基于子空间集成的概念漂移数据流分类算法①

李 南，郭躬德

(福建师范大学 数学与计算机科学学院，福州 350007)

摘 要：具有概念漂移的复杂结构数据流分类问题已成为数据挖掘领域研究的热点之一。提出了一种新颖的子空间分类算法，并采用层次结构将其构成集成分类器用于解决带概念漂移的数据流的分类问题。在将数据流划分为数据块后，在每个数据块上利用子空间分类算法建立若干个底层分类器，然后由这几个底层分类器组成集成分类模型的基分类器。同时，引入数理统计中的参数估计方法检测概念漂移，动态调整模型。实验结果表明：该子空间集成算法不但能够提高分类模型对复杂类别结构数据流的分类精度，而且还能够快速适应概念漂移的情况。

关键词：概念漂移；数据流；子空间；分类；集成

Classification Algorithm for Concept-Drifting Data Stream Based on Subspace Integration

LI Nan, GUO Gong-De

(School of Mathematics and Computer Science, Fujian Normal University, Fuzhou 350007, China)

Abstract: The classification of concept-drifting data streams with complex category structures has recently becomes one of the most popular topics in data mining. This paper proposes a novel subspace classification method, and uses it to form an ensemble classifier in a hierarchical structure for concept-drifting data streams classification. After dividing a given data stream into several data blocks, it uses the subspace classification method to train some bottom classifiers on each data block, and then uses these bottom classifiers to form a base classifier. The base classifers are used to build the ensemble classifier. Meanwhile, it introduces the parameter estimation method to detect concept drift. Experimental results show that the proposed method does not only significantly improve the classification performance on datasets with complex category structures, but also quickly adapts to the situation of concept drift.

Key words: concept drift; data stream; subspace; classification; integration

随着社会的发展，在网络安全、电子商务等众多应用领域每天都产生大量的数据流，这些数据流蕴含着丰富的有价值的知识有待挖掘。由于数据流具有快速、广域、持续等特点，使得传统的挖掘算法显得有些力不从心。同时，数据流中隐含的概念或知识可能会随着时间或环境的改变而发生变化，即 1996 年 Widmer 和 Kubat 提出的概念漂移问题[1]。

目前，处理数据流上概念漂移的方法大致有三种[2]：1)实例选择；2)实例加权；3)集成学习。Hansen HK 在文献[3]中证明，当学习模型的错误相对独立时，集成学习能取得较好的效果。Street 等[4]提出一个可用于数据流的概念漂移检测的集成分类器算法 SEA,展示了集成学习解决该问题的有效性。Wang 等[5]提出了一个集成学习的通用框架用于挖掘概念漂移数据流。他证实把数据流分成连续固定大小的数据块，并在这些数据块上建立集成分类器对发现概念漂移比较有效。此后，许多学者深入研究了集成分类器的权值设计和融合策略，包括文献[6,7]等。同时在解决概念漂移问题的时间复杂度和策略上也有许多进展，如文献[8,9]等。

然而，上述已存在的数据流分类模型都难以适应

① 基金项目:福建省高校产学合作重大项目资助(2010H6007);教育部留学回国人员基金(教外司留[2008]890 号)
收稿时间:2011-04-19;收到修改稿时间:2011-05-29

复杂的数据结构，容易受到维度效应的影响[10]。同时，真实数据往往具有复杂的结构（如类间存在重叠现象或者数据维度数较高）。以文档数据流为例，其不仅维度数高，不同的文档类别还经常出现共享某个相似主题的现象，在向量空间模型(VSM)中表现为类别之间的重叠。面对复杂结构的数据流，现有算法[4,6-9]的分类模型不仅构建分类模型耗时多，分类精度也很难得到保证。

本文提出了一种子空间集成算法 SIA(Subspace integrated algorithm)用于解决带概念漂移的数据流的分类问题,主要工作有以下几个方面：首先，将最近邻分类的思想用于分类模型中底层分类器的构建，提出一种线性时间复杂度的子空间分类算法，并采用层次结构将其用于构建集成分类器，同时给出期望错误分析。实验证明该分类模型在数据结构复杂时还能够保持较好的分类效果。其次，区别于传统的数据流上的集成分类算法，SIA 算法依据数理统计中(0-1)分布参数的区间估计来判定概念漂移。当检测到数据流产生概念漂移时，算法立刻抛弃已经过时的分类模型，从而提高分类模型对概念漂移的适应能力。

本文其他内容安排如下：第一节介绍 SIA 算法底层分类器的构建;第二节详细介绍 SIA 算法分类模型;第三节给出该方法在一些应用数据集上的实验数据和结果分析；最后，在第四节进行了总结并给出进一步的研究方向。

1 SIA底层分类器构建

本节首先介绍和 SIA 底层分类器构建有关的背景知识以及相关工作，然后给出模型簇的形式定义，再在此基础上对 SIA 算法底层分类器构建的算法进行描述，最后对算法进行分析。

1.1 背景知识以及相关工作

为了有效的避免“维度灾难”，同时减低训练和分类过程的时间复杂度，将子空间集成用于解决高维复杂数据分类问题受到了众多学者的亲睐[11]。但是，现有的子空间集成方法大都将所有的数据都投影到同一个子空间上[12,13]，显然无法体现数据的真实特性，使得分类精度受到影响。从直观上看，不仅不同的概念会处在不同的子空间中，同一个概念也会存在于不同的子空间中（比如入侵检测数据中 DoS 攻击这个概念，就可以分为 Smurf,Back, Neptune 等不同的子概念，他们存在于不同的子空间中）。

1.1.1 kNNModel 算法

最近邻分类是一种已经被广泛研究的有监督机器学习方法。经典的 k-最近邻(kNN)算法由于简单但颇有效被列为十大数据挖掘算法之一[14]。然而，其存在参数 k 难以确定和分类效率低的问题。

为了克服这些缺点，很多学者提出了多种基于最近邻思想的改进算法，如 Guo 等[15]提出的 kNNModel 算法。kNNModel 算法使用代表点集合建立最近邻分类模型，从而在学习过程中自动确定 k 的取值。算法的基本思想是：先以每个训练样本为中心向外扩展成一个区域，使这个区域覆盖最多同类点的同时不覆盖任何异类点；然后选择覆盖最多点的区域，以四元组<数据点的类别, 区域的半径, 该区域覆盖点的数量, 圆心>形式保存下来形成一个模型簇。算法重复迭代多次，直至所有的训练样本至少被一个模型簇所覆盖。这样，对于给定的测试样本，kNNModel 算法根据其落入的模型簇确定类别。

然而，kNNModel 算法存在计算复杂度较高的缺点，其训练时间复杂度为 $O(n^2)$。同时，当数据结构复杂时，其分类精度会受到影响。

1.1.2 FWKM 算法

FWKM 算法[16]是投影聚类[17]方法中的一种。它能够在高维空间挖掘隐藏在不同低维子空间中的簇类，具有对数据维度数目增长不敏感等优点。

FWKM 算法过程是：给定簇数目 K，在一个类 k-means 算法过程中对数据集进行划分的同时，根据维度权重大小与数据点投影到该维度上的分布离散程度成“反比”的思想，搜索每个划分所在的最佳投影子空间。它对于数据集的每个划分，给各维度赋予 [0,1] 区间的权值，用来表示维度与对应划分之间“模糊的”关联度。权重越大表示该维度与此划分的关联性越强。

1.2 模型簇的定义

给定 N 个样本组成的训练数据集 $X=\{(x_1,y_1),(x_2,y_2),...,(x_i,y_i)...,(x_n,y_n)\}$。其中 xi 表示第 i 个样本，是一个 D 维欧氏空间的向量，即 $x_i=\langle x_{i1},x_{i2},...,x_{iD}\rangle$；$y_i\in\{1,2,...,K\}$ 表示 xi 的类别标号, K（K>1）表示数据集中包含的类别数目。SIA 算法底层分类器训练过程的目标是从训练集 X 中学习得到底层分类模型 $\{p_1,p_2,...,p_l...,p_\alpha\}$。其中的每个元素 Pl 被称为一个模型簇，用于描述特定空间区域内的样本点。我

们将 P_l 用一个五元组表示。

定义 1 (模型簇). 模型簇

$$p_l=\left(Center_l,Weight_l,Class_l,\mathrm{Tolerance}_l,Radius_l\right)$$

其中：

① $Center_l=\{c_{l1},c_{l2},...,c_{lD}\}$。一个 D 维向量，表示该模型簇覆盖范围内所有样本点的中心，该“虚拟点”采用公式(1)计算。其中，Num(l)表示 Pl 范围内包含的样本点数。

$$Center_l=\frac{1}{Num(l)}\sum_{x\in P_l}x \tag{1}$$

② $Weight_l=\begin{pmatrix} w_{l1} & & & \\ & w_{l2} & & \\ & & \cdots & \\ & & & w_{lD} \end{pmatrix}$。一个 D 阶的对角矩阵，在高维数据的投影聚类中，$Weight_l$ 与一个模糊投影子空间相对应。矩阵的每一个元素表示该模型簇每个维度的权重。如果某个维度与该模型簇具有强相关性，则赋予较大的数值；而相关性较弱的维度，则赋予较小的数值。其中每个元素 wld 满足以下约束条件：$\forall d=1,2\cdots D: w_{ld}\ge 0;\sum_{d=1}^{D}w_{ld}=1$

③ $Class_l$：表示模型簇范围内该类别样本个数占绝大部分的样本的类别标号。

④ $Tolerance_l$：表示 P_l 中允许的样本数目占少数的异类样本的最大个数。与 kNNModel 算法不同，我们允许 Pl 范围内存在最多存在 Tolerancel 个异类样本以提高模型的泛化能力和抗噪能力。

⑤ $Radius_l$：表示模型簇 P_l 的半径。设 FHl(farthest hit)表示模型簇 Pl 范围内距中心最远的同类点，NMl(nearest miss)表示训练集 X 上距 Pl 中心第 β 近的异类点，那么 $Radius_l$ 采用公式(2)计算。其中 $dist\left(x_i,x_j\right)$，采用公式(3)计算。

$$Radius(l)=\begin{cases} dist(c_l,NM_l) & if\ \ dist(c_l,FM_l)>dist(c_l,NM_l) \\ \dfrac{dist(c_l,NM_l)+dist(c_l,FM_l)}{2} & otherwise \end{cases} \tag{2}$$

$$dist(x_i,x_j)=\sqrt{\sum_{d=1}^{D}w_{ld}(x_{id}-x_{jd})^2} \tag{3}$$

从直观上看，如果训练集上距簇中心第 β 近的异类点更接近簇中心，那么为了保证模型簇内只存在至多 $Tolerance_l$ 个异类点，模型簇的边界只能触及到 NM_l 点；否则可以将模型簇的边界放大到 FHl 和 NM_l 的中点。同时，为了检验样本 x_i 是否在模型簇 P_l 的范围内，需要首先将样本投影到 P_l 对应的投影子空间上，即使用加权欧氏距离函数来衡量 x_i 和 P_l 中心的相似度。

1.3 底层分类器训练算法

给定包含 K 个类别的训练集，训练过程调用 K 次训练算法 BaseTraining,为每个类别建立 3 个模型簇，这 3*K 个模型簇构成一个底层分类器。为每个类别选取 3 个模型簇是基于文献[18]的实验结果，对每类数据细化到 2-3 块时，学习的效果最佳。BaseTraining 算法的过程如下：

输入：训练集 TrainInstances,待生成分类模型的类标号 k(k=1,2,...,K)，容忍度 β

输出：3 个 Class 为 k 类的模型簇

Begin

1) 令 Instances={TrainInstances 中类标号为 k 的样本}；

2) 使用 FWKM 聚类算法，将 Instances 中的样本划分为 3 个样本子集。

3) 存储每个样本子集和特征权重矩阵 Wl。

4) 使用公式(1)和公式(2)以及容忍度 β，根据定义 1 构造每个样本子集相对应的模型簇，return。

End.

1.4 底层分类器分类算法

给定 N 个样本组成的测试数据集，算法调用 N 次分类算法 BaseTesting。算法过程如下：

输入：K 类样本的 3*K 个模型簇（即一个底层分类器），待分类样本 x_i

输出：x_i 的类别 y_i

Begin

1) 将 x_i 投影到每个模型簇所在的投影子空间上，即采用公式 3 计算 xi 与每个模型簇中心的距离 dist(xi, Pl)。如果 $dist(x_i, P_l)$小于该模型簇的半径（x_i 落入模型簇 Pl 的覆盖范围），那么将此模型簇的类标号存入类标号集合 S 中。

2) 当且仅当 S 中只包含一个类标号时，输出该类标号，结束分类。

3) 否则，分别计算 x_i 与 K 个类中同一类标号的 3 个模型簇中心距离之和，输出类标号为离 x_i 的距离之和最短的同一类标号的 3 个模型簇所属的类别。

End

1.5 底层分类器算法分析

本小节从以下三个方面对基分类器的构建方法进

行分析：

1) kNNModel 算法以欧式距离来衡量两个样本之间的相似度，虽然解决了传统 kNN 算法参数 k 难以确定的缺点，但是当类别之间存在重叠部分或样本维度较高时，分类精度容易受到影响。如图一所示，由 X,Y,Z 构成的三维特征空间中，两类训练样本(分别用实心椭圆、实心三角形表示)相互重叠，使用 kNNModel 算法在全空间中为这两类训练样本构造模型簇，必然导致模型簇数目的大量增加，使得分类模型的预测风险增大。若采用 BaseTraining 算法，将样本分别投影到两个不同的二维子空间{X,Y}和{X,Z}上(相应的投影点用空心椭圆和空心三角形表示)，则可以相对容易地为这些重叠的样本构造不同投影子空间上的模型簇，同时模型簇的预测能力也得到保证。图 1 中标出的椭圆样本就可以用两个绿色点代表的模型簇来表示，从而与三角形样本区分开来。

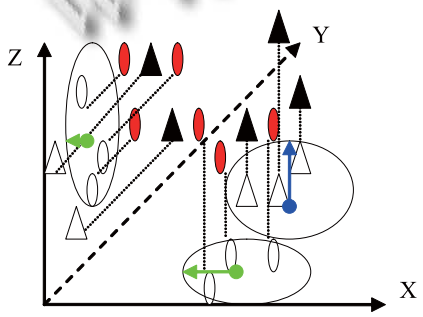

图 1 投影子空间模型簇的例子

2) 在 BaseTraining 算法中，我们允许每类模型簇存在一定数量的异类样本点以提高分类模型的泛化能力。同时，由于初始模型簇内距离簇中心最远的同类点可能是噪音，我们利用训练集上距中心第 β 近的异类点对模簇半径进行修正，这样也可以在一定程度上避免噪声对分类模型造成的影响。

3) 根据 BaseTraining 算法的过程中，若给定的数据集包含 K 个类别，算法时间复杂度为 $O(Kn)$。通常 K<<n，故相对于训练样本的数目 n，BaseTraining 算法具有线性的时间复杂度。同时，根据 BaseTesting 的过程，我们得到其时间复杂度为 $O(\alpha)$。

2 SIA算法

本节先介绍 SIA 算法分类模型并进行期望错误分析，然后描述 SIA 算法概念漂移检测机制，最后对 SIA 算法进行具体描述。SIA 算法使用滑动窗口模型,将数据流沿时间轴组织成固定大小Ψ的数据块序列，每个数据块用 D1,D2 ..., Dn 表示。

2.1 SIA 分类模型

SIA 分类模型采用层次结构进行分类器集成，可以看成是集成分类器的集成，其每个基分类器都是在各个数据块上利用 BaseTraining 算法采用类似 v 折交叉验证的方式建立的集成分类器。SIA 分类模型的框架如图 2 所示,SIA 分类模型中底层分类器建立方法如图 3 所示。

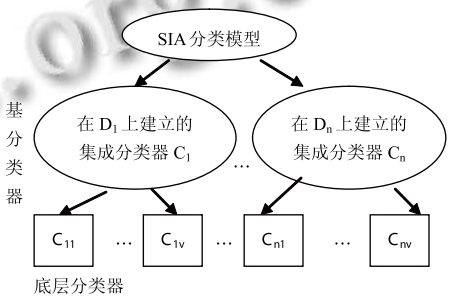

图 2 SIA 层次分类模型

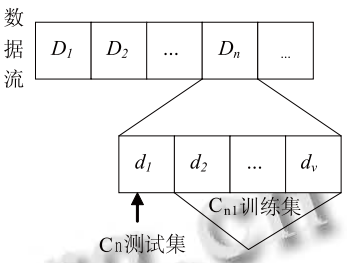

图 3 SIA 算法底层分类器建立方法

2.1.1 SIA 分类器期望错误分析

对于给定一个待分类数据 x，根据贝叶斯最优决策规则，如果 $p(y_i \mid x) > p(y_j \mid x)$,$i \neq j$,则将 x 分为 y_i 类。其中 $p(y_i \mid x)$ 为 x 为类别 y_i 的后验概率。根据文献[19]，分类器 $f(x)$ 输出 x 为 y_i 类的后验概率为:

$$f_{yi}(x) = p(y_i \mid x) + \eta_{yi}(x)$$

其中，η_{yi} 表示面对相同规模不同训练集，分类器的估计结果偏离平均估计结果的程度。假设一个二分类问题(即 y=“+”或者“-”)，分类器的附加错误可表示为：

$$R = \frac{\sigma^2_{\eta_{y+}} + \sigma^2_{\eta_{y-}}}{2S}$$

其中 $\sigma^2_{\eta_{y+}}$ 和 $\sigma^2_{\eta_{y-}}$ 分别表示 $\eta_{y+}(x)$ 和 $\eta_{y-}(x)$ 的方差，

S 是与数据集特性有关的一个常数。假设 $\eta_{y+}(x)$ 和 $\eta_{y-}(x)$ 独立同分布，那么

$\sigma^2_{\eta_{y+}}=\sigma^2_{\eta_{y-}}$，即 $R=\frac{\sigma^2_{\eta_y}}{S}$。

根据文献[20]，我们有 $\sigma^2_{EC}\le\frac{1}{rv}\sigma^2_s$：。其中 EC 表示在数据流最近 r 个数据块上建立的容量为 v 的集成分类器，S 表示用同样的分类器构建算法在最近的 r 个数据块上建立的一个单分类器。也就是说，SIA 算法基分类器建立算法就是 r=1 的特殊情况，即：

$\sigma^2_{base}\le\frac{1}{v}\sigma^2_{bottom}$。其中，base 表示在最近的一个数据块上建立的一个含有 v 个底层分类器的 SIA 算法的基分类器，bottom 表示在最近的一个数据块上建立的一个 SIA 算法的底层分类器。

由于 FWKM 算法采用了类 K-Means 算法的过程，初始点选择的差异会导致求得的同一概念数据投影子空间的差异。因此，SIA 算法采用类似 v 折交叉验证的方式，为同一个数据块建立 v 个底层分类器，并将其集成构成一个集成分类器（也就是 SIA 分类模型的基分类器）。假设在同一个数据块上数据具有相同的概念，那么在相应测试数据集上分类精度高的底层分类器更符合该数据块的概念，故我们采用每个底层分类器在各自测试数据集上的分类精度进行加权投票。同时，本小节也证明了这种基分类器建立方式的附加错误不会高于在同一数据块上建立的一个底层分类器。

2.1.2 SIA 分类模型基分类器算法描述

SIA 分类模型基分类器分类时，以其底层分类器在各自测试数据集上的分类精度为权重，采用最大投票策略对待分类样本进行分类。基分类器的训练及分类流程如下：

算法　MiddleTraining

输入：当前数据块 D_n，底层分类器容忍度 β，一个 SIA 分类模型基分类器中的底层分类器个数 v

输出：D_n 上的基分类器 C_n

Begin

1) 将 D_n 分成大小相等的 v 个部分{$d_1,d_2,...,d_v$}

2) for i=1 to v

a) 以 D_n-d_i 为训练集，根据底层分类器容忍度 β 采用 BaseTraining 算法训练底层分类器 C_{ni};;

b) 以 di 为测试集，采用 BaseTesting 算法得到 C_{ni} 在 d_i 上的分类精度，将其作为 C_{ni} 的权重。

3) end　for

4) 将 v 个底层分类器{$C_{n1},C_{n2},...,C_{nv}$}组成一个基分类器 C_n，其权重为这 v 个基分类器权重的平均值，return

End

一个 SIA 分类模型基分类器的分类流程如下：

算法　MiddleTesting

输入：一个 SIA 分类模型基分类器 Cn={$C_{n1},C_{n2},...,C_{nv}$}，待分类样本 x_i

输出: x_i 的类别 y_i

Begin

1) for i=1 to v

a) C_{ni} 采用 BaseTesting 算法判断 xi 的类别，并用 Cni 的权重参与投票

2) end for

3) 用最大投票策略决定 x_i 的类别，return

End

2.2 漂移检测

本文采用（0-1）分布参数的区间估计来进行漂移检测，这样避免了绝大部分集成分类算法[4,6,7]需要很长时间才能淘汰旧样本形成的基分类器，因而适应新概念慢的缺点。它的基本思想是认为小概率事件在一次实验中几乎是不可能发生，如果发生了，那么我们有理由怀疑这一假设的真实性。

当数据流稳定时，单个样本被错误分类的概率为 $P\{X=1\}=p$，正确分类的概率为 $P\{X=0\}=1-p$，即 X 服从两点分布 $B(1,P)$。设 1 个数据块有Ψ个样本，其中有 m 个被错误分类，那么 $\overline{X}=\frac{m}{\Psi}$。根据概率论和数理统计的相关知识，我们得到 p 的一个近似的置信水平为 1-α 的置信区间为：

$$\left[\frac{1}{2a}\left(-b-\sqrt{b^2-4ac}\right),\frac{1}{2a}\left(-b+\sqrt{b^2-4ac}\right)\right]$$

其中，$a=\Psi+Z^2_{\alpha/2}$，$b=-(2\Psi\overline{X}+Z^2_{\alpha/2})$，$c=\Psi\overline{X}^2$。

SIA 算法使用现有分类模型在上个数据块的分类精度，求得分类模型对原有概念误分类率 p 的置信水平为 0.95 的置信区间。如果现有分类模型在当前数据块上的误分类率不在此区间内，那么说明分类正确率出现了较大程度的变化，从而说明发生了概念漂移；反

之，我们认为概念分布平稳。

2.3 SIA 分类模型算法描述

先对算法中使用到的符号说明如下：D_n 表示数据流 S 中大小为 Ψ 的第 n 个数据块；EC 表示 SIA 集成分类器；EC_n 表示第 n 个数据块时的集成分类器；$|EC_n|$表示此时集成分类器中的基分类器个数；Num 表示分类模型中集成分类器容量；C_i 表示 EC 中的第 i 个基分类器；v 表示构成一个基分类器的底层分类器个数。

在构建新分类器之前，SIA 算法使用现有基分类器，利用加权投票的方式，采用最大投票策略依次分类数据块 D_n 上的各训练实例；然后以 D_n 为训练集使用 MiddleTraining 算法建立新的基分类器 C_n，以 C_n 中 v 个底层分类器$\{C_{n1},C_{n2},...,C_{nv}\}$的平均权值作为 Cn 的权值，同时以现有基分类器在 D_n 上的分类精度更新其权值；如果 EC 在 D_n 上的误分类率不在 D_{n-1} 的误分类率置信水平为 0.95 的置信区间内，那么我们认为发生概念漂移，抛弃 EC_{n-1} 中的所有基分类器，将 C_n 加入 EC_{n-1} 中构成 EC_n；当没有出现概念漂移时，如果 EC_n1 中基分类器的数量小于 Num 则直接将 Cn 加入 EC_{n-1} 中，否则从 C_n 和 EC_{n-1} 中选出 Num 个权值较大的基分类器组成 EC_n。具体过程描述如下：

算法：SIA

输入：集成分类器 EC_{n-1},当前数据块 D_n，底层分类器容忍度 β，集成分类器容量 Num,构成一个基分类器的底层分类器个数 v

输出：当前分类模型 EC_n

Begin

1) if EC_{n-1} 为空

a) 使用 MiddleTraining 算法，根据容忍度 β 在 Dn 上建立基分类器 Cn=$\{C_{n1},C_{n2},...,C_{nv}\}$，权重为这 v 个底层分类器权重的平均值。

b) 将 C_n 加入 EC_{n-1} 中组成 EC_n，return

2) end if

3) for i=1 to Ψ

a) for j=1 to | ECn-1|

b) 用 C_j 采用 MiddleTesting 算法预测实例的类别, 并用 Cj 的权重参与投票

c) end for

d) 用最大投票策略决定实例的类别

4) end for

5) 使用 MiddleTraining 算法, 根据容忍度 β 在 Dn 上建立基分类器 Cn=$\{C_{n1},C_{n2},...,C_{nv}\}$，权重为这 v 个底层分类器权重的平均值。

6) if EC 在 D_n 上的误分率不在 D_{n-1} 的误分类率置信水平为 0.95 的置信区间内

a) 删除 EC_{n-1} 中所有基分类器

b) 将 C_n 加入 EC_{n-1} 中组成 EC_n，return

7) end if

8) if | E D_{n-1}| < Num

a) 以 EC D_{n-1} 中基分类器在 Dn 上的分类精度为权重，更新 EC D_{n-1} 中基分类器的权重

b) 将 C_n 加入 E D_{n-1} 中组成 EC n，return

9) end if

10) if | E D_{n-1}| =Num

a) 以 E D_{n-1} 中基分类器在 D n 上的分类精度为权重，更新 E D_{n-1} 基分类器的权重

b) 从 E D_{n-1} 和 C_n 中选出权重较大的 Num 个基分类器组成 EC_n，return

11) end if

End

3 实验分析与讨论

为了评估 SIA 算法的性能，我们分别在真实数据集和实验数据集上对算法的精确度和适应性进行实验。实验环境如下：2.6GHz CPU 和 2G RAM；操作系统为 Windows ;开发环境为基于 JAVA 语言的 Weka 平台，编译运行环境为 jdk1.5.

3.1 使用的算法

为了验证本文提出的算法的有效性，对比算法使用经典的 SEA 算法[4]目前比较流行的实例加权集成分类器算法 EWAMDS[6]以及分类器动态集成的 DWM 算法[21]。实验中各种算法的具体参数设置分别参考文献[4]、[6]、[21]中的实验参数，SIA 算法的参数设置见表 1 所示。

表 1 SIA 参数设置

数据块大小 Ψ	基分类器中底层分类器个数 v	基分类器个数 Num	底层分类器容忍度 β
500	3	5	5

3.2 数据集

1)移动超平面(Hyperplane)[22]：一个 d 维超平面上

的样本 X 满足形式：$\sum_{i=1}^{d} a_i x_i = a_0$。在实验中，我们设 d 为 3，并且随机产生 3 个不同的权重集合。在一次测试中，我们产生三万条数据，其中蕴含 3 个概念，2 次漂移。其中每个概念含有一万条样本,并包含 5%的噪声样本。

2) 20-Newsgroups：一个常用的文本数据集，它是由 K. Lang 收集自 20 个不同新闻组的文档。出于效率考虑，本文所用的数据集是 20-Newsgroups 来自同一个新闻组的部分样本集合，一共分为 6 类：med,baseball,autos,motor,space,politics。其中随机抽取了 4498 条样本，各类分布情况见表 2 所示，每个样本包含 500 个特征属性。同时，这 6 个类中的文档会相互间共享某个共同的主题。例如，某篇文档以"奥巴马参观某医药工厂"为主题，那该文档就处于 politics 和 med 两个类的重叠区域，难以确定其类别。在经过 WEKA 软件自带的[0,1]标准化方法进行处理后，我们将数据划分为 2 大块。在第一大块数据中，只有 med,baseball,autos,motor 四个类；在第二大块数据中，添加了两个新的类别 space 和 politics，并淘汰了 motor 类的数据。由此，实验模拟一个多类别漂移的情况，以验证算法对真实复杂数据中出现新类问题的快速适应性以及对多类分类问题的处理能力。

表 2　20-Newsgroups 中各类分布

med	baseball	autos	motor	space	politics
1162	1162	450	600	562	562

3.3 实验结果分析

3.3.1 移动超平面数据集

各种算法在移动超平面数据集上的精度对比结果如图 4 所示，其中横坐标表示数据块序号，纵坐标表示分类正确率：

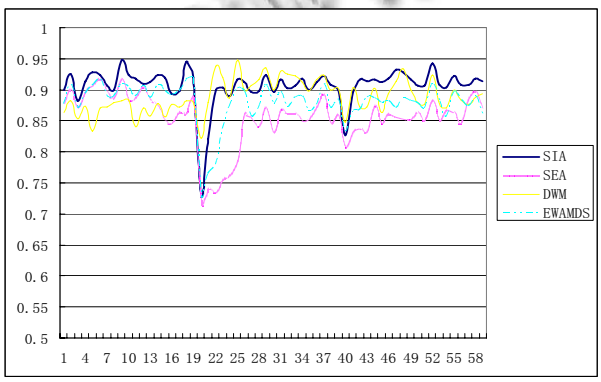

图 4　含 5%噪音的移动超平面数据集上分类精度比较

从图 4 可以看出，在第 20 和 40 个数据块时，由于数据发生概念漂移，各种算法的分类精度骤然下降。但是，随着数据的不断流入，分类器逐渐适应了新的概念，分类精度也恢复到了原先的水平。DWM 算法不受数据块大小影响，根据分类模型在当前数据上的分类结果动态的删除和新建基分类器，因而在精度突降后回复的速度最好。SIA 算法在发现概念漂移后，马上抛弃原有过时的基分类器，因而能够较早的达到较高的分类精度。然而，SEA 算法和 EWAMDS 算法需要一定的时间才能淘汰由旧样本训练形成的基分类器，从而在最大投票策略下整体的分类精度才能有所提高，故需要最长时间才能适应新的概念。同时，我们可以看出，在大部分情况下，SIA 算法在稳定后的分类精度方面明显的优于 SEA 算法、EWAMDS 算法以及 DWM 算法。

此外，各种算法在移动超平面数据集上的平均分类精度(各算法在每个数据块上的分类精度的平均值)见表 3 所示：

表 3　各种算法在移动超平面数据集上平均分类精度

算法	SIA	SEA	DWM	EWAMDS
精度%	90.6	85.5	88.9	87.9

3.3.2 20-Newsgroups 数据集

移动超平面是一个二分类问题，数据结构不够复杂而且没有出现新类到来的情况。为了验证 SIA 算法在真实复杂结构数据流中出现新类问题的快速适应性以及对多类分类问题的处理能力，我们在 20-Newsgroups 数据集上进行了测试，在这次测试中我们将数据块大小设置为 250，对比结果如图 5 所示，其中横坐标表示数据块序号，纵坐标表示分类正确率。

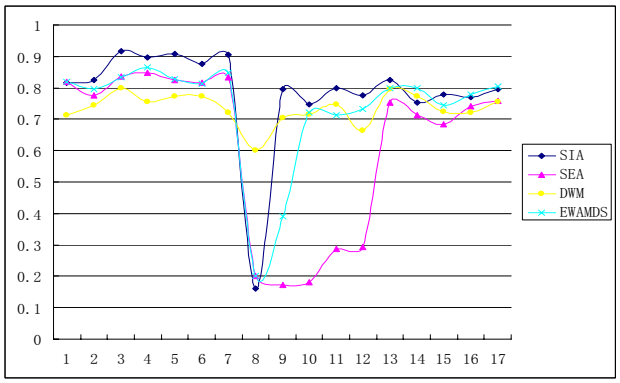

图 5　20-Newsgroups 数据集上的分类精度比较

从图 5 可以看出，在第 8 个数据块时，由于新类别数据的出现，各种算法的分类精度骤然下降。同各种算法在移动超平面数据集上的实验结果类似，DWM 算法最快的达到了较高的分类精度，其次是 SIA 算法，SEA 算法和 EWAMDS 算法适应新概念的速度最慢。由于将不同的类别投影到各自的子空间上，减少了类重叠部分对分类精度造成的影响，因此 SIA 算法在复杂的真实数据集上表现出了很好的分类能力。

此外，各种算法对 20-Newsgroups 数据集中每类数据的检测率见表 4 所示。

表 4 各种算法在 20-Newsgroups 数据集上每类的分类

精度%	SIA	SEA	DWM	EWAMDS
med	83.0	87.7	68.8	91.9
baseball	86.0	64.9	79.8	76.6
autos	66.2	49.8	73.8	61.6
motor	95.5	82.7	77.2	83.4
space	81.1	67.5	87.2	80.2
politics	84.6	67.0	87.9	76.0

从表 4 可以看出，对于 6 种类别的判断，SIA 算法都具有较高的分类精度。综合图 4 和表 4，我们可以得出结论：SIA 算法对复杂的真实数据集中出现的新类问题具有良好的适应性，对多类分类问题也有较好的处理能力。

3.4 平均分类精度测试

为了进一步观察 SIA 算法的性能，我们对于不同参数取值下算法的平均分类精度进行了测试，以下实验均采用含噪音 5%的移动超平面数据集。

1) 我们对在不同数据块大小 Ψ 下的 SIA 算法平均分类精度进行分析。其结果见表 5 所示：

表 5 不同数据块大小的平均分类精度比较

Ψ 值	100	250	500	750	1000
精度%	89.5	90.1	90.6	89.7	89.5

从表 5 可以看出，一开始，随着Ψ值的增大，由于有更多的训练数据，所以 SIA 算法的平均分类精度逐渐上升。然而，当Ψ值超过 500 以后，平均分类精度不再显著的增加，而是保持在一个相对稳定的水平。这可能是由于过大的数据块导致概念漂移发现的滞后，从而导致平均分类精度的略微下降。

2) 本次试验对在不同容忍度 β 下的 SIA 算法平均分类精度进行分析。其结果见表 6 所示：

表 6 不同容忍度的平均分类精度比较

β 值	0	3	5	10
精度%	90.3	90.5	90.6	89.8

从表 6 可以看出，在起始阶段，β 值的增大引起 SIA 算法平均分类精度升高。但是，当 β 值超过 5 时，继续增大的容忍度使得算法的平均分类精度下降。这可能与移动超平面数据集 5%的噪音有关，过大的容忍度导致过多的噪音数据加入分类模型中，从而对模型的分类精度造成负面影响。

3) 我们对在不同基分类器个数 Num 下的 SIA 算法平均分类精度进行分析。其结果见表 7 所示：

表 7 不同基分类器个数的平均分类精度比较

Num 值	3	5	7	10
精度%	89.8	90.6	90.4	90.5

从表 7 可以看出，较大的 Num 值使得 SIA 算法有着较高的分类精度。然而，当基分类器个数超过 5 时，算法的平均分类精度保持相对稳定。这种情况与文献[25]实验中测试的绝大部分集成分类器的算法相同。

4) 本次试验对在不同一个基分类器中底层分类器个数 v 下的 SIA 算法平均分类精度进行分析。其结果见表 8 所示：

表 8 不同基分类器中底层分类器个数的平均分类精度比较

v 值	2	3	5	8
精度%	89.4	90.6	90.3	90.4

从表 8 可以看出，较大的 v 值使得 SIA 算法保持较高的分类精度。然而，随着 v 值的继续增大，新建立的底层分类器之间的差异性逐渐减少，错误的独立性减低，因而整体集成分类器的性能不再提高，反而略微下降。

3 结语

本文将最近邻分类的思想运用于数据流分类，提出一种基于子空间集成的概念漂移数据流分类算法 SIA。它采用数理统计的相关知识判定概念漂移，将一种新的线性时间复杂度的子空间分类算法集成用于建立分类模型。新算法克服了绝大大部分集成分类算法适应数据流概念漂移缓慢和在结构复杂的数据流上分类精度不高的缺点。在移动超平面和 20-Newsgroups 数据集上的实验表明，与经典的 SEA 算法、当前比较

流行的 EWAMDS 算法和 DWM 算法相比，新算法具有很强的数据适应性以及较高的分类精度。尝试新的子空间集成方式是我们下一步的研究方向。

参考文献

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

2 Tsymbal A, Pechenizkiy M, Cunningham P, et al. Dynamic integration of classifiers for handling concept drift. Information Fusion, 2008,9(1):56−68.

3 Hanen LK, Salamon P. Neutral network ensemble. IEEE Trans. on Pattern Anaylsis and Machine Intelligence, 1990, 12(10):993 −1001.

4 Street W, Kim Y. A streaming ensemble algorithm (SEA) for large-scale classification. Proc. of 7th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining KDD-2001. New York: ACM Press, 2001:77−382.

5 Wang H, Fan W, Yu P, et al. Mining concept drifting data streams using ensemble classifiers. Proc. of 9th International Conference on Knowledge Discovery and Data Mining, Washington DC, 2003:226−235.

6 胡学刚,潘春香.基于实例加权方法的概念漂移问题研究.计算机工程与应用,2008,44(21):188−190.

7 孙岳,毛国君,刘旭,等.基于多分类器的数据流中的概念漂移挖掘.自动化学报,2008,34(1):93−97.

8 费洪晓,戴弋,穆珺,等.基于优化时间窗的用户兴趣漂移方法.计算机工程,2008,34(16):210−211.

9 富春岩,葛茂松.一种能够适应概念漂移变化的数据流分类方法.智能系统学报,2007,2(4):86−91.

10 Agrawal R, Gehrke J, Gunopulos, et al. Automatic subspace clustering of high dimensional data for data mining applications. Proc. of ACM SIGMOD Conference on Management of Data, New York: ACM Press, 1998.94−105.

11 欧吉顺,朱玉全,陈耿,于海平.基于动态加权的粗糙子空间集成.计算机工程,2010,36(22):178−180.

12 叶云龙,杨明.基于随机子空间的多分类器集成.南京师范大学学报(工程技术版),2008,8(4):87−90.

13 李敏,王勇,蔡立军.数据流分类中的增量特征选择算法.计算机应用,2010,30(9):2321−2323.

14 Yang Q, Wu X. 10 Challenging problems in data mining research. Journal of Information Technology and Decision Making, 2006,5(4):597−604.

15 Guo G, Wang H, Bell DA, et al. Using KNN Model for Automatic Text Categorization. Soft Computing. 2006, 10(5):423−430.

16 Huang JZ, Ng MK, Rong H, et al. Automated variable weighting in k-means type clustering. IEEE Trans. on Pattern Analysis and Machine Intelligence, 2005,27(5):657−668.

17 Aggarwal C, Procopiuc C, Wolf JL, et al. Fast algorithm for projected clusting. Proc. of the ACM-SIGMOD. New York: ACM Press, 1999:61−71.

18 辛轶,郭躬德,陈黎飞,黄杰.基于 KNN 模型的层次纠错输出编码算法.计算机应用,2009,29(11):3051−3055.

19 Tumer K, Ghosh J. Error correlation and error reduction in ensemble classifiers. Pattern Recognition,1996,29(2):341−348.

20 Mohammad M, Jing G, Latifur K, et al. Mining Concepr-Drifting Stream to Detect Peer to Peer Botnet Traffic. Proc. of the 4th Annual Workshop on Cyber Security and Information Intelligence Research (2008). New York: ACM Press, 2008:56−68.

21 Jeremy ZK, Marcus AM, Dynamic Weighted Majority:An Ensemble Method for Drifting Concepts. Journal of Machine Research, 2007,8(12):2755−2790.

22 Hulten G, Spencer L, Domingos P. Mining Time-Changing Data Streams. Proc. of ACM International Conference on Knowledge Discovery and Data Mining, New York: ACM Press, 2001:97−106.