第3章 基于数据流二重概念漂移检测的增量学习

3.1 引言

数据流的在线分类学习分为两部分，一是对数据流进行快速而准确的概念漂移检测，二是在检测到概念漂移发生后及时进行模型更新。由第 2 章可以知道，一般是把数据流划分为窗口的形式，通过检测当前数据窗口数据流的概率分布或分类结果来对概念漂移现象进行检测。

在概念漂移检测方面，本章将采用二重检测机制，即分别从数据属性和分类性能这两个角度对数据流进行检测。本章首先提出了基于 K-means 聚类和基于伯努利分布的二重概念漂移检测模型，该模型既考虑了数据属性，又结合了分类器的分类性能，有较强的抗概念漂移能力。数据流无限性的性质要求分类模型能够在保留旧知识的基础上连续性地学习新知识，而且，数据流经常发生概念漂移现象，这就要求分类器有较强的泛化能力。Lifna C S 等人[82]用在模型初始化时对主题进行按组分类且为每个组分配类别标签的方法识别概念漂移，使用滑动窗口的机制处理数据流。受这种思想的影响，一般认为首先对数据流进行聚类是一种比较好的模型构造方法。

本章内容主要分为 4 个部分：3.2 节介绍一种基于数据流的二重概念漂移检测机制；3.3 节介绍基于数据属性的 K-means 聚类（K-means Based on Data Property，KMBDP）二重概念漂移检测机制；3.4 节介绍基于分类性能的伯努利分布（Bernoulli Distribution Based on Classification Performance，BDBCP）二重概念漂移检测机制，在此基础上，引入增量支持向量机的概念；3.5 节给出一种基于增量 SVM 与二重检测的概念漂移数据流分类模型 TDD-ISVM。

3.2 二重概念漂移检测机制

第 1 章中把概念漂移的检测分为两个方面，即从可能引起概念漂移的数据属性与概念漂移发生后可能引起的分类结果这两个方面分析（数据属性方面和分类性能方面）。不妨称前者为性质法，后者为性能法。

无论是从数据属性方面还是从分类性能方面，不同的概念漂移检测方法都有不同的效果。性质法的优点是可以在对数据流进行分类之前通过数据的概率分布、特征关联等实例的属性来做概念漂移检测，从而指导分类器的构造；缺点是对某些类型的概念漂移检测不出，存在

漏报现象。性能法往往利用分类器分类后的结果作为判断有无概念漂移发生的依据，优点是可检测出多种类型的概念漂移，缺点是分类性能会受分类器的影响，分类性能反过来又影响概念漂移的检测。若把两者结合起来，则可以发挥各自的优势。

在基于性质法的概念漂移检测中，无论是通过数据概率分布还是通过属性相关性来对数据流进行检测，都是一个事倍功半的任务。正如前面所描述的，这需要系统对数据流的概率分布和属性相关性进行持续的监控，而实时的监控增大了系统的负荷。由于 K-means 相似性的概念要基于实例的属性来定义，即聚类的依据是实例的属性，因此，可以用 K-means 聚类来间接地反映数据属性的变化，即从数据属性方面对概念漂移进行检测。在分类性能方面，J. Gama 等人[83]介绍了一种基于伯努利分布的概念漂移检测方法。该方法认为，在一系列数据中，分类错误率是一个满足伯努利分布的随机变量。而且，该方法简单且效果好，不仅可以降低模型的复杂度，还可以提高模型的效率。基于此，可以利用伯努利分布对分类错误率进行评判，即从分类性能方面检测概念漂移。

本节从数据属性与分类性能两个维度介绍数据流的概念漂移检测机制，在两个维度上分别给出基于数据属性的 K-means 聚类数据流概念漂移检测机制和基于分类性能的伯努利分布数据流概念漂移检测机制，并详细描述两种机制的流程。

3.3　基于数据属性的二重概念漂移检测机制

数据的属性（特征）用来描述数据最小的单元，从数据的属性分析中可以了解数据的基本信息、学习数据背后的隐含信息。在很多对数据的处理方法中，数据的清洗、数据的分类学习等都是从数据的属性分析开始的。可见，认识数据属性是一种简单而有效的数据分析处理方式。

K-means 聚类不使用类别标签，而相似性的定义又是基于数据属性的，所以，K-means 是基于数据属性进行聚类的。把相似度高的数据实例划分在同一个簇内，把相似度低的数据实例划分在不同簇内，使得每个数据实例仅属于 K 个簇中的一个，以此达到对数据实例进行分类的效果。

K-means 聚类的概念漂移检测机制的算法伪代码如算法 1 所示。

算法 1　K-means 聚类的概念漂移检测机制的算法伪代码

输入： 聚簇中心代表集合 $C=\{(\text{center}_1,r_1),(\text{center}_2,r_2),\cdots,(\text{center}_k,r_k)\}$，测试集 $T=\{(X_1,y_1),(X_2,y_2),\cdots,(X_n,y_n),\cdots\}$，预先设定的阈值 cthd 和 uthd	
输出： 是否发生概念漂移	
1	val lines = ssc.socketTextStream();
2	for(i < |窗口实例个数|)
3	求得单点所属簇心，记录所有实例中心信息及其距离；
4	记录不可分实例，不可分实例个数 cnum++；
5	end for
6	if(cnum > cthd)
7	发生概念漂移
8	end if

```
9    利用均值计算除不可分实例外每个新簇的中心；
10   计算所有簇 Ck 的两两中心距离；
11   if(Set(Ci)&Set(Cj))
12       找出距每个簇中心最远的若干点，标记为不明实例，并记录其数目 unum；
13       if(unum > uthd)
14           k++，在新窗口上重新进行聚类；
15       else
16           将不明实例与不可分实例合并，构成新的不可分实例，发生概念漂移；
17       end if
18   end if
```

对于数据流中出现的概念漂移的现象，采取重新构造分类器的方法可解决新窗口数据分类精度下降的问题。由于数据流中出现的可能是噪声数据，因此可采用抛弃噪声数据的方法进行处理，从而在流式数据上进行概念漂移检测，需要把数据流以窗口的形式划分为一段段的批数据。当新窗口的数据到来时，根据 K-means 聚类的输出结果并按照聚类的定义，分析每个实例与 K 个聚簇中心的关系，为每个实例找到一个归属簇。当数据流中未发生概念漂移时，几乎可以为每个实例找到一个最近的簇。但当有概念漂移发生或出现噪声时，会导致两种情况发生：一是数据流中出现较多实例且不能被任何簇覆盖，即不能划分到任何一个已有的簇中，这种情况极可能是数据流发生了概念漂移；二是数据流中的某几个实例可以划分到某一个簇中，但划分的结果是将导致该簇与其他簇有交集，这种情况表明该实例极可能是噪声数据。

该算法会检测到两种异常现象：第一种是不能划分到某个簇中的实例较多，此时极可能是类别中出现了新的概念；第二种是某些实例可以划分到某一个簇中，但划分会改变该簇的结构，使得该簇与其他簇有交集，这种情况下，这些实例可能是噪声数据，也可能是新的概念。当这样的实例个数小于预先设定的阈值 uthd 时，则把其当成噪声数据抛弃；若大于 uthd，则很可能出现了新的概念，需要对新窗口数据重新进行聚类。

3.4 基于分类性能的二重概念漂移检测机制

3.3 节描述了基于 K-means 聚类的概念漂移检测机制，利用该算法可以快速地对数据流进行概念漂移检测，并能够识别一定的噪声。然而，基于 K-means 聚类的检测机制并不能检测出所有的概念漂移，该算法只对因先验概率 $P(X)$发生变化而导致的概念漂移或数据流中出现的新的概念有较好的检测效果。对于数据流中的另一类概念漂移，即对因数据的后验概率（或条件概率）$P(Y|X)$发生改变而导致的概念漂移，则检测不出。基于这种原因，本节继续研究概念漂移问题。基于 K-means 聚类的概念漂移检测是从数据属性方面进行的，作为对 3.3 节方法的补充，本节讨论基于分类性能的概念漂移检测，引入一种基于伯努利分布的概念漂移检测机制。

在由一系列实例组成的数据集中，若实例的分布保持稳定，则分类错误率会减小；若实例的分布发生了改变，则分类错误率会增大[85]。文献[83]指出，分类错误率是一个服从伯努利分布的随机变量，所以可以利用伯努利分布的性质来评判分类错误率的变化，以此来检测

概念漂移是否发生。

对于数据流中的某个窗口W_j，一共有 n 条实例，其中错误分类实例的个数为error_j，则该数据窗口的分类错误率和标准差分别为

$$p_j = \frac{\text{error}_j}{n} \tag{3.1}$$

$$s_j = \sqrt{\frac{p_j(1-p_j)}{n}} \tag{3.2}$$

基于伯努利分布的概念漂移检测算法要求对每个数据窗口的分类错误率和标准差进行计算，在当前窗口W_i的数据到来后，利用分类器对其进行分类后可计算其分类错误率 p_i，结合上一窗口的分类错误率 p_{i-1}和标准差s_{i-1}，可根据不等式（3.3）进行概念漂移检测

$$p_{i-1} + \alpha s_{i-1} \leqslant p_i \tag{3.3}$$

式中，α 为常数。若不等式（3.3）成立，则认为数据流中发生了概念漂移，这时，无论 K-means 聚类有没有检测到概念漂移现象的发生，均要对分类器进行增量式学习。反之，若不等式不成立，则认为数据稳定，未发生概念漂移。基于伯努利分布的概念漂移检测算法的伪代码如算法 2 所示。

算法 2　基于伯努利分布的概念漂移检测算法的伪代码

输入：分类错误率 p_i，上一窗口的分类错误率 p_{i-1} 和标准差 s_{i-1}，常数 α
输出：概念漂移是否发生
1　计算 $p_{i-1} + \alpha s_{i-1}$
2　if($p_{i-1} + \alpha s_{i-1} \leqslant p_i$)
3　　发生概念漂移
4　end if

基于伯努利分布的概念漂移检测算法非常简单，保证了流式数据的实时处理效果，因此在每次分类器进行分类后都可利用该方法做一次概念漂移检测。

3.5　基于增量 SVM 与二重检测的概念漂移数据流分类模型 TDD-ISVM

在常用的增量式学习算法中，本章选择增量支持向量机学习算法，并同 K-means 聚类进行有机融合。

3.4 节介绍了一种二重概念漂移检测机制，在增量支持向量机分类学习的基础上，利用 KMBDP 概念漂移检测机制和 BDBCP 概念漂移检测机制一前一后对数据流进行概念漂移检测。本节在基于数据属性的 K-means 聚类概念漂移检测算法和基于分类性能的伯努利分布概念漂移检测算法的基础上，引入增量学习的思想，基于支持向量机提出一种基于二重检测的概念漂移数据流分类模型 TDD-ISVM。

首先，在训练集$D=\{(X_1,y_1),(X_2,y_2),\cdots,(X_n,y_n)\}$上利用 K-means 聚类算法进行聚类，把聚类结果训练集划分为 K 个簇，并输出聚簇中心代表$C=\{(\text{center}_1,r_1),(\text{center}_2,r_2),\cdots,(\text{center}_k,r_k)\}$。

然后，利用 K-means 聚类的结果在测试集上进行概念漂移检测。在新窗口数据到来时，

利用数据属性的相似性为每个实例分配一个类别标签，当存在不可分实例时，存储该实例并记录不可分实例的个数 num；在为每个实例分配了类别标签后，排除不可分实例，利用均值重新计算新的聚类中心及其半径，若新的簇中心的半径覆盖另一个簇的半径，则依次找出该簇内相似度最小（距离最远）的实例，直到两个簇不再有交集时，把这些实例当成未标记实例。若不可分实例的个数没超过预先设定的阈值，则直接利用当前分类器对新窗口数据进行分类；若不可分实例的个数超过阈值，则很可能是新窗口数据出现了新的类别。原有的支持向量集 SV 决定的分类器可以对新窗口数据中的可分实例进行正确分类，而不可分实例的增多会导致分类效果变差。利用增量学习的思想，把不可分实例作为增量与当前支持向量集合并，重新学习新的分类模型，利用更新后的分类器对新窗口数据进行分类。

最后，利用伯努利分布再次检测数据流中有无概念漂移发生。在利用 K-means 聚类检测到数据流中发生概念漂移并更新了支持向量机分类模型后，若分类精度没有较大变化，则认为数据流中没有发生其他类型的概念漂移，当前分类器的分类效果好，把检测到的未标记实例当成噪声数据进行抛弃处理。若更新分类器后的分类精度依然不高，则可能是数据流中发生了其他类型的概念漂移现象，如因后验概率 $P(Y|X)$发生变化而导致的概念漂移。发生这种情况时，要把新窗口的全部数据当成增量与模型更新之前的支持向量集合并进行增量学习。利用更新后的分类器对新窗口数据再次进行分类，并保存当前分类器的支持向量集。概念漂移数据流分类模型 TDD-ISVM 算法伪代码如算法 3 所示。

算法 3　概念漂移数据流分类模型 TDD-ISVM 算法伪代码

输入：训练集 $D=\{(X_1,y_1),(X_2,y_2),\cdots,(X_n,y_n),\cdots\}$，聚类中心个数 k，阈值 cthd 和 uthd，常数 α，窗口长度 sw，滑动频率 s，迭代次数 knum、snum

输出：数据流实例的类别

```
1   val ssc = new StreamingContext(sc, Seconds(20))；
2   val clusters = KMeans.train(parsedData, k, knum)；
3   val model = SVMWithSGD.train(parsedData, snum)；
4   记录 K-means 聚类的聚簇中心代表 C 和 SVM 的支持向量集 SV；
5   把新到的数据流作为训练集，按照窗口长度 sw 来划分数据流；
6   对当前窗口 W_i 运用 KMBDP 算法进行概念漂移检测；
7   if(发生概念漂移)
8       重新聚类，更新 SVM 分类器并进行分类，利用 BDBCP 进行概念漂移检测；
9       if(发生概念漂移)
10          重新构造训练集并更新 SVM 分类器，重新分类并输出类别；
11      else
12          保留原 SV，抛弃噪声数据，输出类别；
13      end if
14  else
15      利用 SVM 分类器进行分类，利用 BDBCP 进行概念漂移检测；
16      if(发生概念漂移)
17          更新 SVM 分类器，重新分类并输出类别；
18      else
19          保留原 SV，输出类别；
```

```
20            end if
21    end if
```

TDD-ISVM 的算法流程图如图 3.1 所示。

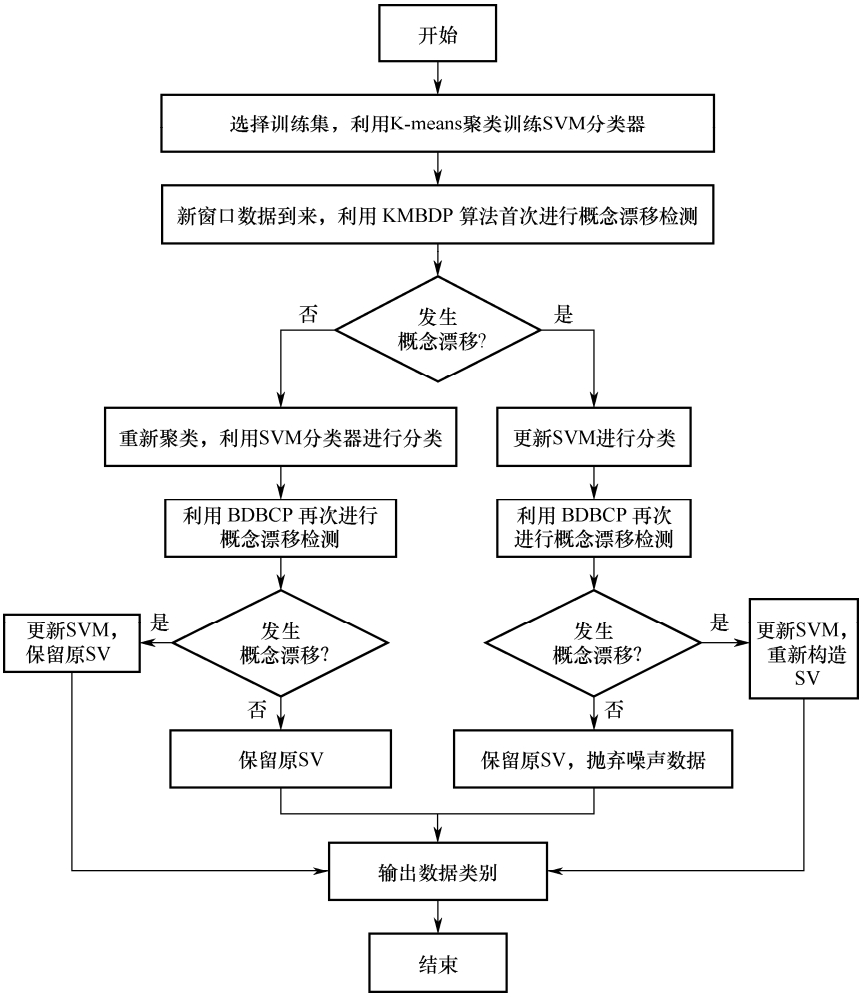

图 3.1　TDD-ISVM 的算法流程图

基于二重检测的概念漂移数据流分类增量 SVM 算法对流式数据进行两次概念漂移检测，可得到三种不同的情况，根据情况的不同再选择是否利用增量学习更新分类器。两次检测、多次更新的模式有助于提高分类器的分类精度，提高分类器的抗概念漂移能力。

3.6　TDD-ISVM 算法的时间复杂度分析

TDD-ISVM 算法的时间复杂度主要包括三个方面的内容：第一部分是基于 K-means 聚类的概念漂移检测；第二部分是基于增量学习思想的支持向量机模型的构造；第三部分是基于伯努利分布的概念漂移检测。在 K-means 聚类中，假设训练集的个数为 n，聚簇类别的个数为 k，迭代次数为 t，新窗口数据的长度为 m，则第一部分的时间复杂度为 $O(nkt+mk)$；在学

习支持向量机分类器模型中，假设训练集的个数为 n，数据的特征维度为 d，更新分类器时支持向量的个数为 N_{sv}，则第二部分的时间复杂度为 $O(dn^2+dN_{sv}^2)$；第三部分的时间复杂度为 $O(1)$。所以，TDD-ISVM 算法的总时间复杂度为三部分时间复杂度之和，即 $O(nkt+mk)+O(dn^2+dN_{sv}^2)+O(1)$。

3.7 实验设计与结果分析

3.7.1 实验数据集

本实验共采用三种数据集来验证算法的有效性，包括两种人工数据集，即 SEA 模拟数据集[87]和移动超平面数据集（Hyperplane）[88]，以及一种真实数据集——KDD Cup99 网络入侵数据集[89]。

1．人工数据集

（1）2001 年，Street 提出了著名的 SEA 模拟数据集，其数据形式为：空间表现形式为三维空间的随机点，所有属性的取值范围是 0～10，f_1 和 f_2 是两个仅有的相关属性。该数据集共包括 4 个概念，发生 3 次概念漂移。数据集的概念方程为 $f_1+f_2 \leqslant \theta$，其中 θ 为概念漂移划分阈值，有 4 个不同的取值，分别为 9、8、7、9.5，可以用来表示数据的 4 种概念。本实验选择 SEA 模拟数据集的大小为 150k，数据集中有 10%的噪声数据。实验中，把 100k 数据划分为训练集，剩余的作为测试集。

（2）移动超平面数据集（Hyperplane）是一个常用来模拟概念漂移数据流的数据集。在 d 维空间上，一个移动超平面定义为 $\sum_{i=1}^{d} a_i x_i = a_0$，把满足不等式 $\sum_{i=1}^{d} a_i x_i \geqslant a_0$ 的点标记为正实例，把满足不等式 $\sum_{i=1}^{d} a_i x_i < a_0$ 的点标记为负实例。其中，实例 $x=(x_1,x_2,\cdots,x_d)$ 从 $[0,1]^d$ 中随机产生，a_i 为权重，其取值是[0,1]范围内的随机数，同时有 $a_0=\frac{1}{2}\sum_{i=1}^{d} a_i$。在本实验中分别产生 300k 的包含 5%噪声的数据和 300k 不包括噪声的数据，其中，平均每 5k 数据发生一次概念漂移。

2．真实数据集

KDD Cup99 网络入侵数据集原是用于对网络入侵检测系统进行性能评估的数据集，也是常用于数据流学习的数据集。该数据集中有 41 维属性、24 种类别标签，包含一系列 TCP 访问记录。

3.7.2 实验结果与分析

本节将给出具体的实验结果来验证本章所提算法的效果，并对结果进行性能分析。具体将从分类精度、概念漂移检测、抗噪性三个方面进行验证。在分类精度方面，把本章所提算法 TDD-ISVM 与其他基于增量支持向量机的两种算法 IDSCBUC、IDSCBES 进行对比，其中，IDSCBUC 是较新的基于增量支持向量机的算法模型，IDSCBES 是传统的基于错误样本的增量支持向量机算法模型；另外，比较 TDD-ISVM 算法与 IBPNN-CDCA 算法、VFDTC 算法在三种数据集上的精度，其中，IBPNN-CDCA 算法为增量式 BP 神经网络的算法，VFDTC 算法

为传统的基于实例选择构造的算法。前面算法的对比，是在增量支持向量机基础上的对比；后面算法的对比，是与其他模型的比较，且都选择了一种时间较近的方法和一种传统的方法，在实验对比方面比较有说服力。在概念漂移检测方面，比较 TDD-ISVM、IBPNN-CDCA 和 VFDTC 三种算法对概念漂移检测的效果。在抗噪性方面，利用含噪声和不含噪声的两种移动超平面数据集（Hyperplane）来检测算法的抗噪性。

1）分类精度方面

该实验选择 150k 不包含噪声的移动超平面数据集作为训练集和测试集，在该数据集中，平均每 5k 数据发生一次概念漂移，分别验证 TDD-ISVM、IDSCBUC 和 IDSCBES 这三种算法的分类精度。

从图 3.2 所示的无噪声移动超平面数据集三种算法的分类精度可以观察到，TDD-ISVM 算法在分类精度方面明显高于另外两种算法，并且，在概念漂移比较严重的数据窗口上的分类精度并未大幅减小，具有稳定性。有两个原因：其一是在对数据流进行分类时，本章提出的 TDD-ISVM 算法充分考虑了数据属性和分类性能的变化，从两个方面对数据流进行概念漂移检测，使得在增量学习时能够及时而准确地更新分类器，使分类器具有较高的分类精度和泛化能力，保证了其稳定性；其二是在增量学习的过程中，TDD-ISVM 算法首先利用 K-means 聚类的结果，把不能正确分类的实例作为增量与当前支持向量集 SV 结合，从而更新分类器，增强了分类器的分类性能。

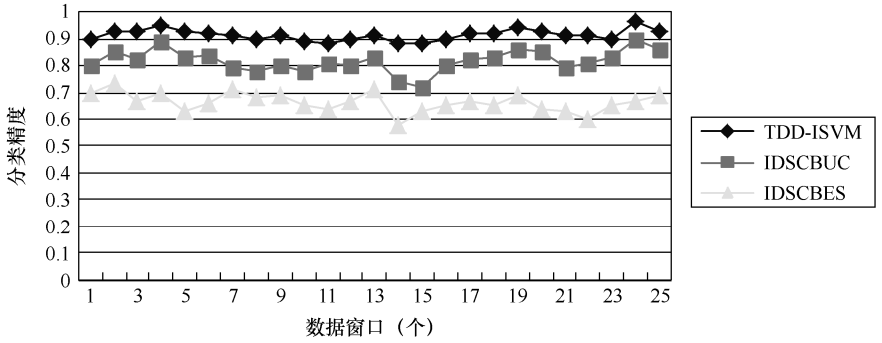

图 3.2　无噪声移动超平面数据集三种算法的分类精度

为了进一步验证该算法的分类精度，这里选择在 SEA 模拟数据集、包含 5%噪声的移动超平面数据集、KDD Cup99 网络入侵数据集上，分别对 TDD-ISVM、IBPNN-CDCA 和 VFDTC 三种算法进行验证。

表 3.1 所示为三种增量学习算法在三种数据集上的分类精度，从表中的数据可以看出，在三种数据集上，算法 TDD-ISVM 的分类精度相对较高。在 SEA 模拟数据集上，IBPNN-CDCA 算法的分类精度最低，TDD-ISVM 算法的分类精度最高，这是因为 SEA 模拟数据集是一种具有突变性质的概念漂移的数据集。由于基于 BP 神经网络的 IBPNN-CDCA 算法的网络结构没有发生变化，因此其对突变式新增概念漂移不能很好地识别；而 TDD-ISVM 算法采用 K-means 聚类的方式可以扩展类别标签，提高对新概念漂移的识别能力。在移动超平面数据集（Hyperplane）上，由于该数据集包含一定的噪声数据和概念漂移，因此 IBPNN-CDCA 算法和 VFDTC 算法的效果相仿，精度较低，而算法 TDD-ISVM 在一定程度上能够区别噪声数据，具有一定的抗噪性，其分类精度较高。三种增量学习算法在 KDD Cup99

数据集上的分类效果相当。

表 3.1　三种增量学习算法在三种数据集上的分类精度

数　据　集	TDD-ISVM 算法	IBPNN-CDCA 算法	VFDTC 算法
SEA	0.93	0.82	0.88
5% Hyperplane	0.89	0.8	0.82
KDD Cup99	0.95	0.94	0.95

2）概念漂移检测方面

不同的算法对概念漂移的检测能力不同，下面分别验证 TDD-ISVM、IBPNN-CDCA 和 VFDTC 这三种算法在概念漂移检测方面的差别。

表 3.2 所示为 TDD-ISVM、IBPNN-CDCA 和 VFDTC 算法在三种数据集上的概念漂移检测情况。从表 3.2 可以看出，无论是在哪种数据集上，算法 TDD-ISVM 的概念漂移检测次数都是最多的，尤其在移动超平面数据集 Hyperplane 上，算法 TDD-ISVM 的概念漂移检测次数甚至超过 IBPNN-CDCA 算法和 VFDTC 算法的检测次数之和，这是因为算法 TDD-ISVM 采用的是基于数据属性和分类性能的二重概念漂移检测机制，检测要求更严格。而误报次数的对比同样说明了这个问题。在移动超平面数据集 Hyperplane 中，不仅有概念漂移，而且有噪声数据，而 TDD-ISVM 算法在利用 K-means 聚类进行概念漂移检测时，若噪声数据量较大，则其容易把一部分噪声数据当成概念漂移现象，进而向系统报告检测到概念漂移。通过较多的检测次数和较多的误报次数，可以尽可能地把所有概念漂移均检测出来。从表 3.2 可以看出，在三种算法中，TDD-ISVM 算法的漏报次数最少。检测次数的增多势必增大模型的复杂度，使时间效率降低，而本实验是在 Spark 集群上进行的，基于内存式计算的 Spark 平台可以有效地解决时间效率问题。

表 3.2　TDD-ISVM、IBPNN-CDCA 和 VFDTC 算法在三种数据集上的概念漂移检测情况

数 据 集	TDD-ISVM 算法			IBPNN-CDCA 算法			VFDTC 算法		
	检测次数	误报次数	漏报次数	检测次数	误报次数	漏报次数	检测次数	误报次数	漏报次数
SEA	76	9	0	42	8	3	55	11	2
Hyperplane	183	16	2	77	10	6	91	12	5
KDD Cup99	216	19	8	115	12	11	137	10	8

3）抗噪性方面

噪声数据是对概念漂移检测的一大干扰，直接影响分类精度的高低。下面在包含 5%噪声数据的 Hyperplane 和不包含噪声数据的 Hyperplane 两种数据集上，对 TDD-ISVM、IBPNN-CDCA 和 VFDTC 这三种算法进行分类精度的对比，观察噪声数据对各算法的影响。

图 3.3、图 3.4 和图 3.5 所示为 TDD-ISVM、IBPNN-CDCA 和 VFDTC 三种算法分别在有噪声数据与无噪声数据的 Hyperplane 数据集上的分类精度对比。对比三幅图可以发现，同样的两种 Hyperplane 数据集，不同的算法不仅在分类精度上存在差异，而且在应对数据集中的噪声问题上的能力不同。图 3.3 中，TDD-ISVM 算法在有噪声和无噪声的 Hyperplane 数据集上的分类精度都较高，并且，噪声数据对分类精度的影响不大，说明 TDD-ISVM 算法有一定的抗噪性。而图 3.4 中的 IBPNN-CDCA 算法和图 3.5 中的 VFDTC 算法在有噪声的 Hyperplane 数据集上的分类精度明显低，且分类精度的波动幅度较大，说明它们的抗噪性不如 TDD-ISVM 算法。TDD-ISVM 算法之所以具有抗噪性，是因为在进行概念漂移检测时，将检测到的特殊

实例分别当成噪声数据和概念漂移数据加以区别，采用不同的方式解决。

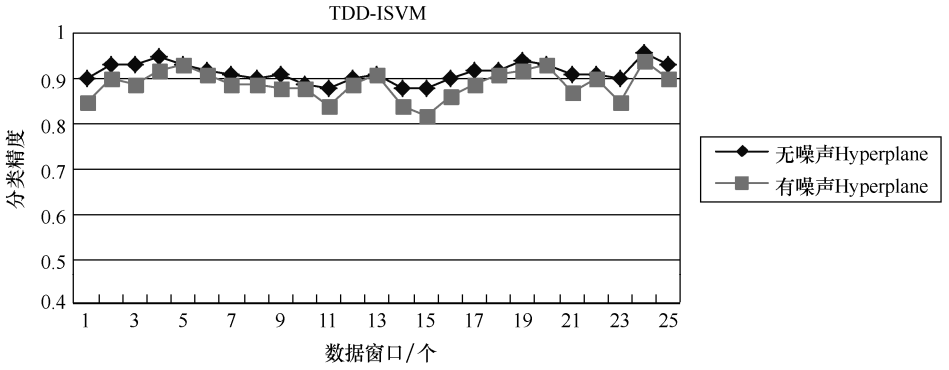

图 3.3　TDD-ISVM 算法在两种 Hyperplane 数据集上的分类精度对比

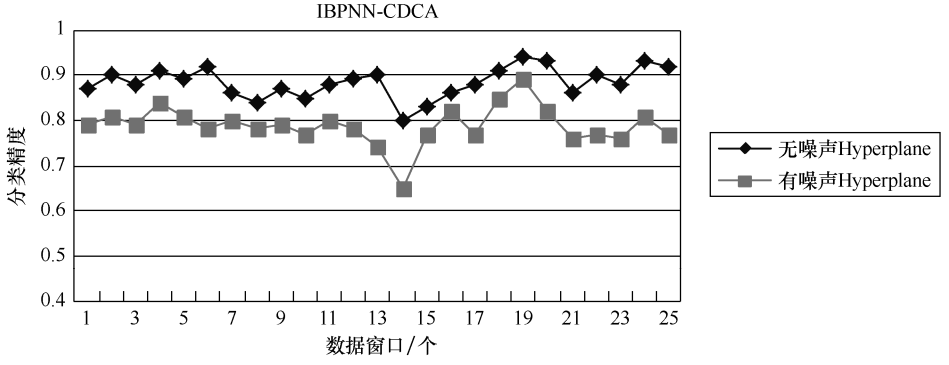

图 3.4　IBPNN-CDCA 算法在两种 Hyperplane 数据集上的分类精度对比

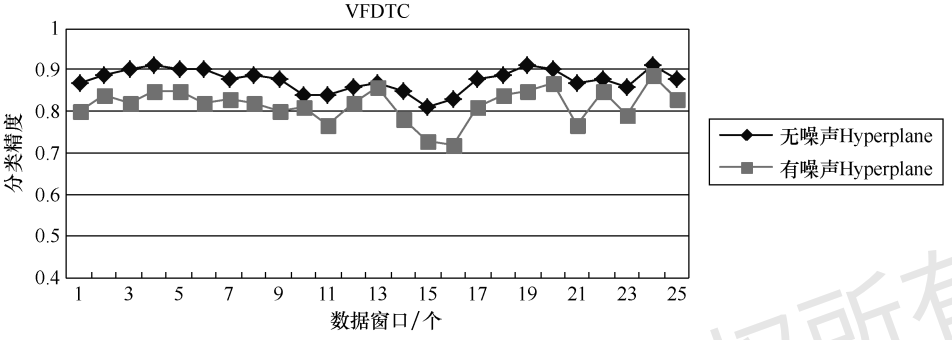

图 3.5　VFDTC 算法在两种 Hyperplane 数据集上的分类精度对比

3.8　本章小结

本章首先介绍了一种基于数据属性和分类性能的二重概念漂移检测机制，接着给出了基于 K-means 聚类的概念漂移检测算法和基于伯努利分布的概念漂移检测算法，这两种检测算法分别可以从数据属性和分类性能两个方面进行概念漂移检测。在此基础上，又介绍

了引入增量学习思想的支持向量机模型，提出了一种基于二重检测的概念漂移数据流分类算法 TDD-ISVM，该算法可以有效地检测不同类型的概念漂移，并根据不同情况采取不同的应对措施，对数据流有较强的适应能力，并且该算法也具有抗噪性，可以在一定程度上检测噪声数据。还介绍了三种概念漂移检测常用的数据集，包括两种人工数据集和一种真实数据集。最后给出了实验的具体结果，并对结果进行分析，验证了本章所提算法具有良好的效果。