Request-and-Reverify: Hierarchical Hypothesis Testing for Concept Drift Detection with Expensive Labels

Shujian Yu[*1,2], **Xiaoyang Wang**[1], **José C. Príncipe**[2]
[1] Nokia Bell Labs, Murray Hill, NJ, USA
[2] Dept. of Electrical and Computer Engineering, University of Florida, Gainesville, FL, USA
yusjlcy9011@ufl.edu, xiaoyang.wang@nokia-bell-labs.com, principe@cnel.ufl.edu

Abstract

One important assumption underlying common classification models is the stationarity of the data. However, in real-world streaming applications, the data concept indicated by the joint distribution of feature and label is not stationary but drifting over time. Concept drift detection aims to detect such drifts and adapt the model so as to mitigate any deterioration in the model's predictive performance. Unfortunately, most existing concept drift detection methods rely on a strong and over-optimistic condition that the true labels are available immediately for all already classified instances. In this paper, a novel Hierarchical Hypothesis Testing framework with **Request-and-Reverify** strategy is developed to detect concept drifts by requesting labels only when necessary. Two methods, namely Hierarchical Hypothesis Testing with Classification Uncertainty (HHT-CU) and Hierarchical Hypothesis Testing with Attribute-wise "Goodness-of-fit" (HHT-AG), are proposed respectively under the novel framework. In experiments with benchmark datasets, our methods demonstrate overwhelming advantages over state-of-the-art unsupervised drift detectors. More importantly, our methods even outperform DDM (the widely used supervised drift detector) when we use significantly fewer labels.

1 Introduction

In the last decades, numerous efforts have been made in algorithms that can learn from data streams. Most traditional methods for this purpose assume the stationarity of the data. However, when the underlying source generating the data stream, i.e., the joint distribution $\mathbb{P}_t(\mathbf{X}, y)$, is not stationary, the optimal decision rule should change over time. This is a phenomena known as concept drift [Ditzler *et al.*, 2015; Krawczyk *et al.*, 2017]. Detecting such concept drifts is essential for the algorithm to adapt itself to the evolving data.

Concept drift can manifest two fundamental forms of changes from the Bayesian perspective [Kelly *et al.*, 1999]: 1) a change in the marginal probability $\mathbb{P}_t(\mathbf{X})$; 2) a change in the posterior probability $\mathbb{P}_t(y|\mathbf{X})$. Existing studies in this field primarily concentrate on detecting posterior distribution change $\mathbb{P}_t(y|\mathbf{X})$, also known as the real drift [Widmer and Kubat, 1993], as it clearly indicates the optimal decision rule. On the other hand, only a little work aims at detecting the virtual drift [Hoens *et al.*, 2012], which only affects $\mathbb{P}_t(\mathbf{X})$. In practice, one type of concept drift typically appears in combination with the other [Tsymbal, 2004]. Most methods for real drift detection assume that the true labels are available immediately after the classifier makes a prediction. However, this assumption is over-optimistic, since it could involve the annotation of data by expensive means in terms of cost and labor time. The virtual drift detection, though making no use of true label y_t, has the issue of wrong interpretation (i.e., interpreting a virtual drift as the real drift). Such wrong interpretation could provide wrong decision about classifier update which still require labeled data [Krawczyk *et al.*, 2017].

To address these issues simultaneously, we propose a novel Hierarchical Hypothesis Testing (HHT) framework with a **Request-and-Reverify** strategy for concept drift detection. HHT incorporates two layers of hypothesis tests. Different from the existing HHT methods [Alippi *et al.*, 2017; Yu and Abraham, 2017], our HHT framework is the first attempt to use labels for concept drift detection **only when necessary**. It ensures that the test statistic (derived in a fully unsupervised manner) in Layer-I captures the most important properties of the underlying distributions, and adjusts itself well in a more powerful yet conservative manner that only requires labeled data when necessary in Layer-II. Two methods, namely Hierarchical Hypothesis Testing with Classification Uncertainty (HHT-CU) and Hierarchical Hypothesis Testing with Attribute-wise "Goodness-of-fit" (HHT-AG), are proposed under this framework in this paper. The first method incrementally tracks the distribution change with the defined *classification uncertainty* measurement in Layer-I, and uses permutation test in Layer-II, whereas the second method uses the standard Kolmogorov-Smirnov (KS) test in Layer-I and two-dimensional (2D) KS test [Peacock, 1983] in Layer-II. We test both proposed methods in benchmark datasets. Our methods demonstrate overwhelming advantages over state-of-the-art unsupervised methods. Moreover, though using significantly fewer labels, our methods outperform supervised methods like DDM [Gama *et al.*, 2004].

[*]This work was done when Shujian Yu was a research intern at Nokia Bell Labs, Murray Hill, NJ, USA.

2 Background Knowledge

2.1 Problem Formulation

Given a continuous stream of labeled samples $\{\mathbf{X}_t, y_t\}$, $t = 1, 2, ..., T$, a classification model $\hat{f}$ can be learned so that $\hat{f}(\mathbf{X}_t) \mapsto y_t$. Here, $\mathbf{X}_t \in \mathbb{R}^d$ represents a d-dimensional feature vector, and y_t is a discrete class label. Let $(\mathbf{X}_{T+1}, \mathbf{X}_{T+2}, ..., \mathbf{X}_{T+N})$ be a sequence of new samples that comes chronologically with unknown labels. At time $T + N$, we split the samples in a set $S_A = (\mathbf{X}_{T+N-n_A+1}, \mathbf{X}_{T+N-n_A+2}, ..., \mathbf{X}_{T+N})$ of n_A recent ones and a set $S_B = (\mathbf{X}_{T+1}, \mathbf{X}_{T+2}, ..., \mathbf{X}_{T+N-n_A})$ containing the $(N - n_A)$ samples that appear prior to those in S_A. The problem of *concept drift detection* is identifying whether or not the source $\mathscr{P}$ (i.e., the joint distribution $\mathbb{P}_t(\mathbf{X}, y)$[1]) that generates samples in S_A is the same as that in S_B (even without access to the true labels y_t) [Ditzler *et al.*, 2015; Krawczyk *et al.*, 2017]. Once such a drift is found, the machine can request a window of labeled data to update $\hat{f}$ and employ the new classifier to predict labels of incoming data.

2.2 Related Work

The techniques for concept drift detection can be divided into two categories depending on reliance of labels [Sethi and Kantardzic, 2017]: supervised (or explicit) drift detectors and unsupervised (or implicit) drift detectors. **Supervised Drift Detectors** rely heavily on true labels, as they typically monitor one error metrics associated with classification loss. Although much progress has been made on concept drift detection in the supervised manner, its assumption that the ground truth labels are available immediately for all already classified instances is typically over-optimistic. **Unsupervised Drift Detectors**, on the other hand, explore to detect concept drifts without using true labels. Most unsupervised concept drift detection methods concentrate on performing multivariate statistical tests to detect the changes of feature values $\mathbb{P}_t(\mathbf{X})$, such as the Conjunctive Normal Form (CNF) density estimation test [Dries and Rückert, 2009] and the Hellinger distance based density estimation test [Ditzler and Polikar, 2011]. Considering their high computational complexity, an alternative approach is to conduct univariate test on each attribute of features independently. For example, [Reis *et al.*, 2016] develops an incremental (sequential) KS test which can achieve exactly the same performance as the conventional batch-based KS test.

Besides modeling virtual drifts of $\mathbb{P}_t(\mathbf{X})$, recent research in unsupervised drift detection attempts to model the real drifts by monitoring the classifier output $\hat{y}_t$ or posterior probability as an alternative to y_t. The Confidence Distribution Batch Detection (CDBD) approach [Lindstrom *et al.*, 2011] uses Kullback-Leibler (KL) divergence to compare the classifier output values from two batches. A drift is signaled if the divergence exceeds a threshold. This work is extended in [Kim and Park, 2017] by substituting the classifier output value with the classifier confidence measurement. Another representative method is the Margin Density Drift Detection (MD3) algorithm [Sethi and Kantardzic, 2017], which tracks the proportion of samples that are within a classifier (i.e., SVM) margin and uses an active learning strategy in [Žliobaite *et al.*, 2014] to interactively query the information source to obtain true labels. Though not requiring true labels for concept drift detection, the major drawback of these unsupervised drift detectors is that they are prone to false positives as it is difficult to distinguish noise from distribution changes. Moreover, the wrong interpretation of virtual drifts could cause wrong decision for classifier update which require not only more labeled data but also unnecessary classifier re-training [Krawczyk *et al.*, 2017].

[1]The distributions are deliberated subscripted with time index t to explicitly emphasize their time-varying characteristics.

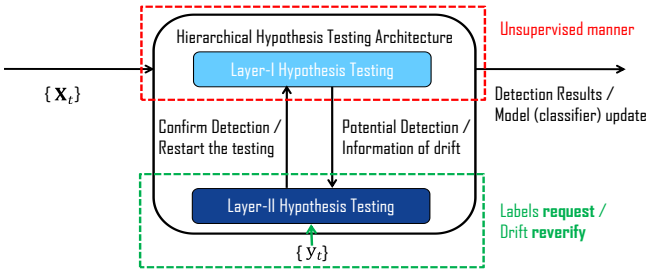

Figure 1: The Request-and-Reverify Hierarchical Hypothesis Testing framework for concept drift detection with expensive labels.

3 Request-and-Reverify HHT Approach

The observations on the existing supervised and unsupervised concept drift detection methods motivate us to propose the Request-and-Reverify Hierarchial Hypothesis Testing framework (see Fig. 1). Specifically, our layer-I test is operated in a fully unsupervised manner that does not require any labels. Once a potential drift is signaled by Layer-I, the Layer-II test is activated to confirm (or deny) the validity of the suspected drift. The result of the Layer-II is fed back to the Layer-I to reconfigure or restart Layer-I once needed.

In this way, the upper bound of HHT's Type-I error is determined by the significance level of its Layer-I test, whereas the lower bound of HHT's Type-II error is determined by the power of its Layer-I test. Our Layer-I test (and most existing single layer concept drift detectors) has low Type-II error (i.e., is able to accurately detect concept drifts), but has relatively higher Type-I error (i.e., is prone to generate false alarms). The incorporation of the Layer-II test is supposed to reduce false alarms, thus decreasing the Type-I error. The cost is that the Type-II error could be increased at the same time. In our work, we request true labels to conduct a more precise Layer-II test, so that we can significantly decrease the Type-I error with minimum increase in the Type-II error.

3.1 HHT with Classification Uncertainty (HHT-CU)

Our first method, HHT-CU, detects concept drift by tracking the *classification uncertainty* measurement $u_t = \|\hat{y}_t - \hat{\mathbb{P}}(y_t|\mathbf{X}_t)\|_2$, where $\|\cdot\|_2$ denotes the ℓ_2 distance, $\hat{\mathbb{P}}(y_t|\mathbf{X}_t)$ is the posterior probability estimated by the classifier at time index t, and $\hat{y}_t$ is the target label encoded from $\hat{\mathbb{P}}(y_t|\mathbf{X}_t)$ using the 1-of-K coding scheme [Bishop, 2006]. Intuitively, the distance between $\hat{y}_t$ and $\hat{\mathbb{P}}(y_t|\mathbf{X}_t)$ measures the

classification uncertainty for the current classifier, and the statistic derived from this measurement should be stationary (i.e., no "significant" distribution change) in a stable concept. Therefore, the dramatic change of the uncertainty mean value may suggest a potential concept drift.

Different from the existing work that typically monitors the derived statistic with the three-sigma rule in statistical process control [Montgomery, 2009], we use the Hoeffding's inequality [Hoeffding, 1963] to monitor the moving average of u_t in our Layer-I test.

Theorem 1 (Hoeffding's inequality) *Let X_1, X_2,..., X_n be independent random variables such that $X_i \in [a_i, b_i]$, and let $\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$, then for $\varepsilon \geq 0$:*

$$\mathbb{P}\{\bar{X} - \mathbb{E}(\bar{X}) \geq \varepsilon\} \leq e^{\frac{-2n^2\varepsilon^2}{\sum_{i=1}^{n}(b_i - a_i)^2}}. \quad (1)$$

where $\mathbb{E}$ denotes the expectation. Using this theorem, given a specific significance level α, the error ε_α can be computed as:

$$\varepsilon_\alpha = \sqrt{\frac{1}{2n}\ln\frac{1}{\alpha}}. \quad (2)$$

The Hoeffding's inequality does not require an assumption on the probabilistic distribution of u_t. This makes it well suited in learning from real data streams [Frías-Blanco *et al.*, 2015]. Moreover, the Corollary 1.1 proposed by Hoeffding [Hoeffding, 1963] can be directly applied to detect significant changes in the moving average of streaming values.

Corollary 1.1 (Layer-I test of HHT-CU) *If X_1, X_2, ..., X_n, X_{n+1}, ..., X_{n+m} be independent random variables with values in the interval $[a, b]$, and if $\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$ and $\bar{Z} = \frac{1}{n+m}\sum_{i=1}^{n+m} X_i$, then for $\varepsilon \geq 0$:*

$$\mathbb{P}\{\bar{X} - \bar{Z} - (\mathbb{E}(\bar{X}) - \mathbb{E}(\bar{Z})) \geq \varepsilon\} \leq e^{\frac{-2n(n+m)\varepsilon^2}{m(b-a)^2}}. \quad (3)$$

By definition, $u_t \in [0, \sqrt{\frac{K-1}{K}}]$, where K is the number of classes. $\bar{X}$ denotes the *classification uncertainty* moving average before a cutoff point, and $\bar{Z}$ denotes the moving average over the whole sequence. The rule to reject the null hypothesis $H_0 : \mathbb{E}(\bar{X}) > \mathbb{E}(\bar{Z})$ against the alternative one $H_1 : \mathbb{E}(\bar{X}) \leq \mathbb{E}(\bar{Z})$ at the significance level α will be $\bar{Z} - \bar{X} \geq \varepsilon_\alpha$, where

$$\varepsilon_\alpha = \sqrt{\frac{K-1}{K}} \times \sqrt{\frac{m}{2n(n+m)}\ln\frac{1}{\alpha}}. \quad (4)$$

Regarding the cutoff point, a reliable location can be estimated from the minimum value of $\bar{X}_i + \varepsilon_{\bar{X}_i}$ ($1 \leq i \leq n+m$) [Gama *et al.*, 2004; Frías-Blanco *et al.*, 2015]. This is because $\bar{X}_i$ keeps approximately constant in a stable concept, thus $\bar{X}_i + \varepsilon_{\bar{X}_i}$ must reduce its value correspondingly.

The Layer-II test aims to reduce false positives signaled by Layer-I. Here, we use the permutation test which is described in [Yu and Abraham, 2017]. Different from [Yu and Abraham, 2017], which trains only one classifier f_{ord} using S_{ord} and evaluates it on S'_{ord} to get a zero-one loss $\hat{E}_{ord}$, we train another classifier f'_{ord} using S'_{ord} and evaluate it on S_{ord} to get another zero-one loss $\hat{E}'_{ord}$. We reject the null hypothesis if either $\hat{E}_{ord}$ or $\hat{E}'_{ord}$ deviates too much from the prediction loss of the shuffled splits. The proposed HHT-CU is summarized in Algorithm 1, where the window size N is set as the number of labeled samples to train the initial classifier $\hat{f}$.

Algorithm 1 HHT with Classification Uncertainty (HHT-CU)

Input: Unlabeled stream $\{\mathbf{X}_t\}_{t=0}^{\infty}$ where $\mathbf{X}_t \in \mathbb{R}^d$; Initially trained classifier $\hat{f}$; Layer-I significance level Θ_1; Layer-II significance level Θ_2; Window size N.
Output: Detected drift time index $\{T_{cd}\}$; Potential drift time index $\{T_{pot}\}$.

variables declaration
  $\bar{X}_{cut}$: moving average of u_1, u_2, ..., u_{cut};
  $\bar{Z}_n$: moving average of u_1, u_2, ..., u_n;
  $\varepsilon_{\bar{X}_{cut}}$ and $\varepsilon_{\bar{Z}_n}$: error bounds computed using Eqs. (2) and (4) respectively;
end variables declaration
$\{T_{cd}\} = \phi$; $\{T_{pot}\} = \phi$;
for $t = 1$ to ∞ **do**
  Compute u_t using $\hat{f}$;
  Update $\bar{Z}_t$ and $\varepsilon_{\bar{Z}_t}$ by adding u_t;
  if $\bar{Z}_t + \varepsilon_{\bar{Z}_t} \leq \bar{X}_{cut} + \varepsilon_{\bar{X}_{cut}}$ **then**
    $\bar{X}_{cut} = \bar{Z}_t$; $\varepsilon_{\bar{X}_{cut}} = \varepsilon_{\bar{Z}_t}$;
  end if
  if $H_0 : \mathbb{E}(\bar{X}_{cut}) \geq \mathbb{E}(\bar{Z}_t)$ is rejected at Θ_1 **then**
    $\{T_{pot}\} \leftarrow t$;
    Request $2N$ labeled samples $\{\mathbf{X}_i, y_i\}_{i=t-N}^{t+N-1}$;
    Perform Layer-II test using $\{\mathbf{X}_i, y_i\}_{i=t-N}^{t+N-1}$ at Θ_2;
    if (Layer-II confirms the potentiality of t) **then**
      $\{T_{cd}\} \leftarrow t$;
      Update $\hat{f}$ using $\{\mathbf{X}_i, y_i\}_{i=t}^{t+N-1}$;
      Initialize declared variables;
    else
      Discard t;
      Restart Layer-I test;
    end if
  end if
end for

3.2 HHT with Attribute-wise "Goodness of fit" (HHT-AG))

The general idea behind HHT-AG is to explicitly model $\mathbb{P}_t(\mathbf{X}, y)$ with limited access to y. To this end, a feasible solution is to detect potential drift points in Layer-I by just modeling $\mathbb{P}_t(\mathbf{X})$, and then require limited labeled data to confirm (or deny) the suspected time index in Layer-II.

The Layer-I test of HHT-AG conducts "Goodness-of-fit" test on each attribute $x^k|_{k=1}^d$ individually to determine whether $\mathbf{X}$ from two windows differ: a baseline (or reference) window W_1 containing the first N items of the stream that occur after the last detected change; and a sliding window W_2 containing N items that follow W_1. We slide the W_2 one step forward whenever a new item appears on the stream. A potential concept drift is signaled if at least for one attribute there is a distribution change. Factoring $\mathbb{P}_t(\mathbf{X})$ into $\prod_{k=1}^{d}\mathbb{P}_t(x^k)$ for multivariate change detection is initially proposed in [Kifer *et al.*, 2004]. Since then, this factorization strategy becomes widely used [Žliobaite, 2010;

Reis *et al.*, 2016]. Sadly, no existing work provides a theoretical foundation of this factorization strategy. In our perspective, one possible explanation is the Sklar's Theorem [Sklar, 1959], which states that if $\mathbb{H}$ is a d-dimensional joint distribution function and if $\mathbb{F}_1$, $\mathbb{F}_2$, ..., $\mathbb{F}_d$ are its corresponding marginal distribution functions, then there exists a d-copula C: $[0,1]^d \rightarrow [0,1]$ such that:

$$\mathbb{H}(\mathbf{X}) = C(\mathbb{F}_1(x^1), \mathbb{F}_2(x^2), ..., \mathbb{F}_d(x^d)). \tag{5}$$

The density function (if exists) can thus be represented as:

$$\mathbb{P}(\mathbf{X}) = c(\mathbb{F}_1(x^1), \mathbb{F}_2(x^2), ..., \mathbb{F}_d(x^d)) \prod_{k=1}^{d} \mathbb{P}(x^k) \propto \prod_{k=1}^{d} \mathbb{P}(x^k),$$

where c is the density of the copula C.

Though Sklar does not show practical ways on how to calculate C, this Theorem demonstrates that if $\mathbb{P}(\mathbf{X})$ changes, we can infer that one of $\mathbb{P}(x_i)$ should also changes; otherwise, if none of the $\mathbb{P}(x_i)$ changes, the $\mathbb{P}(\mathbf{X})$ would not be likely to change.

This paper selects Kolmogorov-Smirnov (KS) test to measure the discrepancy of $\mathbb{P}_t(x^k)|_{k=1}^d$ in two windows. Specifically, the KS test rejects the null hypothesis, i.e., the observations in sets $\mathbf{A}$ and $\mathbf{B}$ originate from the same distribution, at significance level α if the following inequality holds:

$$\sup_x |\mathbb{F}_{\mathbf{A}}(x) - \mathbb{F}_{\mathbf{B}}(x)| > s(\alpha)\sqrt{\frac{m+n}{mn}}, \tag{6}$$

where $\mathbb{F}_{\mathbf{C}}(x) = \frac{1}{|\mathbf{C}|}\sum \mathbf{1}_{\{c \in \mathbf{C}, c \leq x\}}$ denotes the empirical distribution function (an estimation to the cumulative distribution function $\mathbb{P}(X < x)$), $s(\alpha)$ is a α-specific value that can be retrieved from a known table, m and n are the cardinality of set $\mathbf{A}$ and set $\mathbf{B}$ respectively.

We then validate the potential drift points by requiring true labels of data that come from W_1 and W_2 in Layer-II. The Layer-II test of HHT-AG makes the conditionally independent factor assumption [Bishop, 2006] (a.k.a. the "naive Bayes" assumption), i.e., $\mathbb{P}(x^i|x^j, y) = \mathbb{P}(x^i|y)$ ($1 \leq i \neq j \leq d$). Thus, the joint distribution $\mathbb{P}_t(\mathbf{X}, y)$ can be represented as:

$$\begin{aligned}\mathbb{P}_t(\mathbf{X}, y) &= \mathbb{P}_t(y)\mathbb{P}_t(x^d|y)\mathbb{P}_t(x^{d-1}|x^d, y)...\mathbb{P}_t(x^1|x^2, ..., x^d, y) \\ &\propto \mathbb{P}_t(y)\prod_{k=1}^{d}\mathbb{P}_t(x^k|y) \propto \prod_{k=1}^{d}\mathbb{P}_t(x^k, y).\end{aligned} \tag{7}$$

According to Eq. (7), we perform d independent two-dimensional (2D) KS tests [Peacock, 1983] on each bivariate distribution $\mathbb{P}_t(x^k, y)|_{k=1}^d$ individually. The 2D KS test is a generalization of KS test on 2D plane. Although the cumulative probability distribution is not well-defined in more than one dimension, Peacock's insight is that a good surrogate is the integrated probability in each of the four quadrants for a given point (x, y), i.e., $\mathbb{P}(X \leq x, Y \leq y)$, $\mathbb{P}(X \leq x, Y \geq y)$, $\mathbb{P}(X \geq x, Y \leq y)$ and $\mathbb{P}(X \geq x, Y \geq y)$. Similarly, a potential drift is confirmed if the 2D KS test rejects the null hypothesis for at least one of the d bivariate distributions. HHT-AG is summarized in Algorithm 2, where the window size N is set as the number of labeled samples to train the initial classifier $\hat{f}$.

Algorithm 2 HHT with Attribute-wise Goodness of fit (HHT-AG)

Input: Unlabeled stream $\{\mathbf{X}_t\}_{t=0}^{\infty}$ where $\mathbf{X}_t \in \mathbb{R}^d$; Significance level Θ_1; Significance level Θ_2 ($= \Theta_1$ by default); Window size N.
Output: Detected drift time index $\{T_{cd}\}$; Potential drift time index $\{T_{pot}\}$.

for $i = 1$ to d **do**
  $c_0 \leftarrow 0$;
  $\mathrm{W}_{1,i} \leftarrow$first N points in x^i from time c_0;
  $\mathrm{W}_{2,i} \leftarrow$next N points in x^i in stream;
end for
while not end of stream **do**
  for $i = 1$ to d **do**
    Slide $\mathrm{W}_{2,i}$ by 1 point;
    Perform KS test with Θ_1 on $\mathrm{W}_{1,i}$ and $\mathrm{W}_{2,i}$;
    if (KS test rejects the null hypothesis) **then**
      $\{T_{pot}\} \leftarrow$current time;
      $\mathrm{W}_1 \leftarrow$first N tuples in (x^i, y) from time c_0;
      $\mathrm{W}_2 \leftarrow$next N tuples in (x^i, y) in stream;
      Perform 2D KS test with Θ_2 on W_1 and W_2;
      if (2D KS test rejects the null hypothesis) **then**
        $c_0 \leftarrow$current time;
        $\{T_{cd}\} \leftarrow$current time;
        Clear all windows and **GOTO** Step 1;
      end if
    end if
  end for
end while

4 Experiments

Two sets of experiments are performed to evaluate the performance of HHT-CU and HHT-AG. First, quantitative metrics and plots are presented to demonstrate HHT-CU and HHT-AG's effectiveness and superiority over state-of-the-art approaches on benchmark synthetic data. Then, we validate, via three real-world applications, the effectiveness of the proposed HHT-CU and HHT-AG on streaming data classification and the accuracy of its detected concept drift points. This paper selects soft margin SVM as the baseline classifier because of its accuracy and robustness.

4.1 Experimental Setup

We compare the results with three baseline methods, three topline supervised methods, and two state-of-the-art unsupervised methods for concept drift detection. The first two baselines, DDM [Gama *et al.*, 2004] and EDDM [Baena-García *et al.*, 2006], are the most popular supervised drift detector. The third one, we refer to as Attribute-wise KS test (A-KS) [Žliobaite, 2010; Reis *et al.*, 2016], is a benchmark unsupervised drift detector that has been proved effective in real applications. Note that, A-KS is equivalent to the Layer-I test of HHT-AG. The toplines selected for comparison are LFR [Wang and Abraham, 2015], HLFR [Yu and Abraham, 2017] and HDDM [Frías-Blanco *et al.*, 2015]. HLFR is the first method on concept drift detection with HHT framework, whereas HDDM introduces Hoeffding's inequality on concept drift detection. All of these methods are operated in supervised manner and significantly outperform DDM. However, LFR and HLFR can only support binary classification. In addition, we also compare with MD3 [Sethi and

Kantardzic, 2017] and CDBD [Lindstrom *et al.*, 2011], the state-of-the-art concept drift detectors that attempt to model $\mathbb{P}_t(y|\mathbf{X})$ without access to y. We use the parameters recommended in the papers for each competing method. The detailed values on significance levels or thresholds (if there exist) are shown in Table 1.

Algorithms	Significance levels (or thresholds)
HHT-CU	$\Theta_1 = 0.01, \Theta_2 = 0.01$
HHT-AG	$\Theta_1 = 0.001, \Theta_2 = 0.001$
A-KS	$\Theta = 0.001$
MD3	$\Theta = 3$
HLFR	$\delta_\star = 0.01, \epsilon_\star = 0.00001, \eta = 0.01$
LFR	$\delta_\star = 0.01, \epsilon_\star = 0.00001$
DDM	$\alpha = 3, \beta = 2$
EDDM	$\alpha = 0.95, \beta = 0.9$
HDDM	$\alpha_W = 0.005, \alpha_D = 0.001$

Table 1: Parameter settings for all competing algorithms.

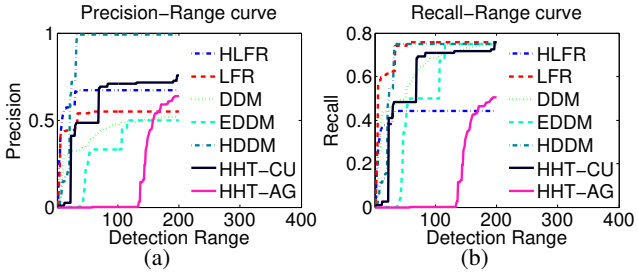

Figure 2: Concept drift detection evaluation using (a) the **Precision-Range** (PR) curve and (b) the **Recall-Range** (RR) curve on UG-2C-2D dataset over 100 Monte-Carlo trails. The X-axis represents the predefined detection delay range, whereas the Y-axis denotes the corresponding Precision or Recall values. For a specific delay range, a higher **Precision** or **Recall** value suggests better performance. This figure shows our methods and their supervised counterparts.

4.2 Results on Benchmark Synthetic Data

We first compare the performance of the HHT-CU and HHT-AG against aforementioned concept drift approaches on benchmark synthetic data. Eight datasets are selected from [Souza *et al.*, 2015; Dyer *et al.*, 2014], namely 2CDT, 2CHT, UG-2C-2D, MG-2C-2D, 4CR, 4CRE-V1, 4CE1CF, 5CVT. Among them, 2CDT, 2CHT, UG-2C-2D and MG-2C-2D are binary-class datasets, while 4CR, 4CRE-V1, 4CE1CF and 5CVT have multiple classes. To facilitate detection evaluation, we cluster each dataset into 5 segments to introduce 4 abrupt drift points, thus controlling ground truth drift points and allowing precise quantitative analysis. Quantitative comparison is performed by evaluating detection quality. To this end, the True Positive (*TP*) detection is defined as a detection within a fixed delay range after the precise concept change time. The False Negative (*FN*) is defined as missing a detection within the delay range, and the False Positive (*FP*) is defined as a detection outside the delay range range or an extra detection in the range. The detection quality is measured jointly with Precision, Recall and delay detection using **Precision-Range** curve and **Recall-Range** curve respectively (see Fig. 2 for an example), where $\textbf{Precision} = TP/(TP + FP)$, and $\textbf{Recall} = TP/(TP + FN)$.

For a straightforward comparison, Table 2 reports the number of required labeled samples (in percentage) for each algorithm, whereas Table 3 summarizes the Normalized Area Under the Curve (NAUC) values for two kinds of curves. As can be seen, HLFR and LFR can provide the most accurate detection as expected. However, they are only applicable for binary-class datasets and require true labels for the entire data stream. Our proposed HHT-CU and HHT-AG, although slightly inferior to HLFR or LFR, can strike the best tradeoff between detection accuracy and the portion of required labels, especially considering the overwhelming advantage over MD3 and CDBD that are the most relevant counterparts. Although the detection module of MD3 and CDBD are operated in fully unsupervised manner, they either fail to provide reliable detection results or generate too much false positives which may, by contrast, require even more true labels (for classifier update). Meanwhile, it is very encouraging to find that HHT-CU can achieve comparable or even better results than DDM (i.e., the most popular supervised drift detector) with significantly fewer labels. This suggests that our *classification uncertainty* is as sensitive as the total classification accuracy in DDM to monitor the nonstationary environment. And, we can see HHT-AG can significantly improve the Precision value compared to A-KS. This suggests the effectiveness of Layer-II test on reverifying the validity of suspected drifts and denying false alarms. In addition, in the extreme cases when $\mathbb{P}(\mathbf{X})$ remains unchanged but $\mathbb{P}(y|\mathbf{X})$ does change, our methods (and the state-of-the-art unsupervised methods) are not able to detect the concept drift which is the change of the joint distribution $\mathbb{P}(\mathbf{X}, y)$. This limitation is demonstrated in our experiments on the synthetic 4CR dataset where $\mathbb{P}(\mathbf{X})$ remains the same.

	HHT-CU	HHT-AG	A-KS	MD3	CDBD
2CDT	28.97	96.58	78.29	13.69	96.32
2CHT	28.12	98.01	77.44	11.71	93.19
UG-2C-2D	28.43	37.37	18.68	31.21	88.02
MG-2C-2D	25.51	45.46	21.36	11.83	83.58
4CR	0	0	0		
4CRE-V1	29.15	40.44	20.22		
4CR1CF	12.69	33.64	8.33		
5CVT	34.65	35.98	44.45		

Table 2: Averaged number of required labeled samples in the testing set (%) for all competing algorithms. The performances of supervised detectors (HLFR, LFR, DDM, EDDM, and HDDM) are omitted because they require all the true labels (i.e., 100%). Also, MD3 and CDBD cannot be applied to multi-class datasets including 4CR, 4CRE-V1, 4CR1CF, and 5CVT. The least and second least labeled samples used for each dataset are marked with red and blue respectively. Our methods and A-KS do not detect any drifts on 4CR, and thus they use "0" labeled samples for 4CR. MD3 in many cases uses the least labels, but its detection accuracy is the worst as in Table 3.

	Our methods		**Unsupervised methods**			**Supervised methods**				
	HHT-CU	HHT-AG	A-KS	MD3	CDBD	HLFR	LFR	DDM	EDDM	HDDM
2CDT	0.92/0.92	0.15/0.48	0.13/0.62	0.02/0.01	0.08/0.85	0.82/0.79	0.81/0.80	0.79/0.79	0.77/0.77	0.91/0.91
2CHT	0.86/0.86	0.15/0.48	0.15/0.49	0.02/0.01	0.09/0.91	0.93/0.93	0.93/0.93	0.60/0.60	0.89/0.89	0.89/0.89
UG-2C-2D	0.58/0.58	0.16/0.13	0.08/0.05	0.01/0.07	0.04/0.87	0.64/0.42	0.52/0.72	0.43/0.62	0.33/0.49	0.88/0.67
MG-2C-2D	0.52/0.52	0.26/0.49	0.21/0.43	0.05/0.16	0.02/0.80	0.74/0.74	0.46/0.91	0.37/0.60	0.34/0.73	0.68/0.73
4CR	−/0	−/0	−/0					0.94/0.94	0.86/0.86	0.98/0.98
4CRE-V1	0.78/0.78	0.21/0.21	0.19/0.21					0.20/0.22	0.84/0.84	0.98/0.98
4CR1CF	0.49/0.49	0.66/0.86	0.43/0.45					0.10/0.50	0.35/0.63	0.89/0.89
5CVT	0.53/0.73	0.16/0.75	0.16/0.84					0.43/0.73	0.28/0.65	0.35/0.41

Table 3: Averaged Normalized Area Under the Curve (NAUC) values for **Precision-Range** curve (left side of the forward slash) and **Recall-Range** curve (right side of the forward slash) of all competing algorithms. A higher value indicates better performance. The best three results are marked with red, blue and green respectively. "-" denotes no concept drift is detected. MD3, CDBD, HLFR and LFR cannot be applied to multi-class datasets including 4CR, 4CRE-V1, 4CR1CF, and 5CVT. In general, we can see the proposed methods overwhelmingly outperform the unsupervised methods, and achieve similar performances of the supervised methods. In addition, attention should be paid on 4CR, in which only DDM, EDDM, and HDDM can provide satisfactory detection results. This suggests that purely monitoring *classification uncertainty* or modeling the marginal distribution $\mathbb{P}(\mathbf{X})$ become invalid when there is no change on $\mathbb{P}(\mathbf{X})$. In this case, sufficient ground truth labels are the prerequisite for reliable detection.

4.3 Results on Real-world Data

In this section, we evaluate algorithm performance on real-world streaming data classification in a non-stationary environment. Three widely used real-world datasets are selected, namely **USENET1** [Katakis *et al.*, 2008], **Keystroke** [Souza *et al.*, 2015] and **Posture** [Kaluža *et al.*, 2010]. The descriptions on these three datasets are available in [Yu and Abraham, 2017; Reis *et al.*, 2016]. For each dataset, we also select the same number of labeled instances to train the initial classifier as suggested in [Yu and Abraham, 2017; Reis *et al.*, 2016].

The concept drift detection results and streaming classification results are summarized in Table 4. We measure the cumulative classification accuracy and the portion of required labels to evaluate prediction quality. Since the classes are balanced, the classification accuracy is also a good indicator. In these experiments, our proposed HHT-CU and HHT-AG always feature significantly less amount of false positives, while maintaining good true positive rate for concept drift detection. This suggests the effectiveness of the proposed hierarchical architecture on concept drift reverification. The HHT-CU can achieve overall the best performance in terms of accurate drift detection, streaming classification, as well as the rational utilization of labeled data.

(a) USENET1

	Precision	Recall	Delay	Accuracy	Labels
HHT-CU	**1.00**	**1.00**	13.25	**85**	**30.77**
HHT-AG	-	0	-	57	0
A-KS	-	0	-	57	0
MD3	0.14	0.25	16	76	71.85
CDBD	0.10	0.75	**3.33**	82	91.15
HLFR	0.75	0.75	11.67	84	100
LFR	0.75	0.75	11.67	84	100
DDM	0.75	0.75	18.33	83	100
EDDM	**1.00**	**1.00**	57.25	81	100
HDDM	**1.00**	**1.00**	17.75	83	100
NA	-	0	-	57	0

(b) Keystroke

	Precision	Recall	Delay	Accuracy	Labels
HHT-CU	**1.00**	**0.14**	1.5	**88**	**14.29**
HHT-AG	0.5	**0.14**	**1**	81	57.11
A-KS	0.25	**0.14**	**1**	79	52.43
DDM	-	0	-	67	100
EDDM	0.33	0	-	68	100
HDDM	**1.00**	**0.14**	**1**	86	100
NA	-	0	-	56	0

(c) Posture

	Precision	Recall	Delay	Accuracy	Labels
HHT-CU	**1.00**	**1.00**	2421.8	**56**	14.60
HHT-AG	**1.00**	**1.00**	**406**	55	17.97
A-KS	**1.00**	**1.00**	**406**	55	**10.54**
DDM	0.75	0.75	3318.67	54	100
EDDM	0.75	0.75	1253.4	54	100
HDDM	**1.00**	**1.00**	689.25	**56**	100
NA	-	0	-	46	0

Table 4: Quantitative metrics on real-world applications. The **Precision**, **Recall** and **Delay** denote the concept drift detection precision value, recall value and detection delay, whereas the **Accuracy** and **Labels** denote the cumulative classification accuracy and required portion of true labels in the testing set (%). "-" denotes no concept drift is detected or the detected drift points all are false alarms. "NA": using initial classifier without any update.

5 Conclusion

This paper presents a novel Hierarchical Hypothesis Testing (HHT) framework with a **Request-and-Reverify** strategy to detect concept drifts. Two methods, namely HHT with Classification Uncertainty (HHT-CU) and HHT with Attribute-wise "Goodness-of-fit" (HHT-AG), are proposed respectively under this framework. Our methods significantly outperform the state-of-the-art unsupervised counterparts, and are even comparable or superior to the popular supervised methods with significantly fewer labels. The results indicate our progress on using far fewer labels to perform accurate concept drift detection. The HHT framework is highly effective in deciding label requests and validating detection candidates.

References

[Alippi *et al.*, 2017] Cesare Alippi, Giacomo Boracchi, and Manuel Roveri. Hierarchical change-detection tests. *IEEE Trans. Neural Netw. Learn. Syst.*, 28(2):246–258, 2017.

[Baena-García *et al.*, 2006] Manuel Baena-García, José del Campo-Ávila, et al. Early drift detection method. In *Int. Workshop Knowledge Discovery from Data Streams*, 2006.

[Bishop, 2006] Christopher M Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[Brzezinski and Stefanowski, 2014] Dariusz Brzezinski and Jerzy Stefanowski. Prequential auc for classifier evaluation and drift detection in evolving data streams. In *Workshop New Frontiers in Mining Complex Patterns*, 2014.

[Ditzler and Polikar, 2011] Greg Ditzler and Robi Polikar. Hellinger distance based drift detection for nonstationary environments. In *IEEE Symp. CIDUE*, pages 41–48, 2011.

[Ditzler *et al.*, 2015] Greg Ditzler, Manuel Roveri, et al. Learning in nonstationary environments: A survey. *IEEE Comput. Intell. Mag.*, 10(4):12–25, 2015.

[Dries and Rückert, 2009] Anton Dries and Ulrich Rückert. Adaptive concept drift detection. *Statistical Anal. Data Mining: The ASA Data Sci. J.*, 2(5-6):311–327, 2009.

[Dyer *et al.*, 2014] Karl B Dyer, Robert Capo, and Robi Polikar. Compose: A semisupervised learning framework for initially labeled nonstationary streaming data. *IEEE Trans. Neural Netw. Learn. Syst.*, 25(1):12–26, 2014.

[Frías-Blanco *et al.*, 2015] Isvani Frías-Blanco, José del Campo-Ávila, et al. Online and non-parametric drift detection methods based on hoeffding's bounds. *IEEE Trans. Knowl. Data Eng.*, 27(3):810–823, 2015.

[Gama *et al.*, 2004] Joao Gama, Pedro Medas, et al. Learning with drift detection. In *Brazilian Symp. on Artificial Intelligence*, pages 286–295. Springer, 2004.

[Gonçalves *et al.*, 2014] Paulo Gonçalves, Silas de Carvalho Santos, et al. A comparative study on concept drift detectors. *Expert Syst. Appl.*, 41(18):8144–8156, 2014.

[Hoeffding, 1963] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American statistical association*, 58(301):13–30, 1963.

[Hoens *et al.*, 2012] T. Hoens, R. Polikar, and N. Chawla. Learning from streaming data with concept drift and imbalance. *Progress in Artificial Intell.*, 1(1):89–101, 2012.

[Kaluža *et al.*, 2010] Boštjan Kaluža, Violeta Mirchevska, et al. An agent-based approach to care in independent living. *Ambient intelligence*, pages 177–186, 2010.

[Katakis *et al.*, 2008] I. Katakis, G. Tsoumakas, and I. Vlahavas. An ensemble of classifiers for coping with recurring contexts in data streams. In *ECAI*, pages 763–764, 2008.

[Kelly *et al.*, 1999] Mark G Kelly, David J Hand, and Niall M Adams. The impact of changing populations on classifier performance. In *KDD*, pages 367–371, 1999.

[Kifer *et al.*, 2004] Daniel Kifer, Shai Ben-David, and Johannes Gehrke. Detecting change in data streams. In *Int. Conf. on Very Large Data Bases*, pages 180–191, 2004.

[Kim and Park, 2017] Youngin Kim and Cheong Hee Park. An efficient concept drift detection method for streaming data under limited labeling. *IEICE Trans. Inf. Syst.*, 100(10):2537–2546, 2017.

[Krawczyk *et al.*, 2017] Bartosz Krawczyk, Leandro L Minku, et al. Ensemble learning for data stream analysis: a survey. *Information Fusion*, 37:132–156, 2017.

[Lindstrom *et al.*, 2011] Patrick Lindstrom, Brian Namee, and Sarah Delany. Drift detection using uncertainty distribution divergence. In *ICDM Workshops*. IEEE, 2011.

[Montgomery, 2009] Douglas Montgomery. *Introduction to statistical quality control*. John Wiley & Sons, 2009.

[Peacock, 1983] JA Peacock. Two-dimensional goodness-of-fit testing in astronomy. *Monthly Notices of the Royal Astronomical Society*, 202(3):615–627, 1983.

[Reis *et al.*, 2016] Denis dos Reis, Peter Flach, et al. Fast unsupervised online drift detection using incremental kolmogorov-smirnov test. In *KDD*, 2016.

[Sethi and Kantardzic, 2017] T. Sethi and M. Kantardzic. On the reliable detection of concept drift from streaming unlabeled data. *Expert Syst. Appl.*, 82:77–99, 2017.

[Sklar, 1959] M Sklar. Fonctions de répartition à n dimensions et leurs marges. *Publ. Inst. Statist. Univ. Paris*, 8:229–231, 1959.

[Sobolewski and Wozniak, 2013] Piotr Sobolewski and Michal Wozniak. Concept drift detection and model selection with simulated recurrence and ensembles of statistical detectors. *J. UCS*, 19(4):462–483, 2013.

[Souza *et al.*, 2015] Vinícius Souza, Diego Silva, et al. Data stream classification guided by clustering on nonstationary environments and extreme verification latency. In *SIAM Int. Conf. Data Mining*, pages 873–881, 2015.

[Tsymbal, 2004] Alexey Tsymbal. The problem of concept drift: definitions and related work. *Computer Science Department, Trinity College Dublin*, 106(2), 2004.

[Wang and Abraham, 2015] Heng Wang and Zubin Abraham. Concept drift detection for streaming data. In *IJCNN*, pages 1–9. IEEE, 2015.

[Wang *et al.*, 2013] Shuo Wang, Leandro L. Minku, et al. Concept drift detection for online class imbalance learning. In *IJCNN*, pages 1–10. IEEE, 2013.

[Widmer and Kubat, 1993] G. Widmer and M. Kubat. Effective learning in dynamic environments by explicit context tracking. In *ECML*, pages 227–243. Springer, 1993.

[Yu and Abraham, 2017] Shujian Yu and Z. Abraham. Concept drift detection with hierarchical hypothesis testing. In *SIAM Int. Conf. Data Mining*, pages 768–776, 2017.

[Žliobaite *et al.*, 2014] Indre Žliobaite, Albert Bifet, et al. Active learning with drifting streaming data. *IEEE Trans. Neural Netw. Learn. Syst.*, 25(1):27–39, 2014.

[Žliobaite, 2010] Indre Žliobaite. Change with delayed labeling: When is it detectable? In *ICDM Workshops*, pages 843–850. IEEE, 2010.