本与其类别的联合概率随着时间或者环境的改变而发生变化. 无论是某类的先验概率发生变化、某类的类概率发生变化还是样本后验概率发生变化, 都会导致已经稳定的现有模型的分类精度出现较大范围的波动. 即当数据流保持稳定时, 分类模型对于有标记类别样本的分类精度应该保持在一个比较稳定的水平. 因此, 2SDC 算法使用一个高斯分布来拟合分类精度的分布情况. 设前 t 个数据块上有标记样本的平均分类精度为 μ, 方差为 σ. 如果 $t+1$ 个数据块上有标记样本的分类精度低于 $\mu-1.96\sigma$, 那么说明出现概念漂移, 需要删除当前的基分类器和仲裁分类器, 重建分类模型以适应数据流上的现有概率. 此外, 数据流上不可避免的新类别样本的出现也是一种特殊的概念漂移. 因此, 如果当前数据块上标记样本中出现新类别的样本, 那么同样也需要重建分类模型.

3 实验结果与分析

本节在真实的文本数据流上对 2SDC 算法的分类效果进行测试. 实验环境为 Intel(R) Core i5 2.5GHz CPU、4G RAM PC 机. 比较算法使用数据流上常用的对比算法(Streaming Ensemble Algorithm, 简称 SEA)[7]、比较具有代表性的加权集成分类算法(Aggregate Ensemble, 简称 AE)[8]、基于决策树和 Bayes 混合模型的集成分类算法(Weight Ensemble Classifier - Decision Tree and Bayes, 简称 WE-DTB)[9]以及基于半监督学习的数据流集成分类算法(Semi-Supervised Learning Based Ensemble Classifier, 简称 SEClass)[10]. 值得注意的是, 前三种对比算法使用的所有样本均是有类别标记的, 而 SEClass 算法和 2SDC 算法使用的样本是部分标记类别的, 其中有标记类别的样本占所有样本数量的 20%. 各种算法的参数分别参照对应文献中的设置, 数据块大小 500, 2SDC 算法中基分类器个数为 5, 为各类样本保存的簇个数为 3.

实验中使用的文本数据流来自 20-Newsgroups 数据集. 数据集的属性个数为 500, 类别数为 6, 样本的分布情况见表 1. 实验中将数据集划分为两个部分来模拟数据流多类别出现概念漂移的情况. 在前半部分, 只有来自 med, baseball, autos, motor 四个类别的样本;

在后半部分中, motor 类的样本消失, 并新出现了 space 和 politics 类的样本. 各种算法在 20-Newsgroups 数据集上的平均分类精度如表 2 所示. 由于 2SDC 算法基分类器初始中心选择的不同会对分类结果造成影响, 因此结果采用 10 次实验的平均值, 并且在表 2 中给出方差.

表 1 20-Newsgroups 中各类分布

Category	No. of Instance	Category	No. of Instance
med	1162	motor	600
baseball	450	space	562
autos	600	politics	562

表 2 各种算法在文本数据流上的平均分类精度

2SDC		AE	WE-DTB	SEA	SEClass
Average	Variance				
0.756	0.012	0.776	0.714	0.601	0.652

从表 2 中可以看出, 只使用少量有类别标记样本的 2SDC 算法分类效果优于使用全部有标记样本的 WE-DTB 算法和 SEA 算法, 达到和 AE 算法相当的水平, 并且优于基于半监督学习的 SEClass 算法. 虽然 2SDC 算法初始中心的不同会对分类结果造成一定的影响, 但是方差并不太大. 由于 AE 算法使用的所有样本都是有类别的, 而且算法在当前数据块上使用多种分类算法建立相应的多个分类器以构成集成分类模型, 因此分类精度是所有算法中最高的. 为了检测本文使用的概念漂移机制的有效性, 不同算法在测试数据集上每连续 500 条样本的分类精度如图 1 所示.

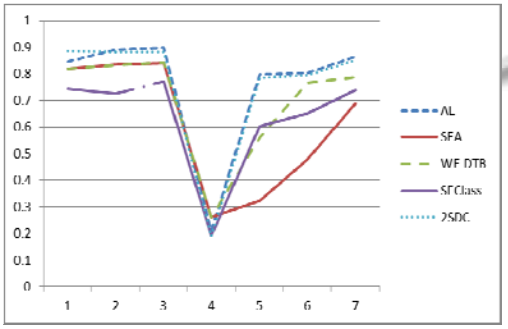

图 1 20-Newsgroups 上分类精度比较

从图 1 可以很清晰的看出, 在某一时刻, 由于新类别样本的出现以及原来类别样本的消失, 各种算法的精度均出现了不同程度的下降. 随着新类别样本的不断出现, 在一定时间后, 分类精度又都逐渐恢复到了之前的水平. 本文提出的 2SDC 算法的恢复速度较快, 这也证实了本文使用的概念漂移检测机制的有效性. 此外, 不同标记比例下算法的平均分类精度见图 2.

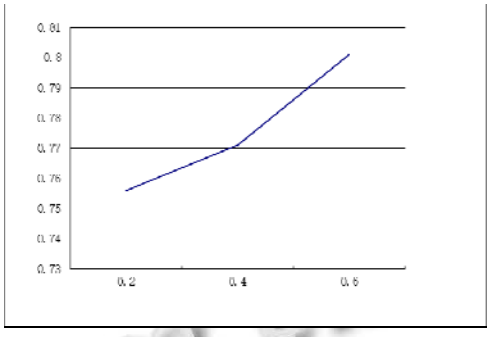

图 2 不同标记比例下平均分类精度

从图 2 中可以看出, 随着有标记类别样本数量的增多, 基分类器的边界更加准确, 也有更多的样本参与到仲裁分类器的更新, 因此 2SDC 算法的分类精度逐渐增高. 当标记样本比例较低时, 模型的分类精度还是稳定在一个相当的水平. 不同类别样本保存的簇个数下算法的平均分类精度见图 3.

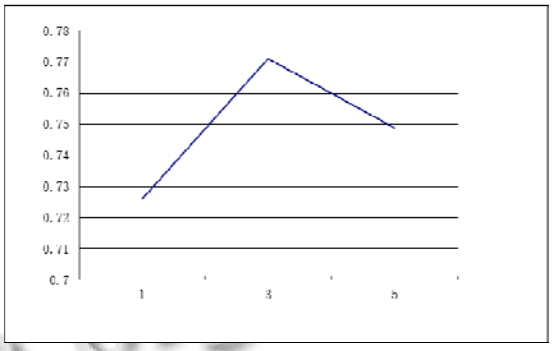

图 3 不同簇个数下平均分类精度比较

从图 3 可以看出, 当为每个类别样本保存 3 个簇时, 分类效果最佳. 当簇个数过低时, 基分类器无法准确拟合各类别样本的分布情况, 导致分类精度较低. 然而, 过高的簇个数降低了分类模型的泛化能力, 同样会降低模型的分类效果.

3 结语

本文提出了一种低代价的数据流分类算法 2SDC. 区别于使用所有待分类样本的真实类别来更新分类模型的现有大部分数据流分类算法, 新算法使用半监督学习的思想, 利用数据块上大量的未标记类别的样本, 通过动态调整权值, 为每个类别建立不同的簇来拟合各类别样本的分布情况. 仲裁分类器的引入也保证了

难处理样本的分类效果. 概念漂移检测机制的使用能够使得分类模型更快地适应数据流上的当前概念. 真实数据流上的实验证明了该算法的有效性. 下一步的工作方向是研究如何让算法自动调整各类别的初始簇个数.

参考文献

1 辛轶,郭躬德,陈黎飞,毕亚新.IKnnM-DHecoc:一种解决概念漂移问题的方法.计算机研究与发展,2011,48(4):592–601.

2 Turner K, Ghosh J. Error collection and error reduction in ensemble classifiers. Connenction Science, 1996, 18(3): 385–403.

3 Aggarwal CC, Procopiuc C, Wolf JL, et al. Fast algorithm for projected clustering. Proc. of the ACM-SIGMOD. New York. ACM Press. 1999. 61–71.

4 Masud MM, Woolam C, Gao J, et al. Facing the reality of data stream classification: coping with scarcity of labeled data. Knowledge and Information System, 2012, 33(1): 213–244.

5 Zhou D, Bosquet O, Lal T N. Learning with local and global consistency. Advances in Neural Information Processing Systems, 2003, 16(1): 321–328.

6 Keller JM, Hand D. The impact of changing populations on classifier performance. Proc. of the 5th International Conference on Knowledge Discovery and Data Mining. New York. ACM Press. 1999. 367–371.

7 Street WN, Kim YS. A streaming ensemble algorithm (SEA) for large-scale classification. Proc. of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York. ACM Press. 2001. 377–382.

8 Zhang P, Zhu X, Shi Y, et al. An aggregate ensemble for mining concept drifting data streams with noise. Proc. of the 13th Pacific-Asia Conference on Knowledge Discovery. Bangkok. 2009. 1021–1029.

9 桂林,张玉红,胡学刚.一种基于混合集成方法的数据流概念漂移检测算法.计算机科学,2012,39(1):152–155.

10 徐文华,覃征,常扬.基于半监督学习的数据流集成分类算法.模式识别与人工智能,2012,25(2):292–299.