流数据的概念漂移检测

Heng Wang

Johns Hopkins University

Email: hwang82@jhu.edu

Zubin Abraham

Research and Technology Center North America

Email: Zubin.Abraham@us.bosch.com

摘要

常见的统计预测模型通常需要并假设数据的平稳性。然而，在许多实际应用中，随着时间的推移，定期观察响应和预测变量之间关系的变化，导致这些模型的预测性能变差。本文将介绍线性四率（LFR），一种用于检测这些概念漂移并随后识别属于新概念的数据点的框架（用于再学习模型）。与传统的概念漂移检测方法不同，LFR 可以应用于批量数据和流数据；它不受响应变量的分布特性的限制（例如，具有不平衡标签的数据集）；并独立于基础统计模型，使用用户指定的直观易懂的参数。将 LFR 的性能与使用跨越概念漂移类型范围的模拟和常用公共数据集的基准方法进行比较，结果表明，LFR 在数据集中概念漂移检测的召回，准确性和延迟方面明显优于基准方法。

I. 介绍

挖掘数据流时的一个共同的挑战是数据流并不总是严格固定的，即数据的概念（输入数据的基础分布）随时间不可预测地漂移。这促使需要及时检测数据流中的这些概念漂移，无论是商业智能还是作为一种追踪使用这些数据流作为输入的统计预测模型性能的手段。

本文侧重于检测影响二元分类模型的概念漂移。对于二元分类问题，据说概念漂移发生在联合分布 $P(\mathbf{X}_t, y_t)$ 随时间变化的时候，其中 $\mathbf{X}_t \in \mathbb{R}^d$ 是时间步长为 t 时的 d 个预测变量，$y_t \in \{0,1\}$ 是相应的二元响应变量。直观地，概念漂移是指生成响应变量的基础分布随时间变化的情况。检测概念漂移的流行方法是确定变化点[1],[2]。DDM 是最广泛使用的概念漂移检测算法，它被严格设计以用于流数据[1]。DDM 采用的检验统计量是总体分类误差之和 $\left(\hat{P}_{e_{rror}}^{(t)}\right)$ 和经验标准差 $\left(\hat{S}_{e_{rror}}^{(t)}\right)$。DDM 关注于整体错误率，因此除非误报和漏报的总和发生变化，否则无法检测到漂移。这种情况的一个例子是：当一个 2×2 的混淆矩阵从 $\begin{pmatrix} 65 & 5 \\ 15 & 15 \end{pmatrix}$ 变化到 $\begin{pmatrix} 75 & 15 \\ 5 & 5 \end{pmatrix}$ 时，就保存它们的整体错误率。如示例所示，这种限制在不平衡的分类任务中更加突出。不幸的是，这种无法在少数类的召回中检测的快速下降常常是重要的。例如，如果上述示例中的少数类别对应于制造工厂中被归类为有缺陷的产品，则“真实正率”（即从 0.75 到 0.25）的这种关键的三倍减少将会被 DDM 忽视。

在线类别不平衡（DDMOCI）的漂移检测方法解决了当类别比例不平衡时 DDM 的局限性[2]。然而，由于模型中固有的缺点，DDM-OCI 触发了许多错误警报。DDM-OCI 假设不平衡分类任务中的概念漂移由潜在的真实正率（即少数类别的召回）的变化表

示。遗憾的是，这一假设并未考虑概念漂移发生而没有影响少数类别召回的情况。可以证明，概念可以从不平衡类数据漂移到平衡类数据，而真实正率 (tpr) ，正的预测值 (ppv) 和 F1 分数保持不变。因此，DDM-OCI 不太可能检测到这种类型的漂移，除非还考虑其他速率，例如真实负率 (tnr) 或负的预测值 (npv) 。另外，DDM-OCI 使用的测试统计量 $R_{tpr}^{(t)}$ 在稳定的概念下的大致分布为并不是 $N\left(P_{tpr}, \frac{P_{tpr}(1-P_{tpr})}{N_{tpr}}\right)$. 因此，构造[1]中规定的置信水平的基本原理不适用于 $R_{tpr}^{(t)}$ 的零分布。这就是 DDM-OCI 快速，频繁地触发误报的原因。

如果数据流具有缓慢的逐渐变化，则早期漂移检测方法（EDDM）比 DDM 实现更好的检测结果。EDDM 监控两个分类错误之间的距离[3]。PerfSim 算法考虑混淆矩阵的所有分量，并监测来自两批数据所有分量的余弦相似系数[4]。如果相似系数低于某个用户指定的阈值，则表示概念漂移。但是，在计算每个决策点的监测统计数据之前，EDDM 需要等待至少 30 个分类错误。也就是说，漂移的决策点之间的时间间隔的长度是取决于出现 30 次分类错误的随机数。在 30 个分类错误之间可能有很多例子。PerfSim 算法还受到收集小批量数据以计算监测统计数据的要求的限制。在[3]，[4]中划分数据流的方法是用户通过训练经验指定的或在检测开始之前学习的。因此，EDDM 和 PerfSim 不是很适合立即做出决策的流媒体环境。[5]中规定的方法利用 SVM 监测三个指标：整体准确性，召回率和精确度随时间的推移。这种方法通过假设数据分批到达来计算三个度量以学习 SVM。为了解决现有方法的局限性，我们提出了线性四率（LFR）来检测 $P(\mathbf{X}_t, y_t)$ 的漂移。与其他提出的方法不同，即使存在不平衡类标签，LFR 也可以检测到概念漂移的所有可能变体，如第 IV 部分所示。LFR 在最早检测

概念漂移方面优于现有方法，具有最少的错误警报和最好的召回。另外，LFR 不要求数据成批到达并且独立于所使用的基础分类器。

II. 问题描述

鉴于概念漂移的检测等同于检测 $P(\mathbf{X}_t, y_t)$ 中的变化点，直观的方法是测试数据流中多变量变量 $(\mathbf{X}_t, y_t)$ 的统计假设[6]，[7]，[8]。这种方法的局限性在于，当 $\mathbf{X}_t$ 的维度 (d) 非常大或者漂移的幅度很小时，统计能力的性能会降低。因此，为了克服这些限制，所提出的方法标识 $P\big(\hat{f}(\mathbf{X}_t), y_t\big)$ 的变化，其中 $\hat{f}$ 是用来预测的分类器。这是因为 $P\big(\hat{f}(\mathbf{X}_t), y_t\big)$ 的任何漂移都意味着 $P(\mathbf{X}_t, y_t)$ 的漂移，且概率为 1。

设 $\hat{f}(\mathbf{X}_t) = y_t$ 是给定数据流 $(\mathbf{X}_t, y_t)$ 的二元分类器。我们为 $\hat{f}$ 定义相应的 2×2 的混淆概率矩阵（*CP*）如下

CP =

Pred \ True	0	1
0	TN	FN
1	FP	TP

其中，*CP*[1,1]，*CP* [0,0]，*CP* [1,0]，*CP* [0,1] 表示分类器 $\hat{f}$ 的基本的比率，分别为真实正率（TP）、真实负率（TN）、错误正率（FP）、错误负率（FN），例如：*CP*[1,1]= $P(y_t = 1, \hat{y}_t = 1)$ 。

四个特征比率（真实正率、真实负率、预测正值、预测负值）可以通过如下的公式计算：$P_{tpr} = TP / (TP + FN)$ ，$P_{tnr} = TN / (TN + FP)$ ，$P_{ppv} = TP / (FP +$

$TP)$，$P_{npv} = TN / (TN + FN)$。如果没有错误分类，则所有提到的 $P_* = \{P_{tpr}, P_{tnr}, P_{ppv}, P_{npv}\}$ 中的特征率等于 1。

在稳定的概念下（即 $P(\mathbf{X}_t, y_t)$ 保持不变），$\{P_{tpr}, P_{tnr}, P_{ppv}, P_{npv}\}$ 保持不变。 因此，任何 P_* 的显着变化都意味着基础联合分布 $(y_t, \hat{y}_t)$ 或概念的变化。 值得注意的是，在每个时间步长 t，对于任何可能的 $(y_t, \hat{y}_t)$ 对，P_* 中的四个经验率中只有两个会改变，这两个率被称为“受 $(y_t, \hat{y}_t)$ 的影响”。 此外，请注意，在某些应用中，如果 P_* 中的所有经验率都在增加，则概念漂移的检测不受关注，因此不必要地发出警报。 这是因为它表明从历史数据中学习的旧模型在当前数据流的分类中表现得更好。我们在本文中没有使用这个假设，但我们在下面提出的所有方法和论据都可以很容易地适应这个假设。

III. 概念漂移检测框架

考虑到 P_*（其中，$* \in \{tpr, tnr, ppv, npv\}$）检测概念漂移的效果，所提出的概念漂移检测框架在每一步使用 P_* 中的速率估计量作为测试统计量来进行统计假设检验。 特别地，该框架在每个时间步长 t 使用以下的 null 和替代的假设进行统计测试：

$$H_0 : \forall *, P\left(P_*^{(t-1)} \text{的估计值}\right) = \left(P_*^{(t)} \text{的估计值}\right)$$

$$H_A : \exists *, P\left(P_*^{(t-1)} \text{的估计值}\right) \neq \left(P_*^{(t)} \text{的估计值}\right)$$

该概念在 H_0 下是稳定的，并且如果 H_0 被拒绝则被认为已经漂移。这个想法是将每个时间步长 H_0 下运行测试统计时的统计的重要性与用户定义的警告 (δ_*) 和检测 (ϵ_*) 的重要性进行比较。这种类型的测试称为“持续测试”[9]，在我们的问题中，所

有时间戳都是接受或拒绝的决策点。然后，当概念稳定时，从长时间运行来看，每隔 $1/\epsilon_*$ 个步骤就会不必要地触发 P_* 上的错误警报。在本文中，我们假设决策点的间距是固定的。 因此，在我们的持续的测试中，家庭错误率及其成本可以通过使用诸如 ϵ_* 上经典 Bonferroni 校正的同时推断方法来控制。在更一般的情况下，决策点的间距不相等且测试统计数据呈强正相关，我们应该考虑测试的平均运行长度[10]或更强大的替代方案来控制家族错误率。

“持续性测试”框架的一个朴素实现（朴素四率）将是使用 $\hat{P}_*^{(t)}$（ $P_*^{(t)}$ 的经验比率）作为估计量和测试统计量。但如第 III-C 部分所示，有更好的估计 $\hat{P}_*^{(t)}$ 的估计值。

在下一节中，将使用线性四速率（LFR）算法来详细说明概念漂移检测框架。LFR 与朴素四率（NFR）在使用的估计量方面有所不同。 然而，由于采用了更全面的检测框架，LFR 和 NFR 都比 DDM 和 DDM-OCI 表现更好。

A. 线性四率（LFR）

*1）算法概要。*LFR 使用修正的比率 $R_*^{(t)}$ 作为 $P_*^{(t)}$ 的测试统计。$R_*^{(t)}$ 是经验比率 $P_*^{(t)}$ 的一个修正版本。在每个 t，$R_*^{(t)}$ 将受 $(y_t, \hat{y}_t)$ 而被更新为：$R_*^{(t)} \leftarrow \eta_* R_*^{(t-1)} + (1-\eta_*)1_{\{y_t=\hat{y}_t\}}$ 。$R_*^{(t)}$本质上是分类器之前的性能 $R_*^{(t-1)}$ 和当前性能 $1_{\{y_t=\hat{y}_t\}}$，的线性组合，其中 η_* 是用于加权当前实例的分类器性能的时间衰减因子。 $R_*^{(t)}$ 已被用作类不平衡检测器，并作为[11][2]中修正的召回测试统计量。我们的测试统计量 $R_*^{(t)}$

的概率特征在§III-A2中进行了研究。在算法1中详细描述了框架的伪代码（使用 $R_*^{(t)}$ 作为 $P_*^{(t)}$ 的估计值用于所需的测试统计）。

算法 1 线性四率方法（LFR）

输入： 数据：$\{(\mathbf{X}_t, y_t)\}_{t=1}^{\infty}$ 其中 $\mathbf{X}_t \in \mathbb{R}^d$，$y_t \in \{0,1\}$；
二元分类器 $\hat{f}(\cdot)$；时间衰减因子 η_*；
警告重要性 δ_*；检测重要性ϵ_*.

输出： 检测到概念漂移的时间 (t_{cd}).

$P_*^{(0)} \leftarrow 0.5$，$R_*^{(0)} \leftarrow 0.5$，其中$* \in \{tpr, tnr, ppv, npv\}$，混淆矩阵 $C^{(0)} \leftarrow [1,1;1,1]$
for $t = 1$ *to* ∞ **do**
  $\hat{y}_t \leftarrow \hat{f}(\mathbf{X}_t)$
  $C^{(t)}[\hat{y}_t][y_t] \leftarrow C^{(t-1)}[\hat{y}_t][y_t] + 1$
  for each $* \in \{tpr, tnr, ppv, npv\}$ **do**
    if ($*$ is influenced by $(y_t, \hat{y}_t)$) **then**
      $R_*^{(t)} \leftarrow \eta_* R_*^{(t-1)} + (1-\eta_*)1_{\{y_t = \hat{y}_t\}}$
    else
      $R_*^{(t)} \leftarrow R_*^{(t-1)}$
    end if
    if ($* \in \{tpr, tnr\}$) **then**
      $N_* \leftarrow C^{(t)}\left[0, 1_{\{*=tpr\}}\right] + C^{(t)}\left[1, 1_{\{*=tpr\}}\right]$
      $\hat{P}_*^{(t)} \leftarrow \frac{C^{(t)}\left[1_{\{*=tpr\}}, 1_{\{*=tpr\}}\right]}{N_*}$
    else
      $N_* \leftarrow C^{(t)}\left[1_{\{*=ppv\}}, 0\right] + C^{(t)}\left[1_{\{*=ppv\}}, 1\right]$
      $\hat{P}_*^{(t)} \leftarrow \frac{C^{(t)}\left[1_{\{*=ppv\}}, 1_{\{*=ppv\}}\right]}{N_*}$
    end if
    warn.bd$_*$ $\leftarrow$ BoundTable$(\hat{P}_*^{(t)}, \eta_*, \delta_*, N_*)$
    detect.bd$_*$ $\leftarrow$ BoundTable$(\hat{P}_*^{(t)}, \eta_*, \delta_*, N_*)$
  end for
  if (any $R_*^{(t)}$ exceeds warn.bd$_*$ & warn.time = 0) **then**
    warn.time $\leftarrow t$
  else if (no $R_*^{(t)}$ exceeds warn.bd$_*$) **then**
    warn.time $\leftarrow 0$
  end if
  if (any $R_*^{(t)}$ exceeds detect.bd$_*$) **then**
    detect.time $\leftarrow t$
    relearn $\hat{f}(\cdot)$ using $\{(\mathbf{X}_t, y_t)\}_{t=\text{warn.time}}^{\text{detect.time}}$
    reset $R_*^{(t)}, \hat{P}_*^{(t)}, C^{(t)}$ as done in Step 1
    return $t_{cd} \leftarrow t$
  end if
end for

三个用户定义的参数是每个比率的时间衰减因子（η_*），警告重要性（δ_*）和检测重要性（ϵ_*）。时间衰减因子是[0,1]之间的权重，以评估分类器 $\hat{f}$ 预测当前实例 $\hat{f}(\mathbf{X}_t)$ 的性能。考虑到检测方法是在每个时间步骤进行假设检验，在标准测试框架中，δ_* 和 ϵ_* 是可解释的统计重要性，即 I 型错误（误报率）。在实践中，在诸如移动装配线的质量控制之类的应用中，所允许的错误警告率和错误检测率是帮助用户选择参数 δ_* 和 ϵ_* 的指南。为了公平比较，对于本文的所有实验，η_* 被设置为与[2]中相同的值 0.9。η_* 的最优选择是依赖于值域的，并且可以在必要时预先学习。

第 III-A2 节中的定理 1 表明，在稳定概念下，$R_*^{(t)}$ 是独立同分布的伯努利随机变量的几何加权和，它强调最近的预测精度，并将指数衰减的权重置于历史预测精度上。利用这种加权方案，$R_*^{(t)}$ 对概念漂移更加敏感，预示了分类器性能的不平稳性。

根据定理 1，我们能够克服[2]的缺陷，为 $R_*^{(t)}$ 构造一个运行更可靠的置信区间来控制 I 型误差 ϵ_*。$R_*^{(t)}$ 分布为伯努利随机变量的几何加权和。Bhati 等人研究了特殊情形 P_* = 0.5 [12]下 $R_*^{(t)}$ 的闭式分布函数。然而，对于 P_* 的其他值，不能得到闭式分布函数。或者，根据定理 1，对于给定的 P_*、N_* 和时间衰减因子 η，也可以通过蒙特卡罗模拟独立地获得合理的经验分布。蒙特卡罗采样过程的伪码在算法 2 中提供。由于 P_* 未知，$\hat{P}_*$ 被用作其替代，以产生 $R_*^{(t)}$ 的经验分布。基于经验分布，给定重要性 α 的下分位数和上分位数用作所需的（警告/检测）边界。引理 1 支持选择 $\hat{P}_*$ 作为 P_* 的最佳代理。

δ_* 和 ϵ_* 分别表示警告和检测重要性，其中 $\delta_* > \epsilon_*$。相应的 warn.bd 和 detect.bd 是从蒙特卡罗模拟中获得的。框架的四个比率 $\{tpr, tnr, ppv, npv\}$ 的边界可以通过具有不同的 ϵ_*。例如，在一些不平衡的分类任务中，分类器在少数类上的性能比在多数类上的性能具有更高的优先级。

在计算了界限之后，当任何 $R_*^{(t)}$ 第一次超过相应的警告界限（warn.bd）时，框架认为可能出现概念漂移，并设置警告信号（warn.time $\leftarrow t$）。如果任何 $R_*^{(t)}$ 达到相应的相关检测界限(detect.bd)，则概念漂移发生(derect.time $\leftarrow t$）。

所有存储在 warn.time 和 derect.time 之间的示例，被提取用来重新学习新的分类器，因为所存储的示例被认为是新概念的样本。如果存储的示例数量太少，无法重新学习合理的分类器，则必须等待足够的训练示例。但是，如果 $R_*^{(t)}$ 跨越相应的警告界限 warn.bd，但是未能到达 detect.bd，则将删除先前的警告标志。在检测到概念漂移后，将 $R_*^{(t)}$，$\hat{P}_*^{(t)}$，$C^{(t)}$重置为它们的初始值，以便重新启动新的监测周期。

*2)分析：*以下定理研究了 LFR 检验统计 $R_*^{(t)}$ 的性质。

定理 1：当达到时间 T 的稳定概念时：即 $R_*^{(t)}$ =（1−η_*）$\sum_{i=1}^{N_*} \eta_*^{N_*-i} I_i$，对于任意 $*$，$R_*^{(t)}$ 是伯努利随机变量的几何加权和。其中 $\{I_i\}_{i=1}^{N_*} \sim Bernoulli(P_*)$ 和 P_* 是基础比率。

算法 2 LFR 中 BoundTable 的生成算法

输入： $\hat{P}$ 的估计值；时间衰减因子 η；重要性 α；时间步数 N_*；
随机变量的个数$num.of.MC$;

输出： 重要性的数值 α.

1: **for** $j = 1$ *to* $num.of.MC$ **do**
2: 生成 N_* 个独立的伯努利随机变量 $\{I_1, I_2 \ldots, I_{N_*}\}$，其中 $I_i \sim Bernoulli(\hat{P})$
3: $R[j] \leftarrow$（1−η）$\sum_{i=1}^{N_*} \eta_*^{N_*-i} I_i$
4: **end for**
5: $\{R[j]\}_{j=1}^{num.of.MC}$ 构造经验分布 $\hat{F}(\mathrm{R})$，找到 α 分位数作为下界: $lb \leftarrow$分位数（$\hat{F}(\mathrm{R})$，α）和（1−α）分位数作为上界 $ub \leftarrow$分位数（$\hat{F}(\mathrm{R})$，$1-\alpha$）

证明: 在全部时间步 T 中，假设 $R_*^{(t)}$ 在时间步骤 $T_1, \ldots, T_{N_*}$ 根据第 7 行发生改变，其中 $T_1 < T_2 < \cdots < T_{N_*}$。因此，

$$
\begin{aligned}
R_*^{(T)} &= R_*^{(T_{N_*})} = \eta_* R_*^{(T_{N_*-1})} + (1-\eta_*)1\{y_{T_{N_*}} = \hat{y}_{T_{N_*}}\} \\
&= \eta_* \left[\eta_* R_*^{(T_{N_*-2})} + (1-\eta_*)1\left\{y_{T_{N_*}-1} = \hat{y}_{T_{N_*}-1}\right\}\right] + (1-\eta_*)1\{y_{T_{N_*}} = \hat{y}_{T_{N_*}}\} \\
&= \eta_*^2 R_*^{(T_{N_*-2})} + \eta_*(1-\eta_*)1\left\{y_{T_{N_*}-1} = \hat{y}_{T_{N_*}-1}\right\} + (1-\eta_*)1\left\{y_{T_{N_*}} = \hat{y}_{T_{N_*}}\right\} \\
&= \cdots \\
&= (1-\eta_*)\sum_{i=1}^{N_*} \eta^{N_*-i} 1\{y_{T_i} = \hat{y}_{T_i}\} \\
&= (1-\eta_*)\sum_{i=1}^{N_*} \eta^{N_*-i} I_i
\end{aligned}
$$

其中最后一个方程由稳定概念假设维持，并且所有的因子是 P_* 的独立同分布的伯努利随机变量。

*引理 1：*假设定理 1 中在稳定概念下的设置，对于任何 $* \in \{tpr, tnr, ppv, npv\}$，$\hat{P}_*^{(T)}$ 是 P_* 的唯一均匀最小方差无偏估计量（UMVUE）。当 $T \to \infty$ 时，$\hat{P}_*^{(T)}$大体服从 $\mathrm{N}\left(P_*, \frac{P_*(1-P_*)}{N_*}\right)$分布。

证明：$\hat{P}_*^{(T)}$ 是 P_* 的无偏估计。这是因为 $\hat{P}_*^{(t)} = \frac{\sum_{i=1}^{N_*} X_{T_i}}{N_*}$ 其中 $\{X_{T_i}\}_{i=1}^{N_*}$ 是独立同分布的伯努利随机变量在时间 T_i 和 P_* 实现。通过分解定理，$\hat{P}_*^{(t)}$ 是一个充分的统计量。同时，

$$E\left(g\left(\hat{P}_*^{(T)}\right)\right) = \sum_{i=0}^{N_*} \binom{N_*}{i} P_*^i (1 - P_*)^{N_* - i} g\left(\frac{i}{N_*}\right)$$

$$= N_*! (1 - P_*)^{N_*} \sum_{i=0}^{N_*} \frac{g\left(\frac{i}{N_*}\right)}{i!\,(N_* - i)!} \left(\frac{P_*}{1 - P_*}\right)^i$$

如果 $E\left(g\left(\hat{P}_*^{(T)}\right)\right) = 0$ 则对于 $\forall P_*$，都有 $g\left(\frac{i}{N_*}\right) = 0$。因为 $E\left(g\left(\hat{P}_*^{(T)}\right)\right)$ 是 $\left(\frac{P_*}{1-P_*}\right)$ 的一个多项式。从而 $P\left(g\left(\hat{P}_*^{(T)}\right) = 0\right) = 1$，并且根据定义，$\hat{P}_*^{(T)}$ 是一个充分的统计数据。由 Lehmann-Scheffe 定理可知，$\hat{P}_*^{(T)}$ 是唯一的 UMVUE。

线性四率（LFR）检测算法对每个时间步长的复杂度为 $O(1)$。LFR 算法可以通过使用算法 2 预先计算的 $BoundTable$ 进行优化。在运行算法 1 之前，可以预先计算并存储具有不同输入（$\hat{P}$，η，δ，N_*）的 4 维 $BoundTable$。由于观察者可以从 $BoundTable$ 中查找最接近 $\hat{P}$ 的 $\hat{P}_*$ 来查找上下分位数，在流监控期间不必花费任何计

算资源来计算分位数。因此，LFR 算法采用 $O(1)$ 检验每个时间点的漂移发生，适合于流式环境。

B. 朴素四率算法（NFR）

为了进行比较，本节详细介绍了所提出的使用 $\hat{P}_*^{(t)}$ 作为测试统计量的框架简单实现的特性。选择这个检验统计量的一个优点是，存在一个如引理 1 所示的闭式分布。使用和 LFR 算法相同的策略，NFR 算法依次监视四个比率 $\hat{P}_*^{(t)}$ 。在每个时间段，对于每个比率，假设检验使用零分布 $\mathrm{N}\left(P_*, \frac{P_*(1-P_*)}{N_*}\right)$ 和当 $\hat{P}_*^{(t)}$ 超过预期界限时设置的警告/检测警报来完成。

NFR 和 LFR 的主要区别在于发现零分布的 P_* 的估计。LFR 算法使用 $\hat{P}_*^{(t)}$ 作为未知 P_* 的代替，而 NFR 算法使用 $\bar{P}_*^{(t)}$，其中 $\bar{P}_*^{(t)}$ 是所有先前运行的 $\hat{P}_*^{(t)}$的平均值。这个更新规则允许之前的预测性能对 P_* 的估计贡献更多，而最近的预测贡献较少。因此，当概念漂移发生时， $\bar{P}_*^{(t)}$ 在估计底层 P_* 方面更加稳健。此外，在引理 2 中提出的稳定概念下，$\bar{P}_*^{(t)}$仍然是和 MSE 一致的估计量。

引理 2： 假设定理 1 中在稳定概念下直到时间 T 的设置，对于任何 $* \in \{tpr, tnr, ppv, npv\}$，NFR 算法的 $\bar{P}_*^{(t)}$ 是 P_* 的 MSE 一致估计量。

证明： 在全部时间步 T 中，假设 $\hat{P}_*^{(t)}$ 随时间 $T_1, \ldots, T_{N_*}$ 发生变化。其中 $T_1 < T_2 < \cdots < T_{N_*} < T$。因此，

$$\bar{P}_*^{(T)} = \bar{P}_*^{(T_{N_*})} = \frac{1}{N_*}\sum_{i=1}^{N_*} \hat{P}_*^{(T_i)}$$

$$= \frac{1}{N_*}[\sum_{n=1}^{N_*}\frac{1}{n}\sum_{i=1}^{n} 1\{y_{T_i} = \hat{y}_{T_i}\}]$$

$$= \frac{1}{N_*}[\sum_{n=1}^{N_*}\sum_{j=i}^{N_*}\frac{1}{j} 1\{y_{T_i} = \hat{y}_{T_i}\}]$$

由独立同分布假设，可得

$$\mathbb{E}\left(\bar{P}^{(T)}\right) = \frac{1}{N_*}[\sum_{i=1}^{N_*}\sum_{j=i}^{N_*}\frac{1}{j}\ P_*] = \ P_*$$

$$\mathbb{VAR}\left(\bar{P}^{(T)}\right) = \frac{1}{{N_*}^2}[\sum_{i=1}^{N_*}(\sum_{j=i}^{N_*}\frac{1}{j})^2\ P_*(1 - \ P_*)]$$

$$\leq \frac{(\sum_{j=1}^{N_*}\frac{1}{j})^2}{N_*}\ P_*(1 - \ P_*)$$

$$\leq \frac{(\log N_*)^2}{N_*}\ P_*(1 - \ P_*) \xrightarrow{T\to\infty} 0$$

其中当 $T \to \infty$ 时，$N_* \to \infty$。因此，当 $\mathbb{E}\left(\bar{P}_*^{(T)}\right) - \ P_* = 0$ 且 $\mathbb{VAR}\left(\bar{P}^{(T)}\right) \to 0$ 时，$\bar{P}^{(T)}$ 是一个 MSE 一致估计量。

C. NFR 与 LFR 比较

为了对 NFR 和 LFR 的测试统计量进行比较，我们使用图 1 来说明在相同的合成流数据 $\{(y_t, \hat{y}_t)\}_{t=1}^{T}$ 上 LFR 和 NFR 算法的单次运行。通过从两个混淆概率矩阵 $CP^{(1)}$ 和

$CP^{(2)}$中采样，生成 $T/2$ 处具有一个变化点的 $\{(y_t,\hat{y}_t)\}_{t=1}^{T}$ 的数据流。这两个概念分别用 $CP^{(1)}$ 和 $CP^{(2)}$来表征。漂移的类型由$(CP^{(1)},CP^{(2)})$的特定设置确定。在这个例子中，为了生成一个平衡的 $\{(y_t,\hat{y}_t)\}_{t=1}^{T}$ 来表示分类器的总体精度下降但 P_{tpr} 保持不变的情况，我们选择 $CP^{(1)}=\begin{pmatrix}0.4 & 0.1\\ 0.1 & 0.4\end{pmatrix}$ 和 $CP^{(2)}=\begin{pmatrix}0.3 & 0.1\\ 0.2 & 0.4\end{pmatrix}$。检测算法的目标是识别转换点$T/2$ 。

显然，LFR 算法中的检验统计量 $R_*^{(t)}$ 对于每个比率都比 $\hat{P}_*^{(t)}$ 具有更大的方差。LFR 算法报告了在 t =5167（真实检测点 t=5000）有较 NFR 更早的检测，即使 $\epsilon_*^{(LFR)}<\epsilon_*^{(NFR)}$。这个观测值与 §III − A1中描述的构造 $R_*^{(t)}$ 的原理非常吻合，通过引入大方差来获得检测灵敏度。为了严格比较 $R_*^{(t)}$ 和 $\hat{P}_*^{(t)}$ 的检测性能，下面提供更多的研究。

利用合成数据对两种竞争检验统计量 $R_*^{(t)}\,(LFR)$ 和 $\hat{P}_*^{(t)}(NFR)$ 的性能特性进行了实证比较。我们分别用 $\hat{\beta}_{R_*^{(t)}}$ 和 $\hat{\beta}_{\hat{P}_*^{(t)}}$ 表示 $R_*^{(t)}$ 和 $\hat{P}_*^{(t)}$ 的性能估计。抗时滞变化 k 和 q_* 的 $\hat{\beta}_{R_*^{(t)}}$ 和 $\hat{\beta}_{\hat{P}_*^{(t)}}$ 如图 2 所示。图 2 表明，$R_*^{(t)}$ 和 $\hat{P}_*^{(t)}$ 都不是总占主导地位的，因为 $R_*^{(t)}$ (红色区域)在时间滞后 K 较小时获得较大的统计性能，而在 K 较大时获得较小的性能。这是因为更新规则线 7 使得估计器 $R_*^{(t)}$ 能够以指数速率从 p_* 到 q_* 从而在短时间滞后中形成性能优势，但代价是 $R_*^{(t)}$ 在零和备选下的极限分布比 $\hat{P}_*^{(t)}$ 具有更大的方差，因此当 K 很大而 $|p_*-q_*|$ 很小时，性能降低。

为了比较 $R_*^{(t)}$ 和 $\hat{P}_*^{(t)}$ 在更一般设置中检测概念漂移的灵敏度，我们定义 $\Delta\beta = \hat{\beta}_{R_*^{(t)}} - \hat{\beta}_{\hat{P}_*^{(t)}}$。结果如图 3 所示。除了当 $p_* = q_*$ 时，我们看到对于任何固定的 (p_*, q_*),当 K 很小时 $\Delta\beta \geq 0$;当 K 很大时 $\Delta\beta \leq 0$。这是因为 $\Delta\beta$ 随着 K 的增加而减少。这表明如果需要早期检测，LFR 是更可取的，警报更可能在概念漂移发生后的最早时间触发。早期检测允许观察者调整模型从而避免错误预测的损失。另一方面，如果观察者只关心检测数据流中漂移的发生而不关心其检测的快速性，那么 NFR 算法提供了更高的性能检测统计量来检测漂移。这是因为 $\hat{P}_*^{(t)} \to q_*$ 的收敛速度为 $O(\frac{1}{K})$ 。从长远来看，随着 $K \to \infty$，$\hat{P}_*^{(t)} \to q_*$ 表明 $\hat{\beta}_{\hat{P}_*^{(t)}} \to 1$ 。

为了指导 LFR 和 NFR 之间的选择，图 4 是使用 $K = 200$ 并限制所有 (p_*, q_*) 的性能估计热图。我们可以看到，对于$K = 200$，当 p_* 和 q_* 明显不相同时， $\hat{\beta}_{R_*^{(t)}}$ 已经近似为 1。

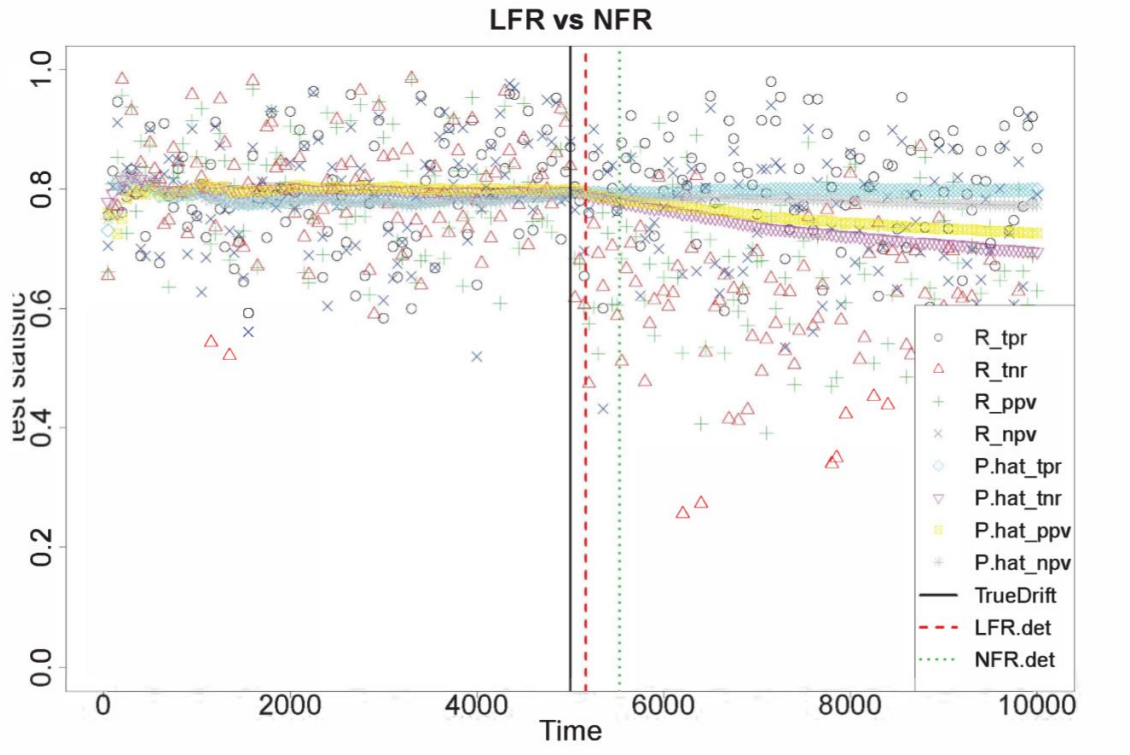

图 1. 在相同的合成流数据上单次运行 LFR 和 NFR。 黑色、红色和绿色垂直线分别表示“真正的漂移时间”、LFR 检测时间和 NFR 检测时间。四个彩色圆点（黑色、红色、绿色、蓝色）是正在运行的 $R_*^{(t)}$ 和四条彩色水平线（靛蓝色、粉红色、黄色、灰色）是正在运行的 $\hat{P}_*^{(t)}$其中 $*\in\{tpr, tnr, ppv, npv\}$ 。

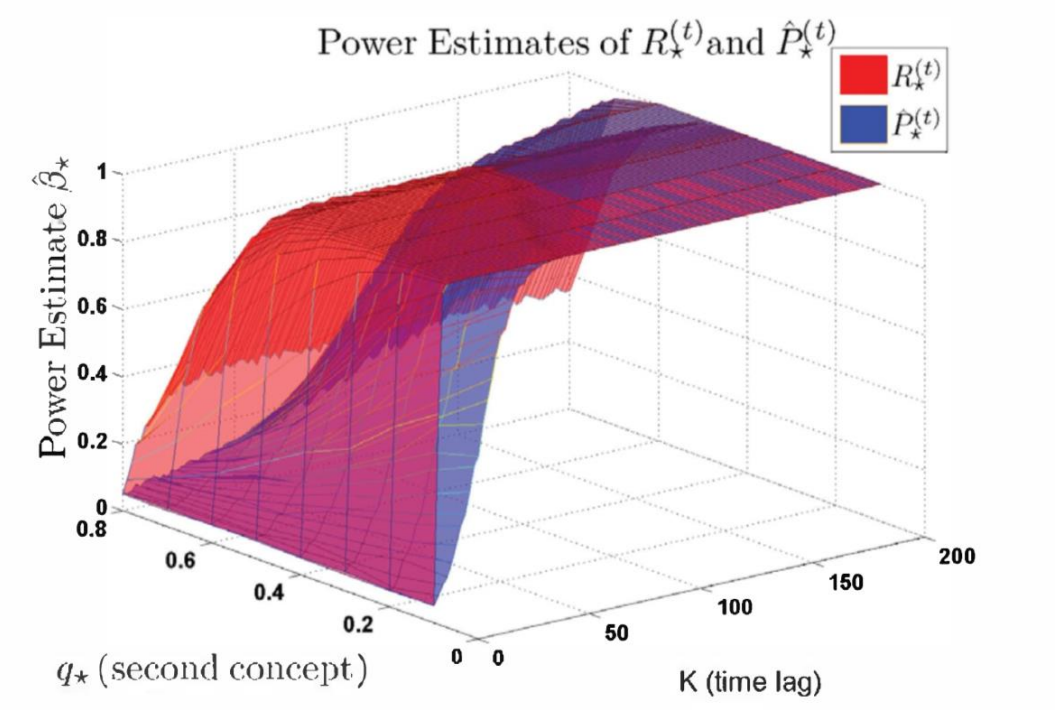

图 2. $R_*^{(t)}$ 和 $\hat{P}_*^{(t)}$ 的性能比较，其中零分布在 $t = M$，而替代分布在 $t = M + k$。M=1000 ，$1 \le k \le K$ 其中$K = 200$ 是最大时滞。基本比率从 $p_* = 0.9$ 漂移到 q_*，其中 $0.1 \le q_* \le 0.8$ 。

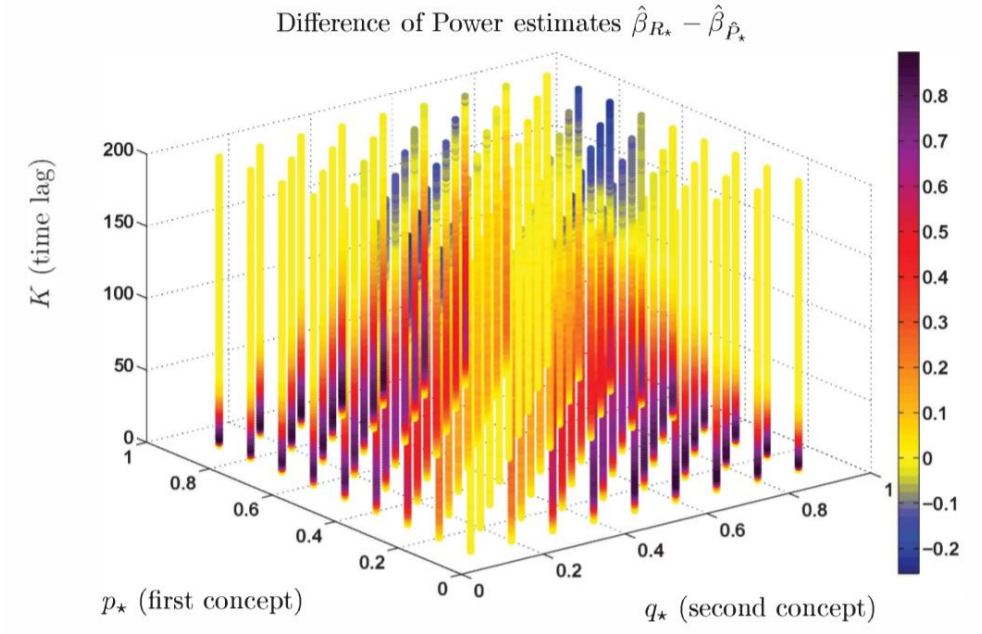

图 3. 随着时滞 $\boldsymbol{K}$ 从 p_* 到 q_*的不同的概念变化组合的性能差 $\Delta\beta = \hat{\beta}_{R_*^{(t)}} - \hat{\beta}_{\hat{P}_*^{(t)}}$。

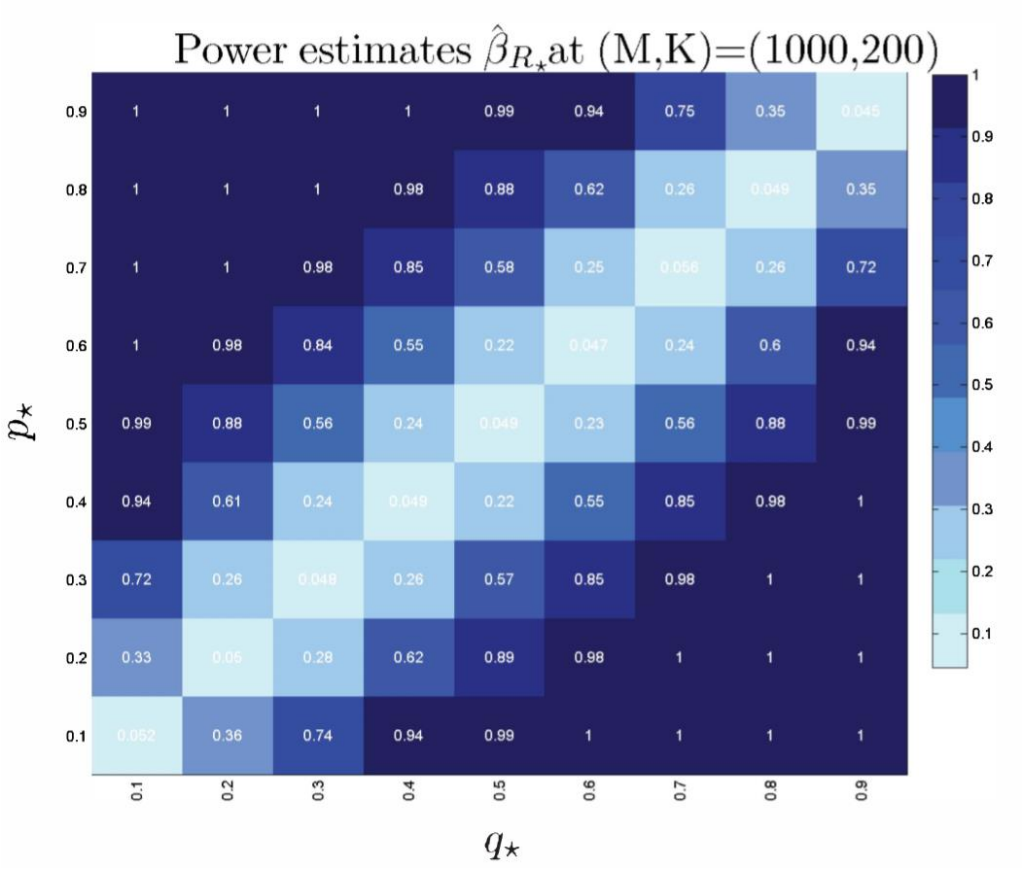

图 4.性能估计$\hat{\beta}_{R_*^{(t)}}$热图。

IV. 实验

在本节中，我们使用合成数据和公共数据集比较了 LFR 与 NFR、DDM 和 DDM-OCI 方法的检测性能。我们考虑了 3 个模拟的类平衡数据集，3 个模拟的类不平衡数据集和 4 个公共数据集，以证明 LFR 算法在各种类型的概念漂移中表现良好，包括基线表现不佳的那些。

为了概括性能并评估算法的置信度，我们使用自举技术。对于每个合成数据集，我们生成 100 个$\{(\mathrm{y}_t, \hat{\mathrm{y}}_t)\}_{t=1}^{T}$而不是 $\{(\mathbf{X}_t, \mathrm{y}_t)\}_{t=1}^{T}$的数据流，以便检测算法的比较独立于所采用的分类器;对于每个公共数据集,每个概念中的 $(\mathbf{X}_t, \mathrm{y}_t)$ 的顺序被置换以创建 100 个自举数据集流。将每个流送到所有检测算法以获得每种方法的单次检测。为了

说明预测的准确性，我们使用重叠直方图来显示从 100 次运行中的概念漂移检测模型获得的检测点的分布。为避免冗余，我们在 10 个实验中提供 6 个直方图，其余的相似。如下所示，LFR 始终优于基线方法。与 NFR 相比，LFR 可以更准确地识别更多真实的漂移点，并且即使使用较小的 ϵ_*，也可以减少错误警报的数量。

A. 合成数据

对合成数据进行了大量实验，涵盖了各种类型的概念漂移。在每个引导程序中，通过使用 **§ III − C**中引入的相同机制生成 $\{(\mathrm{y}_t, \hat{\mathrm{y}}_t)\}_{t=1}^{T}$ 的数据流，其中在 $T/2$ 处具有一个变化点。检测算法的目标是识别变化点 $T/2$。下面讨论 6 个具有挑战性和有趣的场景。

*1）平衡数据集：*在平衡数据集中，基础数据生成需要 $P(y_t = 0) = P(y_t = 1)$。平衡数据是分类任务中最典型的情况，因此通过以下三个代表性实验进行了研究。

（i）Balance1：分类器的总体准确度下降，但 P_{tpr} 保持不变

$$CP^{(1)} = \begin{pmatrix} 0.4 & 0.1 \\ 0.1 & 0.4 \end{pmatrix} \quad CP^{(2)} = \begin{pmatrix} 0.3 & 0.1 \\ 0.2 & 0.4 \end{pmatrix}$$

（ii）Balance2：逐渐漂移，其中总体准确度 $(1 - P_{error})$ 保持不变

$$CP^{(1)} = \begin{pmatrix} 0.35 & 0.05 \\ 0.15 & 0.45 \end{pmatrix} \quad CP^{(2)} = \begin{pmatrix} 0.4 & 0.1 \\ 0.1 & 0.4 \end{pmatrix}$$

（iii）Balance3：总体准确度 $(1 - P_{error})$ 增加而 P_{tpr} 保持不变

$$CP^{(1)} = \begin{pmatrix} 0.3 & 0.2 \\ 0.2 & 0.3 \end{pmatrix} \quad CP^{(2)} = \begin{pmatrix} 0.4 & 0.2 \\ 0.1 & 0.3 \end{pmatrix}$$

*2）不平衡数据集：*对于不平衡数据集，我们使用与平衡情况相同的数据生成机制，但使 $P(y_t = 0)$ 和 $P(y_t = 1)$ 不平衡。我们考虑了以下三种有趣类型的概念漂移，并在实际应用中给予了很多关注。

(i) Imbalance1: 从类平衡数据集到类不平衡数据集

$$CP^{(1)} = \begin{pmatrix} 1/3 & 1/6 \\ 1/6 & 1/3 \end{pmatrix} \quad CP^{(2)} = \begin{pmatrix} 13/15 & 1/30 \\ 1/30 & 1/15 \end{pmatrix}$$

不失一般性，让 $y = 1$ 成为少数类。 值得注意的是，漂移发生后 P_{tpr} 和 P_{ppv} 没有变化。因此，不平衡数据学习社会中的许多检测器时，使用 F1 分数作为监视分类器性能的措施不能警告这种类型的漂移。然而，图 7 显示 LFR 通过控制较额高早期检测率和误报率而表现得非常好。此外，由于 $(1 - P_{error})$ 和 P_{tpr} 的增量，DDM 和 DDM-OCI 在变换点之后没有检测到漂移。

(ii) Imbalance2: 类的比率和 P_{error} 保持不变但 P_{tpr} 减少

$$CP^{(1)} = \begin{pmatrix} 0.65 & 0.05 \\ 0.15 & 0.15 \end{pmatrix} \quad CP^{(2)} = \begin{pmatrix} 0.75 & 0.15 \\ 0.05 & 0.05 \end{pmatrix}$$

(iii) Imbalance3: P_{tpr} ，P_{ppv} 和 $1 - P_{error}$ 都减少。尽管类的比率保持不变，但是 F1 得分和总体准确度都会降低。选择以下两个条件概率矩阵

$$CP^{(1)} = \begin{pmatrix} 0.6 & 0.15 \\ 0.15 & 0.1 \end{pmatrix} \quad CP^{(2)} = \begin{pmatrix} 0.6 & 0.15 \\ 0.15 & 0.1 \end{pmatrix}$$

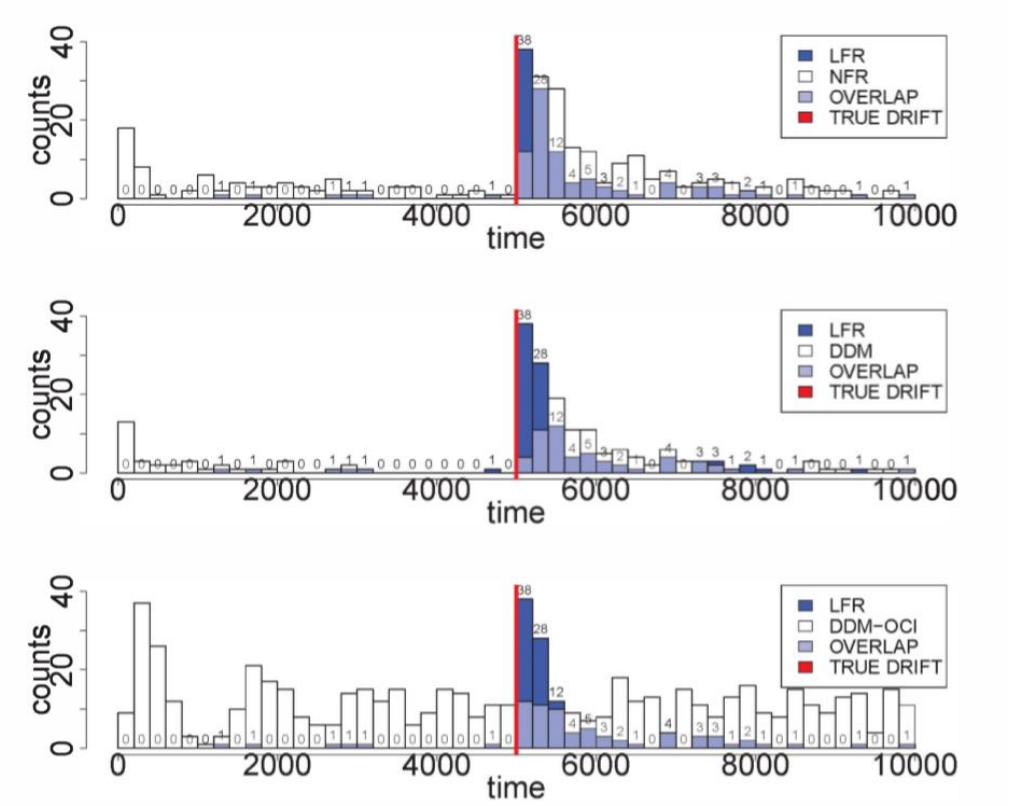

图 5. 比较平衡数据集 1 上的检测时间的重叠直方图，其中分类器的总体准确度下降P_{tpr}保持不变。LFR 的计数数量高于每个矩形的顶部。

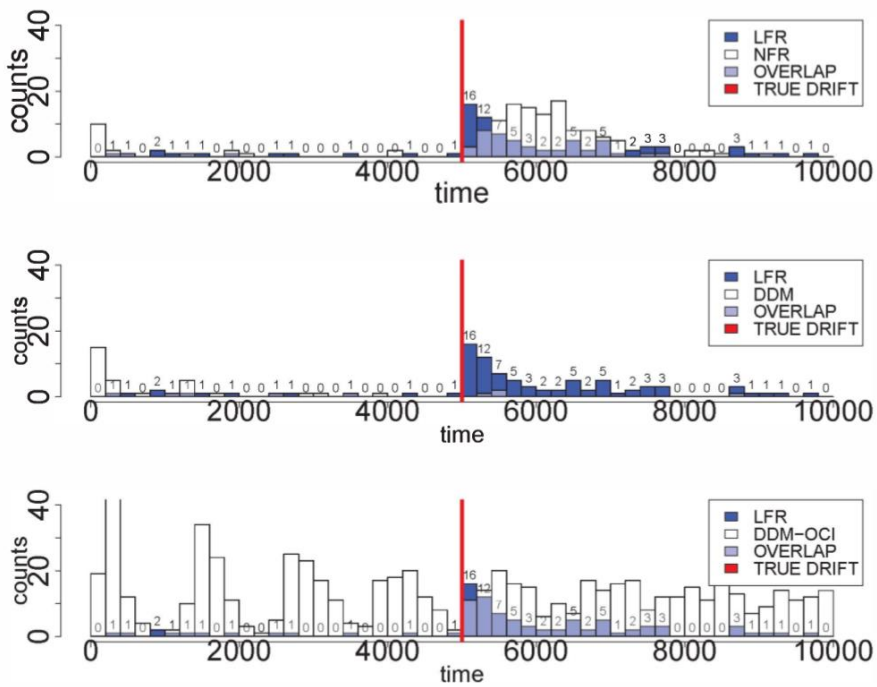

图 6. 比较平衡数据集数据集 2 上的检测时间的重叠直方图，其中发生逐渐漂移但总体精度保持不变。LFR 的计数数量高于每个矩形的顶部。

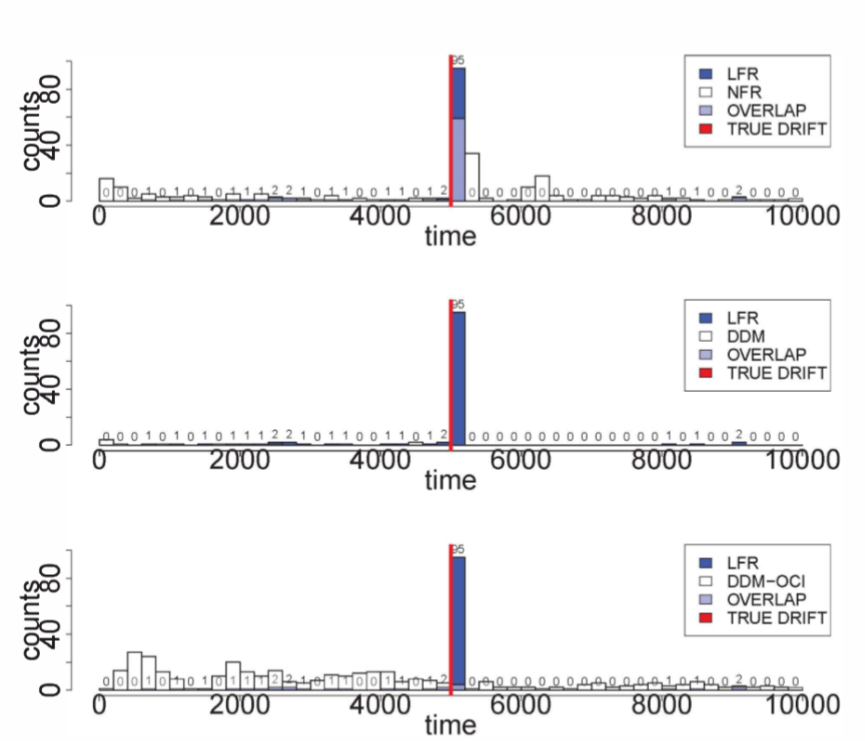

图 7. 重叠直方图比较不平衡数据集 1 上的检测时间，其中类比率从 1：1 到为 9：1 但 Fl-score 保持不变。LFR 的计数数量高于每个矩形的顶部。

B. 公开数据集

所有检测算法都在文献中使用的四个公共数据集上进行评估。在不失一般性的情况下，我们选择了支持向量机（SVM）[13]，其中使用 RBF 内核的分类器 $\hat{f}$，因为所有检测算法都与分类器的类型无关，少数类别的错误分类比大多数类别多 100 倍。如果算法报告潜在的概念漂移，则将存储来自新概念的示例以重新训练新的 SVM 分类器 $\hat{f}_{new}$ 适应新概念。具体而言，1000 个示例用于 SEA 和旋转超平面数据集的重新训练；100 个示例用于对 USENETI 和 USENET2 数据集进行再训练。

Dataset	T	True Drift Time	dimensions(d)
SEA	60000	$\{15000 \times i\}_{i=1}^{3}$	3
HYPER	90000	$\{15000 \times i\}_{i=1}^{8}$	10
USENET1	1500	$\{300 \times i\}_{i=1}^{5}$	100
USENET2	1500	$\{300 \times i\}_{i=1}^{5}$	100

表 1. 数据集的主要特征

*1）数据集：*SEA 概念数据集用于[14]。数据集可在 http://www.liaad.up.pt/kdus/products/datasets-for-concept-drift 上获得，并通过概念漂移检测算法广泛用作测试平台。旋转超平面数据集由[15]创建。每个概念的数据集和特定 (k,t) 可在 http://www.win.tue.nl/~mpechen/data/DriftSets/获得。[16]中使用的 USENET1 和 USENET2 数据集可从 http://mlkd.csd.auth.gr/concepcdrift.html 获得。它们是来自不同新闻组（例如医学，空间，棒球）的消息流集合到用户。USENETI 和 USENET2 之间的差异是漂移的幅度。 USENETI 中的用户主题转移比 USENET2 中的用户更清晰。

以上所有数据集均为 $\{(\mathbf{X}_t, \mathrm{y_t})\}_{t=1}^{T}$ 的形式，其主要特征总结在表 I 中。其他细节，例如每个数据集的不平衡状态和漂移类型，可通过上述链接获得。

*2)评估：*在 SEA 概念数据集实验中，图 8 显示 LFR 在早期检测和较少的错误或延迟检测方面优于其他三种方法。

图 9 显示 LFR 在旋转超平面数据集实验中具有很好的性能。在第二个真正的漂移时间点，潜在的概念变化非常小。因此，所有检测算法都忽略了漂移。

在 USENET1 数据集实验中，图 10 表明 LFR 主导其他方法并且所有漂移点都被警告。同样，在 USENET2 数据集实验中，LFR 也优于其他方法，但检测延迟时延较长。从 USENETI 到 USENET2 的 LFR 优势的减少是由于概念漂移的幅度减小。

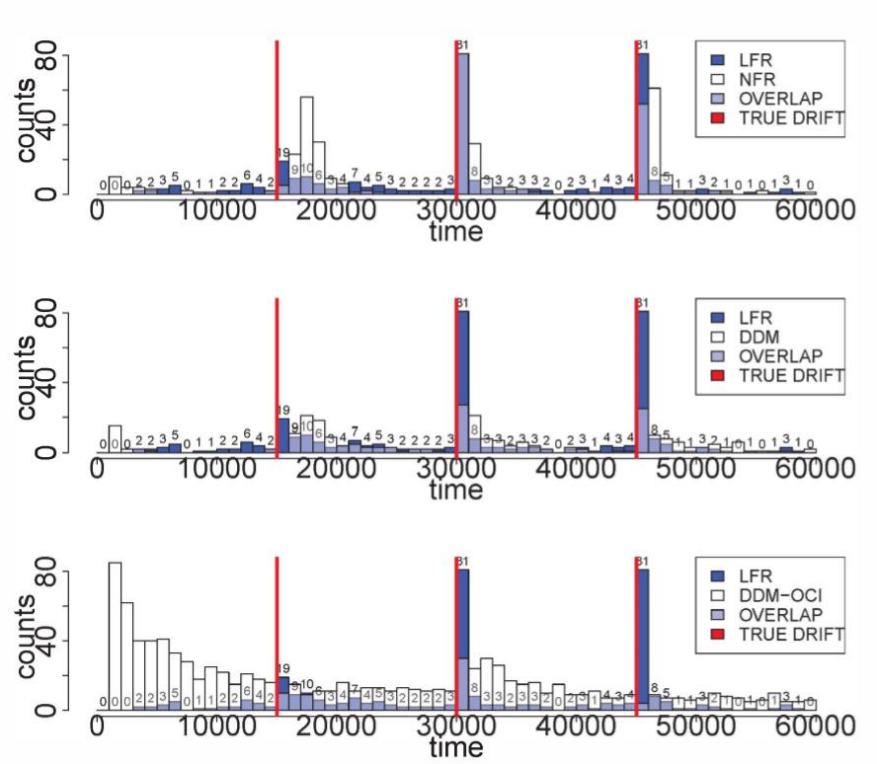

图 8.　比较 SEA 上检测时间的重叠直方图.

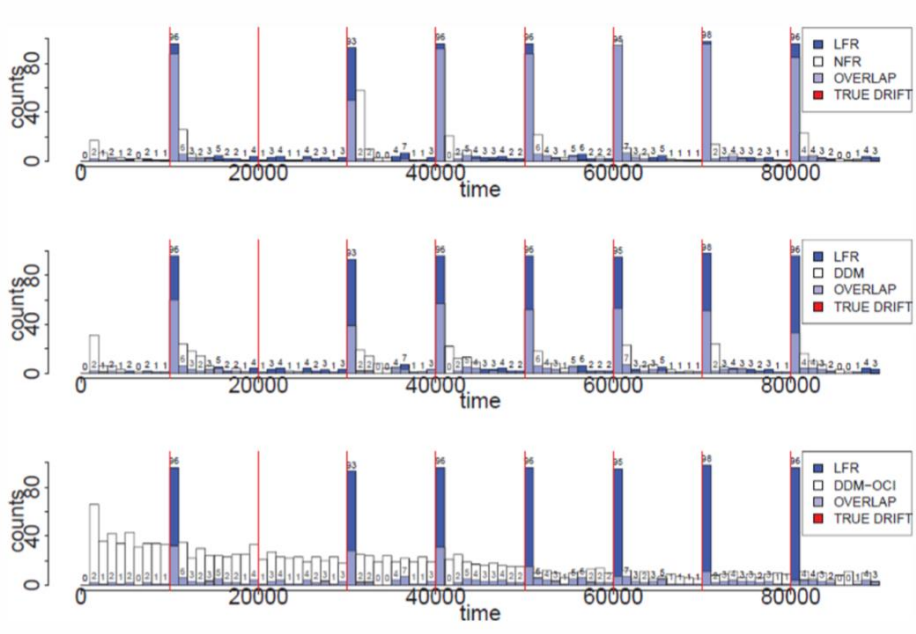

图 9.　比较 HYPERPLANE 上检测时间的重叠直方图.

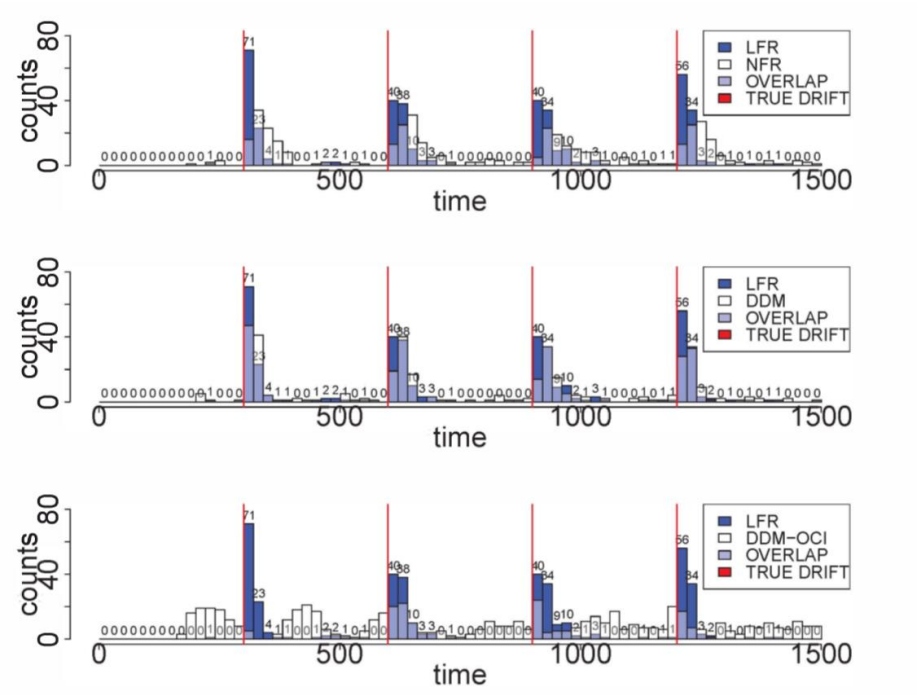

图 10.　比较 USENETI 检测时间的重叠直方图.

C. 摘要统计

通常，最佳算法将具有最小数量的错误警报和最大数量的早期检测，而较差的算法会产生大量错误警报，丢失或严重延迟的真实检测。表 II 和表 III 中提供了对于合成和公共数据集的错误检测期间的真实漂移时间,以及错误检测计数，正确检测计数的总结。

错误检测周期是指属于新概念的数据点之前的周期。对于 § IV-A 中的合成生成的数据集，存在跨越 T 个数据点的两个概念，使得错误检测时段被定义为 $[0，T/2)$。对于 § IV-B 中指定的数据集，如果有两个以上的概念，则错误检测周期对应于从中间概念到下一个真实漂移点的范围。直方图中的每个格子对应于 § IV-A 中的 200 个时间

步长和Ⅳ-B 中的数据集依赖性。由于在[17]，[18]中观察到误报警对预测性能的影响小于后期漂移检测，因此我们实验中的真实检测周期是指在接下来的 200 个时间步长（1 个格子）之后的时间段。§ IV-A 中的 $T/2$ 和§ IV-B 中每个真实漂移点之后跨越 1 个格子的时间段。检测算法的其他参数设置总结在表 IV 和表 V 中。特别选择它们以显示 LFR 的主导性能，即最小允许类型 I 误差，但是最大统计性能，超过基准算法。

如表 II 中所述，LFR 在回忆各种数据集中的真实变化点检测方面表现最佳。同样重要的是，通过产生最少量的错误检测和延迟检测，LFR 在检测变化点方面具有最高的精确度（表 III）。

Metric	LFR	NFR	DDM	DDM-OCI
Balance1	38	12	4	12
Balance2	16	3	0	11
Balance3	25	4	0	3
Imbalance1	95	59	0	4
Imbalance2	91	21	0	43
Imbalance3	95	38	36	39
SEA	142	29	17	26
HYPRPLN	671	598	345	149
USENET1	207	47	108	66
USENET2	3	17	3	21

表 II. 合成（公共）数据集正确检测到（多个）真正的漂移点的数量（之和）

Metric	LFR	NFR	DDM	DDM-OCI
Balance1	6	77	36	304
Balance2	13	19	33	339
Balance3	18	54	11	219
Imbalance1	18	81	16	259
Imbalance2	10	91	23	165
Imbalance3	9	86	55	204
SEA	72	32	54	658
HYPRPLN	84	56	73	826

USENET1	12	50	43	322
USENET2	43	80	65	272

表 III. 合成（公共）数据集错误检测到（多个）漂移点的数量（之和）

Parameters	Detect Sig	Warn Sig	Decay
LFR	$\epsilon_* = 1/100K$	$\delta_* = 1/100$	$\eta_* = 0.9$
NFR	$\epsilon_* = 1/1K$	$\delta_* = 0.025$	$\eta_* = 0.9$
DDM	$\alpha_{detect} = 3$	$\alpha_{warn} = 2$	$\eta_* = 0.9$
DDM-OCI	$\alpha_{detect} = 20$	$\alpha_{warn} = 10$	$\eta_* = 0.9$

表 IV. §IV-A 实验中的参数设置

Parameters	SEA	HYPRPLN	USENET1&2
LFR	$\epsilon_* = 1/10K$ $\delta_* = 1/100$	$\epsilon_* = 1/10K$ $\delta_* = 1/100$	$\epsilon_* = 1/10K$ $\delta_* = 1/100$
NFR	$\epsilon_* = 1/1K$ $\delta_* = 0.025$	$\epsilon_* = 1/1K$ $\delta_* = 0.025$	$\epsilon_* = 1/1K$ $\delta_* = 0.025$
DDM	$\alpha_{detect} = 3$ $\alpha_{warn} = 2$	$\alpha_{detect} = 3$ $\alpha_{warn} = 2$	$\alpha_{detect} = 3$ $\alpha_{warn} = 2$
DDM-OCI	$\alpha_{detect} = 20$ $\alpha_{warn} = 10$	$\alpha_{detect} = 20$ $\alpha_{warn} = 10$	$\alpha_{detect} = 20$ $\alpha_{warn} = 10$

表 V. §IV-B 实验中的参数设置

V. 结论

本文提出了概念漂移检测框架（LFR），用于检测概念漂移的发生并识别属于新概念的数据点。LFR 的多功能性允许它与批处理和流数据集，不平衡数据集一起使用，并且它使用直观易懂的用户指定参数，这与其他流行的概念漂移检测方法不同。LFR 在概念漂移的早期检测，高检测率和低误报率等方面明显优于现有的基准方法。

参考文献

[1] J. Gama, P. Medas, G. Castillo, and P. Rodrigues, "Learning with drift detection," in Advances in Artificial Intelligence-SBIA 2004. Springer, 2004, pp. 286-295.

[2] S. Wang, L. L. Minku, D. Ghezzi, D. Caltabiano, P. Tino, and X. Yao, "Concept drift detection for online class imbalance learning," in Neural Networks (]JCNN), The 2013 International Joint Conference on. IEEE, 2013, pp. 1-10.

[3] M. Baena-Garcia, J. del Campo-AVila, R. Fidalgo, A. Bifet, R. Gavaldil, and R. Morales-Bueno, "Early drift detection method," 2006.

[4] D. K. Antwi, H. L. Viktor, and N. Japkowicz, "The perfsim algorithm for concept drift detection in imbalanced data," in Data Mining Workshops (ICDMW), 2012 TEEE 12th Tnternational Conference on. IEEE, 2012, pp. 619-628.

[5] R. Klinkenberg and T. Joachims, "Detecting concept drift with support vector machines," in Proceedings of the Seventeenth International Conference on Machine Learning. Morgan Kaufmann Publishers Inc., 2000, pp. 487-494.

[6] D. S. Matteson and N. A. James, "A nonparametric approach for multiple change point analysis of multivariate data," Journal of the American Statistical Association, vol. 109, no. 505, pp. 334-345, 2014.

[7] X. Song, M. Wu, C. Jermaine, and S. Ranka, "Statistical change detection for multi-dimensional data," in Proceedings of the J 3th ACM STGKDD international conference on Knowledge discovery and data mining. ACM, 2007, pp. 667-676.

[8] A. Dries and U. RUckert, "Adaptive concept drift detection," Statistical Analysis and Data Mining, vol. 2, no. 5-6, pp. 311-327, 2009.

[9] L. A. Aroian and H. Levene, "The effectiveness of quality control charts," Journal of the American Statistical Association, vol. 45, no. 252, pp. 520-529, 1950.

[10] M. Basseville, I. Y. Nikiforov et al., Detection of abrupt changes: theory and application, vol. 104.

[11] S. Wang, L. L. Minku, and X. Yao, "A learning framework for online class imbalance learning," in Computational Tntelligence and Ensemble Learning (ClEL), 2013 IEEE Symposium on. IEEE, 2013, pp. 36-45.

[12] D. Bhati, P. Kgosi, and R. N. Rattihalli, "Distribution of geometrically weighted sum of bernoulli random variables," Applied Mathematics, vol. 2, p. 1382, 2011.

[13] D. Meyer and F. T. Wien, "Support vector machines," 2014.

[14] W. N. Street and Y. Kim, "A streaming ensssssemble algorithm (sea) for large-scale classification," in Proceedings of the seventh ACM STGKDD international conference on Knowledge discovery and data mining. ACM, 2001, pp. 377-382.

[15] W. Fan, "Systematic data selection to mine concept-drifting data streams," in Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2004, pp. 128-137.

[16] I. Katakis, G. Tsoumakas, and I. Vlahavas, "An ensemble of classifiers for coping with recurring contexts in data streams," in Proceedings of the 2008 conference on ECAT 2008: 18th European Conference on Artificial Tntelligence. lOS Press, 2008, pp. 763-764.

[17] C. Alippi, G. Boracchi, and M. Roveri, "Just-in-time classifiers for recurrent concepts." TEEE transactions on neural networks and learning systems, vol. 24, no. 4, pp. 620-634, 2013.

[18] L. L. Minku and X. Yao, "Ddd: A new ensemble approach for dealing with concept drift," Knowledge and Data Engineering, IEEE Transactions on, vol. 24, no. 4, pp. 619-633, 2012.