Aprendiendo con detección de cambio online

Isvani Frías Blanco[1], José del Campo Ávila[2], Gonzalo Ramos Jiménez[2], Rafael Morales Bueno[2], Agustín Ortiz Díaz[3] y Yailé Caballero Mota[4]

[1] Universidad de las Ciencias Informáticas, Cuba
[2] Universidad de Málaga, España
[3] Universidad de Granma, Cuba
[4] Universidad de Camagüey, Cuba

{lfriasb, aortizd}@grm.uci.cu, {jcampo, ramos, morales}@lcc.uma.es, yailec@yahoo.com

Resumen. En la actualidad, muchas fuentes generan grandes cantidades de datos en largos períodos de tiempo, requiriéndose su procesamiento incremental. Debido a la dimensión temporal de estos datos, un modelo de aprendizaje inducido previamente puede ser inconsistente con los datos actuales, problema comúnmente conocido como cambio de concepto. Una estrategia ampliamente usada para detectar cambio de concepto supervisa a lo largo del tiempo alguna medida de rendimiento del modelo. Si se estima un deterioro significativo del modelo mediante dicha medida se ejecutan algunas acciones para adaptar el aprendizaje. En este sentido, en el presente artículo se propone un nuevo método para detectar cambio de concepto no dependiente del algoritmo de aprendizaje. Se usa la inecuación de probabilidad de Hoeffding para ofrecer garantías probabilísticas de detección de cambios en la media de flujos de valores reales. Dicho método se basa en la comparación de medias correspondientes a dos muestras, mediante la identificación de un único punto de corte relevante en dicha secuencia de valores reales; manteniendo así un número fijo de contadores además con complejidad temporal constante. Evaluaciones empíricas preliminares considerando conocidos flujos de datos, diferentes detectores de cambio de concepto y algoritmos de aprendizaje muestran promisorio el método propuesto.

Palabras clave. Aprendizaje incremental, cambio de concepto, cota de Hoeffding, detección de cambio de concepto, flujos de datos.

Learning with Online Drift Detection

Abstract. Learning in data streams is a problem of growing interest. The target function of data streams may change over time, so in such situations, a learning model induced with some previous data may be inconsistent with the current data. This problem is commonly known as concept drift. The strategy broadly used to handle concept drift is to continuously monitor a chosen performance measure of the model over time; if the model performance drops, adequate actions are executed to adapt the model. Taking this into account, our paper proposes a new method to detect drifting concepts, which is independent of the learning algorithm. We use a probability inequality (Hoeffding's inequality) to offer probabilistic guarantees for the detection of significant changes in the mean of real values. The detection is based on the comparison of averages corresponding to two samples by means of identification of a single relevant cut-point in this sequence of real values maintaining a fixed number of counters and with constant time complexity. As some previous approaches, our method is based on ideas of statistical process control. Preliminary empirical evaluations considering well-known data streams, change detectors and various classifiers reveal advantages of the proposed method.

Keywords. Incremental learning, concept drift, concept drift detection, control chart, data stream, Hoeffding's bound.

Computación y Sistemas Vol. 18 No. 1, 2014 pp. 169-183
ISSN 1405-5546
http://dx.doi.org/10.13053/CyS-18-1-2014-026

1. Introducción

Cada día existen más situaciones en las que se generan grandes cantidades de datos de forma continua en el tiempo. En el campo de la minería de datos y el aprendizaje automático estas grandes cantidades de datos son conocidas como flujos de datos. Por ejemplo, estos datos son generados frecuentemente en Internet (peticiones de usuarios, datos de correo electrónico, comercio electrónico), detección de fraudes e intrusos, monitorización en biomedicina y procesos industriales, redes de sensores, entre otros ambientes reales [21, 45]. Debido a la gran cantidad de datos y a las restricciones de cómputo, generalmente es necesario que los algoritmos para su procesamiento sean incrementales, los requerimientos comunes para un sistema de aprendizaje incremental incluyen procesar cada instancia con un límite de tiempo y una sola vez, usar una cantidad de memoria limitada y ser capaz de predecir en cualquier momento [9].

La dimensión temporal de los datos junto a la naturaleza dinámica del mundo real puede causar que el concepto subyacente cambie en el tiempo, así, un modelo de aprendizaje inducido previamente puede ser inconsistente con los datos actuales, siendo necesaria su actualización. Este problema es comúnmente conocido como cambio de concepto (*concept drift*). El cambio de concepto es un problema inherente al aprendizaje incremental. En el aprendizaje supervisado, las técnicas más frecuentes para manipular cambio de concepto están relacionadas con selección de instancias, variación del peso de las instancias y aprendizaje con múltiples descriptores de concepto [11, 20]. Una estrategia ampliamente usada para manipular cambio de concepto es monitorizar alguna medida de rendimiento (como la precisión) del modelo actual. Si se estima que el modelo de aprendizaje ya no es consistente con los datos más recientes, algunas acciones son llevadas a cabo para actualizar el modelo.

Usualmente, esta medida de rendimiento tiene su imagen en el conjunto de los números reales, obteniéndose un flujo de valores reales ya que el modelo de aprendizaje es evaluado a lo largo del tiempo. Notables investigaciones [4, 7, 24] han propuesto métodos para detectar cambio de concepto monitorizando estadísticos calculados a partir de estos flujos de valores reales. En los detectores de cambio de concepto generalmente se asumen requerimientos de cómputo semejantes a los algoritmos de aprendizaje incremental.

Estos detectores pueden ser usados independientemente del algoritmo de aprendizaje en ambientes de flujos de datos no estacionarios [4, 7, 24]. Además se puede tener en cuenta características específicas del modelo de aprendizaje, ya que esto puede permitir que dicho modelo se adapte de forma más eficiente y rápida a los cambios. Por ejemplo, los detectores de cambio han sido utilizados para monitorizar partes del modelo de aprendizaje en árboles de decisión [30], aprendizaje basado en instancias [6] y en el clasificador Naïve Bayes [7]. Los detectores de cambio también pueden ser usados como estimadores de la precisión, por ejemplo en la evaluación y comparación de modelos de aprendizaje, ya que las técnicas de evaluación convencionales en el campo de la minería de datos y aprendizaje automático no son adecuadas en ambientes de flujos de datos no estacionarios [25].

Este artículo está enfocado en el problema de detección de cambio de concepto dentro del aprendizaje incremental supervisado. Así, se propone un método para monitorizar la media de valores reales en el tiempo con el objetivo de detectar cambios significativos. Es necesario considerar dos cuestiones importantes en el diseño del método propuesto: la estructura de datos usada para almacenar la información más relevante, y la prueba estadística para estimar cambios significativos en la media de estos valores. La prueba estadística está basada en la inecuación de probabilidad de Hoeffding [29], por lo que solo se asumen muestras relativas a variables aleatorias independientes y acotadas, sin restricción acerca de la función de densidad de probabilidades subyacente.

Métodos cuya complejidad temporal y espacial es constante juegan un papel importante en la detección de cambios online [5]. El método propuesto, como algunos acercamientos anteriores [4, 24], está basado en ideas de control de procesos estadísticos e identifica un

solo punto de corte relevante en dicha secuencia de valores para realizar la prueba estadística.

El artículo está estructurado como sigue. La Sección 2 resume investigaciones relacionadas. En la Sección 3 se describe el método propuesto para la detección de cambios online. A continuación, en la Sección 4 se muestra el rendimiento del algoritmo propuesto en conocidos conjuntos de datos reales y sintéticos, el método es evaluado además teniendo en cuenta diferentes clasificadores. Finalmente, la Sección 5 muestra las conclusiones.

2. Trabajos relacionados

STAGGER [41] e IB3 (*Instance-Based learning algorithm*) [2] fueron posiblemente los primeros algoritmos diseñados para manipular cambio de concepto, desde entonces se han desarrollado numerosos algoritmos para ejecutarse en flujos de datos no estacionario [45]. Entre estos encontramos sistemas basados en reglas [19], árboles de decisión [30, 10], Naïve Bayes [7], Máquinas de Vector Soporte [36, 35], Redes de Funciones de Base Radial [37] y aprendizaje basado en instancias [40, 2, 14].

Existen dos categorías donde se ubican las estrategias para enfrentar el problema en cuestión [24]: estrategias que adaptan el aprendizaje en intervalos de tiempo regulares sin considerar que ha ocurrido un cambio en el concepto; y estrategias que primero detectan el cambio de concepto, y luego el aprendizaje es adaptado al cambio. La segunda categoría requiere métodos más complejos, en esta los detectores de cambio de concepto frecuentemente desempeñan un rol fundamental.

Como ya se ha señalado, monitorizar el rendimiento de un modelo de aprendizaje a lo largo del tiempo da lugar frecuentemente a un flujo de valores reales. De esta forma, diferencias significativas en un estadístico convenientemente calculado a partir de estos valores puede ser interpretado como un cambio en el rendimiento del modelo.

Muchos métodos estadísticos comparan muestras acorde a dos distribuciones de probabilidad, haciendo una prueba estadística bajo la hipótesis nula de que ambas muestras tienen la misma distribución [18]. Se han desarrollado poderosas pruebas paramétricas en la comunidad estadística para este problema, pero en la práctica raramente los datos siguen estas conocidas distribuciones de probabilidad [34]. De esta forma, algunos trabajos investigativos han propuesto pruebas no-paramétricas para detectar cambio en flujos de datos [34, 26, 32]. Otros métodos modelan los datos de entrada de forma incremental, aprendiendo una función de densidad de probabilidades, los acercamientos más explorados asumen un modelo paramétrico predefinido como modelos de densidad de probabilidades [34], modelos auto-regresivos [43], y modelos de estado-espacio [33]. En general, muchos métodos no-paramétricos que asumen menos conocimiento relacionado con los datos de entrada no satisfacen los requerimientos comunes necesarios para el aprendizaje incremental [5, 3, 39]. En otros casos, estos métodos requieren el ajuste de varios parámetros [32], lo cual puede ser una tarea muy difícil.

Debido a la gran cantidad de datos, es importante definir como estos serán almacenados. En el área de la minería de datos y el aprendizaje automático estos datos son almacenados frecuentemente en una estructura de datos que se denomina ventana de tiempo [22]. Conjuntamente han sido propuestos relevantes algoritmos para mantener de forma aproximada estadísticos de interés ofreciendo garantías matemáticas para acotar el error de la estimación [15, 3], ya que no es factible el cálculo exacto de dicho estadístico en flujos de datos. Estos resultados han sido utilizados también para detectar cambio de concepto [7]. Aun así, la complejidad temporal de estos algoritmos basados en ventanas depende de la cantidad de valores vistos hasta el momento. Otros acercamientos, más relevantes para el presente trabajo, son capaces de detectar cambio en una sola pasada por los valores vistos [24, 4], garantizando un límite teórico para el tiempo máximo de procesamiento de cada experiencia.

En el campo de la estadística, algunos esquemas paramétricos bien conocidos que se ajustan a este escenario es el procedimiento CUSUM (*CUmulative SUM*), el procedimiento Shiryayev–Roberts, el control de gráficos de

Shewhart, y el control de gráficos de promedios en movimiento [5]. Otros esquemas no-paramétricos adaptan las clásicas hipótesis basadas en rangos como la prueba Mann-Whitney [28, 38, 39, 44].

Un detector de cambio de concepto muy usado en el área del aprendizaje incremental es DDM (*Drift Detection Method*) [24] y está basado en ideas de control de procesos estadístico. DDM toma el error de predicción de un algoritmo de aprendizaje como una variable aleatoria correspondiente a experimentos de Bernoulli. Siendo la predicción del error p_i y su desviación estándar $s_i = \sqrt{p_i(1-p_i)/i}$. DDM mantiene dos registros en el entrenamiento del algoritmo de aprendizaje, p_{min} y s_{min}, que son actualizados cuando una nueva experiencia causa que $p_i + s_i < p_{min} + s_{min}$. Considerando aproximación a la curva normal, son definidos un nivel de alerta y otro de cambio. Para una confianza de 0.95, DDM dispara una señal de alerta si $p_i + s_i > p_i + 2s_i$ y con confianza 0.99 dispara una señal de cambio si $p_i + s_i > p_{min} + 3s_{min}$. DDM tiene un buen comportamiento frente a cambios abruptos y graduales cuando el cambio no es muy lento. Para mejorar la detección de cambios graduales y lentos se ha propuesto el método EDDM (*Early Drift Detection Method*) [4] donde la idea básica es considerar la distancia entre dos errores de clasificación en lugar de considerar el número de errores.

3. Comparando dos muestras mediante la inecuación de probabilidad de Hoeffding

En el campo de las probabilidades y la estadística, las cotas de concentración constituyen un enfoque importante para la estimación por intervalos. Básicamente, estas indican que la probabilidad de que una variable aleatoria X se desvíe de su valor esperado $E[X]$ es pequeña. En el área del aprendizaje incremental se han utilizado cotas de concentración [13, 29] para la inducción de modelos de aprendizaje como árboles de decisión [12, 16] y reglas de decisión [23], así como para la detección de cambio de concepto [7]. Entre las más populares encontramos las cotas de concentración de Hoeffding [29] y Chernoff [13].

Una importante ventaja de la inecuación de probabilidad de Hoeffding es que no hace suposiciones acerca de la función de distribución de probabilidad. Adicionalmente, la cota puede ser calculada eficientemente, lo que permite su aplicación en condiciones de flujos de datos [7, 12, 16]. Particularmente, el corolario propuesto por Hoeffding [29, página 16] puede ser aplicado a la detección de cambios significativos en la media de flujos de valores reales.

Teorema 1 [29]. Si $X_1, X_2, \dots, X_n$ son variables aleatorias independientes y $a_i \leq X_i \leq b_i$ $(i = 1, 2, \dots, n)$, entonces para $\varepsilon > 0$ y $\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$:

$$P\{\bar{X} - E[\bar{X}] > \varepsilon\} \leq e^{\frac{-2\varepsilon^2 n^2}{\sum_{i=1}^{n}(a_i - b_i)^2}}$$

Corolario 1 [29]. Si $X_1, \dots, X_n, Y_1, \dots, Y_m$ son variables aleatorias independientes con valores en el intervalo $[a, b]$, y si $\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$, $\bar{Y} = \frac{1}{n}\sum_{i=1}^{m} Y_i$, entonces para $\varepsilon > 0$:

$$P\{\bar{X} - \bar{Y} - (E[\bar{X}] - E[\bar{Y}]) > \varepsilon\} \leq e^{\frac{-2\varepsilon^2}{(n^{-1}+m^{-1})(b-a)^2}} \quad (1)$$

Acorde al Corolario 1, sea la hipótesis nula $H_0: E[\bar{X}] \leq E[\bar{Y}]$ contra la alternativa $H_1: E[\bar{X}] > E[\bar{Y}]$ y sea $\bar{X} - \bar{Y} \geq \varepsilon_\alpha$ la regla para rechazar H_0, donde

$$\varepsilon_\alpha = (b-a)\sqrt{\frac{n^{-1}+m^{-1}}{2}\ln\frac{1}{\alpha}} \quad (2)$$

Si en la ecuación (1) tenemos que $E[\bar{X}] \leq E[\bar{Y}]$, con nivel de significancia α, la probabilidad del error de tipo I (probabilidad de detección falsa) para dicha prueba estadística está acotada de la siguiente forma:

$$P\{\bar{X} - \bar{Y} \geq \varepsilon_\alpha\} \leq P\{\bar{X} - \bar{Y} - (E[\bar{X}] - E[\bar{Y}]) \geq \varepsilon_\alpha\} \leq \alpha$$

La probabilidad del error de tipo II (probabilidad de no detección) también puede ser acotada. Si $E[\bar{X}] \geq E[\bar{Y}] + \zeta$, entonces para $\zeta > \varepsilon_\alpha$ se cumple que

$$P\{\bar{X} - \bar{Y} \quad = P\{\bar{X} - \bar{Y} - \zeta < \varepsilon_\alpha - \zeta\} \leq$$
$$< \varepsilon_\alpha\}$$

$$\leq P\{\bar{X} - \bar{Y} - (E[\bar{X}] - E[\bar{Y}]) < \varepsilon_\alpha - \zeta\}$$

$$\leq e^{\frac{-2(\zeta-\varepsilon_\alpha)^2}{(n^{-1}+m^{-1})(b-a)^2}}$$

De una forma obvia se deriva una prueba análoga para la hipótesis nula $\mathcal{H}_0: E[\bar{X}] \geq E[\bar{Y}]$ contra la alternativa $\mathcal{H}_1: E[\bar{X}] < E[\bar{Y}]$, siendo $\bar{Y} - \bar{X} \geq \varepsilon_\alpha$ la regla para rechazar $\mathcal{H}_0$. Así, una prueba de dos colas también puede obtenerse.

ADWIN (*ADaptive WINdow*) [7] monitoriza a lo largo del tiempo la diferencia entre valores medios considerando múltiples pruebas en las cuales el Corolario 1 puede aplicarse. Adicionalmente, el Corolario 2, enunciado más abajo (aunque equivalente al Corolario 1), puede ser ligeramente más conveniente en dependencia de la estructura de datos (ventana) usada para almacenar el flujo de valores reales.

Corolario 2. Si $X_1, \ldots, X_n, X_{n+1}, \ldots, X_{n+m}$ son variables aleatorias independientes con valores en el intervalo $[a, b]$, y si $\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$, $\bar{Y} = \frac{1}{n+m}\sum_{i=1}^{n+m} X_i$, entonces para $\varepsilon > 0$:

$$P\{\bar{X} - \bar{Y} - (E[\bar{X}] - E[\bar{Y}]) > \varepsilon\} \leq e^{\frac{-2\varepsilon^2 n(n+m)}{m(b-a)^2}}$$

Por ejemplo, DDM actualiza la ventana que almacena el flujo de bits de forma tal que aplicar el Corolario 2 es ligeramente más adecuado. De igual forma, la aplicación de dicho Corolario 2 es ligeramente más adecuada en el algoritmo propuesto más adelante en la Sección 4. En este caso, la regla para rechazar la misma hipótesis nula H_0 planteada anteriormente sería $\bar{X} - \bar{Y} \geq \varepsilon'_\alpha$, donde

$$\varepsilon'_\alpha = (b-a)\sqrt{\frac{m}{2n(n+m)}\ln\frac{1}{\alpha}} \qquad (3)$$

La demostración del Corolario 1 no se proporciona en Hoeffding [29]. Así, en el Anexo A se derivan ambos corolarios del Teorema 1.

4. Detector de cambio

Para detectar cambios significativos en la media de flujo de valores reales, se emplea la prueba estadística introducida previamente junto con un acercamiento previo soportado en ideas de control de procesos estadístico [24, 4]. Al resultante detector de cambio lo nombramos HDDM, porque este es similar a DDM pero sin asumir variables aleatorias correspondientes a experimentos de Bernoulli con tendencia a la curva normal, usando más bien la prueba derivada de la inecuación de probabilidad de Hoeffding.

El método propuesto sigue un esquema bien conocido para la manipulación de cambio de concepto (Figura 1) en el aprendizaje supervisado. Aunque el presente artículo estudia el problema del cambio de concepto asumiendo que todas las experiencias están etiquetadas, el detector de cambio propuesto puede ser aplicado en situaciones más reales donde las experiencias llegan de forma online pero las etiquetas de clases son difíciles de obtener. Por ejemplo, estos pueden ser usados representando un flujo de experiencias por medio de un flujo de valores reales a través de márgenes de clasificación [17].

Supongamos una secuencia de experiencias $(\vec{a}_i, c_i)$ donde $\vec{a}_i$ representa sus atributos y c_i su etiqueta de clase correspondiente. A la llegada de cada experiencia, con el objetivo de monitorizar la precisión del modelo de aprendizaje en el tiempo, primero dicho modelo predice $\hat{c}_i$ a través de los atributos $\vec{a}_i$ para luego continuar con el aprendizaje mediante dicha experiencia $(\vec{a}_i, c_i)$. Sea $\ell(c_i, \hat{c}_i)$ una función de pérdida (por ejemplo, la función de pérdida 0-1, donde $\ell(c_i, \hat{c}_i) = 0$ si $c_i = \hat{c}_i$ y $\ell(c_i, \hat{c}_i) = 1$ en otro caso). Si asumimos que dicha función de pérdida es una variable aleatoria independiente y acotada $X_i = \ell(c_i, \hat{c}_i)$ que toma valores reales, se pueden aplicar herramientas del área de las probabilidades y la estadística (ej. la prueba estadística introducida en la Sección anterior) para monitorizar el flujo de valores reales que se genera a partir del esquema anterior. Por ejemplo, un aumento significativo en la media de estos valores reales (en el error de la predicción) puede significar efectivamente que el modelo de aprendizaje no

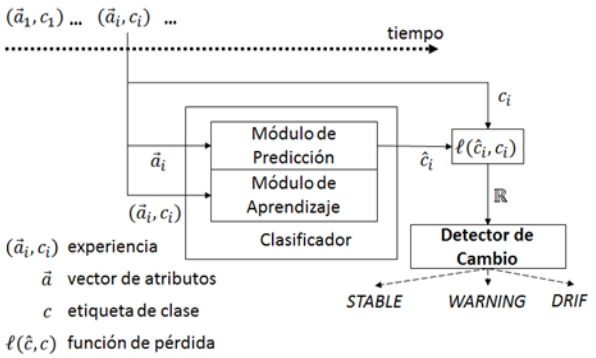

Fig. 1. Esquema ampliamente conocido para la manipulación de cambio de concepto en el aprendizaje supervisado.

es consistente con los nuevos datos, por lo que puede ser necesaria su actualización [24].

De esta forma, a la n-ésima predicción tenemos la secuencia $\mathcal{X} = X_1, \ldots, X_n$ de variables aleatorias independientes y acotadas. Sea $p_n = \frac{1}{n}\sum_{i=1}^{n} X_i$ la estimación del error de un algoritmo de aprendizaje; y sea

$$\varepsilon_n = \sqrt{\frac{1}{2n}\ln\frac{1}{\alpha}}$$

un intervalo de confianza calculado a partir del Teorema 1. Un punto de corte relevante en dicha secuencia, al que podemos llamar min, causa un mínimo en $p_i + \varepsilon_i$ $(1 \leq i \leq n)$ [24]. Entonces se puede aplicar la prueba estadística derivada del Corolario 2 para detectar cambios significativos en la media, teniendo en cuenta dicho punto de corte min en la secuencia de variables aleatorias independientes $\mathcal{X} = X_1, X_2, \ldots, X_{min}, X_{min+1}, \ldots, X_n$.

Sea $\bar{X} = \frac{1}{min}\sum_{i=1}^{min} X_i$ y $\bar{Y} = \frac{1}{n}\sum_{i=1}^{n} X_i$, en la ecuación (3) y acorde a la prueba estadística, $\bar{Y} - \bar{X} \geq \varepsilon'_\alpha$ implica un cambio significativo en la media con nivel de confianza $1 - \alpha$ (ver Algoritmo 1). Un valor considerablemente más grande que α, representado como α_W, puede ser usado como advertencia ante un posible cambio de concepto (nivel de alerta, WARNING). Un valor más conservativo de α, representado como α_D, dispara la definitiva señal de cambio (DRIFT) si $\bar{Y} - \bar{X} \geq \varepsilon'_{\alpha_D}$. Ante una señal de cambio, todas las variables en HDDM son reiniciadas. La situación de concepto estable (STABLE) es recuperada cuando $\bar{Y} - \bar{X} < \varepsilon'_{\alpha_W}$.

Los niveles STABLE, WARNING y DRIFT se han usado efectivamente para manipular cambio de concepto en el aprendizaje incremental en un esquema con dos clasificadores [24, 4]. Así, el clasificador principal es entrenado continuamente y es el encargado de realizar las predicciones. El error de dicho clasificador es monitorizado por medio del detector de cambio. A una señal de alerta (WARNING) disparada por el detector de cambio un segundo clasificador (llamado clasificador alternativo) es creado y entrenado a partir de ese momento en paralelo con el clasificador principal. A una señal DRIFT el clasificador principal es reemplazado por el alternativo. Precisamente siguiendo este esquema, en la presente investigación se ha encontrado empíricamente que $\alpha_W = 0.005$ y $\alpha_D = 0.001$ tiene un buen comportamiento en muchas condiciones de cambio y con varios algoritmos de aprendizaje.

entrada: $X_1, X_2, \ldots, X_n$; $\alpha_{DRIFT}, \alpha_{WARNING}$
salida : *STABLE*, *WARNING*, *DRIFT*

para cada *nuevo valor* X_n **hacer**
  Actualizar $\bar{Y} = \frac{1}{n}\sum_{i=1}^{n} X_i$
  Actualizar el punto de corte *min* en la secuencia $X_1, \ldots, X_{min}, \ldots, X_n$
  si $min == n$ **entonces**
    $\bar{X} = \bar{Y}$
    return *STABLE*
  fin
  si $H_0 : E\left[\bar{Y}\right] \geq E\left[\bar{X}\right]$ *es rechazada con nivel de significancia* $\alpha_{WARNING}$ **entonces**
    return *WARNING*
  fin
  si $H_0 : E\left[\bar{Y}\right] \geq E\left[\bar{X}\right]$ *es rechazada con nivel de significancia* α_{DRIFT} **entonces**
    inicializar todos los contadores
    return *DRIFT*
  fin
  return *STABLE*
fin

Algoritmo 1. Pseudocódigo del detector de cambio de concepto HDDM

En general, la prueba estadística presentada en la Sección anterior puede ser aplicada a otros algoritmos (técnicas de ventana) propuestos anteriormente en la literatura para considerar varios puntos de corte [3, 7, 15]. En este caso, una corrección de *Bonferroni* para α en la ecuación (2) (o análogamente en la ecuación 3) puede corregir múltiples pruebas manteniendo acotada la probabilidad de no detección [7, 8].

5. Evaluación empírica

En esta Sección se muestran resultados preliminares de evaluaciones empíricas hechas al detector de cambio propuesto por medio de flujos de bits, datos sintéticos y reales. El flujo de bits y los conjuntos de datos sintéticos permiten evaluar el comportamiento del algoritmo bajo varias condiciones de cambio; además, la localización del punto de cambio está disponible para el evaluador. El beneficio de experimentar con datos reales es evidente ya que dichos algoritmos están diseñados para ejecutarse en escenarios reales. El propósito fundamental es mostrar la efectividad del método para la detección de cambio de concepto bajo diferentes características del concepto objetivo, varios tipos de cambio y diferentes clasificadores.

En todos los experimentos se incluyeron a DDM y EDDM ya que estos usan un acercamiento similar demostrando un buen rendimiento. También se incluyó ADWIN2, el mismo no tiene complejidad computacional constante ni en tiempo ni en espacio pero tiene garantías matemáticas de desempeño y ha sido usado ampliamente en el aprendizaje en flujos de datos no estacionarios. Para HDDM y ADWIN2 en todos los experimentos las variables aleatorias se definieron acotadas en el intervalo $[0,1]$.

Aunque en la literatura han sido propuestas varias medidas de rendimiento para los detectores de cambios online, sobre flujos de bits se evalúan las tres más usadas en el aprendizaje incremental [7, 25]: retardo en la detección del cambio, tasa de falsos negativos (cambios de concepto no detectados) y tasa de falsos positivos (detecciones de falsos cambios).

También se evaluó la efectividad del detector de cambio en flujos de datos artificiales en combinación con clasificadores diferentes. El cambio de concepto es simulado cambiando la función objetivo en intervalos de tiempo regulares (cada $50{,}000$ experiencias). Así, se mide el comportamiento de los algoritmos sobre cambio abrupto y gradual; presencia y ausencia de ruido; presencia de atributos relevantes, simbólicos y numéricos. Adicionalmente, se toma ventaja de los generadores de flujos de datos ya que ellos pueden producir posiblemente un infinito número de instancias [9]: cada 100 experiencias de entrenamiento, cada algoritmo resultante se prueba con otras 100 experiencias (sólo usadas para la prueba). Se supone que altos valores de precisión implican un mejor comportamiento del respectivo detector de cambio (siempre considerando el mismo clasificador).

Adicionalmente, se han considerado dos conocidos conjuntos de datos reales. Para evaluar la precisión en estos datos que provienen de problemas reales, cada experiencia es, primero, clasificada por el modelo de aprendizaje (para evaluar su precisión) y, posteriormente, utilizada por el algoritmo para su aprendizaje [25]. Junto con esta técnica, la precisión se calcula con los valores almacenados en una ventana deslizante de tamaño $1{,}000$ [7, 25].

5.1. Flujos de bits

Primero se evalúa el comportamiento de los detectores de cambio en un flujo de bits, acorde a una distribución de Bernoulli con parámetro μ. En todos los experimentos se hicieron 50 cambios en la media μ. Se consideraron $1{,}000$ y $100{,}000$ bits por valor de la media. Cada método se ejecutó 50 veces para cada una de estas configuraciones.

Para distinguir (por analogía) la extensión del cambio [42] se restringió en un intervalo la diferencia aleatoria entre consecutivos valores de la media $|\mu_{anterior} - \mu_{nueva}| \in [a, b]$. Para cada detector se estimó la media de retardo para la detección del cambio (Ret., expresada como el número de bits), la tasa de falsos negativos (FN) y la tasa de falsos positivos (FP).

Aunque en experimentos adicionales hemos considerado varios valores para el nivel de significancia (α) en HDDM. En el presente artículo sólo se muestran los resultados para $\alpha = 0.001$ (e.g. $\alpha_D = 0.001$) por cuestiones de

espacio. Este nivel de significancia mantuvo la tasa de falsos negativos y falsos positivos por debajo de los restantes detectores de cambio en muchos experimentos, detectando además los cambios más rápidamente en muchas ocasiones. DDM estima cambio de medias si $p_i + s_i > p_{min} + 3s_{min}$ o si $p_i - s_i < p_{max} - 3s_{max}$ ($p_{max} - s_{max}$ causa un máximo análogo al mínimo de $p_{min} + s_{min}$).

Las Tablas 1 y 2 muestran los resultados del experimento para $1\,000$ y $100\,000$ bits respectivamente. HDDM mantuvo la tasa de falsos positivos y falsos negativos en correspondencia con el nivel de significancia y el nivel de confianza calculados en la Sección 3. Como se puede apreciar en dichas tablas, HDDM detectó cambios en la media más rápido que DDM, siendo esta diferencia más notable cuando el cambio entre medias (longitud de conceptos estables) es más prolongado.

Si bien las Tablas 1 y 2 reflejan que HDDM tuvo un comportamiento superior que DDM bajo la configuración de los experimentos, EDDM por lo general detectó los cambios más rápidamente que HDDM. Sin embargo, la tasa de falsos positivos es mucho más alta en EDDM, lo que implica que este detector es más susceptible al ruido.

En las Tablas 1 y 2 también se puede observar que la longitud de los conceptos (medias) estables influye notablemente en el retardo en la detección de cambio en HDDM y DDM, no ocurriendo lo mismo en EDDM y ADWIN2. En particular, se observó que en dichas configuraciones de los experimentos la estimación del punto de corte en HDDM y DDM estuvo considerablemente más alejada del punto de corte real. Este hecho puede estar dado porque al estimar dicho punto de corte considerando el promedio, cuando la media poblacional se mantiene constante por largo tiempo y luego cambia, los valores más viejos (e.g. que se corresponden con la media poblacional anterior) tienen mucha influencia en la estimación de la media actual, causando que el promedio esté sesgado con respecto a la media anterior y ocurriendo así un retardo considerable en la estimación del punto de corte. Este efecto no es el mismo en EDDM dada su tasa significativamente más alta de falsos positivos (cuando un cambio de concepto es estimado los contadores del método son restaurados), ni en ADWIN2 al considerar múltiples puntos de cortes que no dependen del estimador de la media poblacional.

Sin embargo, HDDM tuvo mejor comportamiento que ADWIN2 en la configuración del experimento correspondiente a la Tabla 1. Cuando los conceptos estables no son tan prolongados, HDDM es capaz de estimar puntos de cambio con más precisión. Al mismo tiempo, al considerar más puntos de corte, ADWIN2 usa una cota menos ajustada para mantener acotada la tasa de falsos positivos, lo que permite a

Tabla 1. Resultados del experimento en 1 000 bits por valor de la media

		[0.1,0.3]	[0.3,0.5]	[0.5,0.7]	[0.7,0.9]
HDDM	FP	3.68E-05	3.72E-05	2.96E-05	1.04E-05
	FN	0.0192	0	0	0
	Ret.	158.03	40.77	21.65	13.38
DDM	FP	1.04E-05	4.64 E-04	6.47E-04	0.001
	FN	0.1272	0.0028	0.0008	0.0028
	Ret.	198.88	115.52	57.39	28.86
EDDM	FP	0.023	0.023	0.025	0.028
	FN	0	0	0	0
	Ret.	23.97	26.76	30.03	31.58
ADWIN2	FP	0.004	0.007	0.008	0.009
	FN	0.05	0	0	0
	Ret.	209.59	56.38	28.26	16.10

Tabla 2. Resultados del experimento en 100 000 bits por valor de la media

		[0.1,0.3]	[0.3,0.5]	[0.5,0.7]	[0.7,0.9]
	FP	9.46E-06	9.20E-06	5.64E-06	1.66E-06
HDDM	FN	0.0004	0	0	0
	Ret.	575.24	226.84	148.94	96.29
	FP	7.75E-06	1.02E-05	1.22E-05	2.27E-05
DDM	FN	0.098	0.0008	0	0.0004
	Ret.	13599.10	9241.13	4458.55	1975.57
	FP	0.024	0.024	0.026	0.030
EDDM	FN	0	0	0	0
	Ret.	23.82	26.49	29.98	31.07
	FP	8.40E-05	8.22E-05	8.83E-05	9.32E-05
ADWIN2	FN	0	0	0	0
	Ret.	198.35	53.27	26.66	15.57

HDDM detectar más rápidamente los cambios con una tasa de falsos positivos y falsos negativos aceptable.

5.2. Datos sintéticos

HDDM, DDM, EDDM y ADWIN2 monitorizan la tasa del error de tres algoritmos de aprendizaje [9]: el clasificador Naïve Bayes, Perceptrón y VFDT (*Very Fast Decision Tree*) [16]. Cada intento de predicción (controlado por instancias de DDM o HDDM) da lugar a un 0 (si el respectivo clasificador predijo correctamente) o 1. Así, ambos detectores tienen como entrada un flujo de bits, y como salida las señales STABLE, WARNING o DRIFT. A la señal WARNING un clasificador alternativo es inducido en paralelo al original que lo sustituirá si dicha señal es seguida de otra señal DRIFT.

Lógicamente, la combinación de los detectores con los clasificadores da lugar a 12 algoritmos de aprendizaje. También se incluyen en cada una de las tablas los resultados correspondientes a los algoritmos ejecutados sin detector de cambio (fila SD, Sin Detector). Naturalmente, la comparación de valores de precisión entre diferentes modelos de aprendizaje no es relevante.

Se evaluaron los algoritmos resultantes en tres conocidos generadores de datos artificiales: pantalla LED (*Light Emitting Diode*) de 7-segmentos [9], AGRAWAL [1] y STAGGER [41]. Los conceptos cambiaron cada $50{,}000$ instancias.

En el primer flujo de datos, la tarea es predecir el dígito mostrado en una pantalla de 7 segmentos, donde cada atributo tiene un 10% de probabilidad de ser invertido. La configuración particular del generador usado en el presente experimento produce 24 atributos binarios, 17 de los cuales son irrelevantes. El cambio es simulado intercambiando atributos relevantes.

Agrawal y otros autores [1] generan una función a partir de otras diez funciones diferentes predefinidas. El generador produce un flujo donde cada experiencia contiene 9 atributos, de los cuales 6 son numéricos y tres categóricos. Las 10 funciones definidas generan etiquetas de clase binarias. El cambio es simulado cambiando la función que clasifica las experiencias.

STAGGER genera conceptos introducidos por Schlimmer y Granger [41]. Los conceptos son funciones booleanas de tres atributos que representan objetos: tamaño (pequeño, medio y largo), forma (círculo, triángulo y rectángulo) y color (rojo, azul y verde). Se simula cambio eligiendo entre tres funciones de clasificación distintas:

$(tamaño = pequeño) \wedge (color = rojo)$,
$(color = verde) \vee (forma = círculo)$ o
$(tamaño = medio) \vee (tamaño = largo)$.

Cada 100 experiencias, cada clasificador fue probado con otras 100 experiencias. Las Tablas 3, 4 y 5 muestran el valor medio y desviación estándar para el porcentaje de instancias bien clasificadas. En cada tabla se ejecutaron 10 cambios de concepto para cada una de las configuraciones.

Como DDM y EDDM, también fue definido un nivel de alerta en HDDM (y análogamente en ADWIN2) para $\alpha_W = 0.005$ y el nivel de cambio $\alpha_D = 0.001$.

Las Tablas 3, 4 y 5 muestran que HDDM, en combinación con los clasificadores Perceptrón y Naïve Bayes, en la mayoría de los casos alcanzó mayores valores de precisión que DDM y EDDM. Sin embargo, VFDT (en comparación con el Perceptrón y Naïve Bayes) de forma general tarda más en aprender el concepto objetivo, dando lugar a que DDM en este caso, que es más robusto al ruido, alcance niveles más altos de precisión ya que con VFDT, en muchos casos existe un coste adicional considerable en inducir un nuevo árbol de decisión.

En las Tablas 3, 4 y 5 también se puede observar que a pesar de que ADWIN2 de forma general alcanzó los valores más altos de precisión, HDDM en muchas ocasiones tuvo un comportamiento similar. Es válido recalcar que a diferencia de HDDM, DDM y EDDM; ADWIN2 no tiene complejidad temporal ni espacial constante, sino que dicha complejidad computacional depende del número de valores vistos hasta el momento.

Como se esperaba, todos los detectores de cambio de concepto en combinación con el algoritmo de aprendizaje alcanzaron niveles de precisión más altos que el algoritmo de aprendizaje sin la manipulación de cambio de concepto (excepto en LED con el clasificador VFDT, que no fue capaz de aprender los conceptos objetivos bajo las condiciones del experimento). Esto muestra la efectividad del método propuesto en el aprendizaje en flujos de datos no estacionarios.

Tabla 3. Precisión ± desviación estándar en LED

	Perceptrón	Naïve Bayes	VFDT
HDDM	77.35±2.64	**74.77±1.76**	15.33±8.40
DDM	77.30±3.31	68.22±14.37	26.72±14.58
EDDM	70.61±14.33	71.26±11.56	33.87±13.70
ADWIN2	**77.48±2.41**	**74.76±1.52**	15.23±8.33
SD	64.37±17.01	48.46±20.28	34.51±14.02

Tabla 4. Precisión ± desviación estándar en AGRAWAL

	Perceptrón	Naïve Bayes	VFDT
HDDM	**64.38±22.41**	84.18±12.30	88.80±9.04
DDM	64.17±22.48	83.93±12.63	**89.57±8.38**
EDDM	62.33±23.56	81.74±14.99	86.62±12.56
ADWIN2	**64.39±22.41**	**84.29±12.29**	88.73±9.55
SD	60.46±24.45	63.72±15.80	63.30±15.34

Tabla 5. Precisión ± desviación estándar en STAGGER

	Perceptrón	Naïve Bayes	VFDT
HDDM	**91.00±11.60**	100.00±0.03	98.07±7.88
DDM	89.94±11.22	100.00±0.03	**98.51±6.67**
EDDM	89.50±13.33	100.00±0.03	96.90±8.76
ADWIN2	**91.04±11.52**	100.00±0.03	**98.52±6.65**
SD	61.60±29.24	63.72±15.80	65.82±20.23

5.3. Datos reales

Las Tablas 6 y 7 muestran valores de precisión también en términos de valor medio y desviación estándar correspondientes a dos conjuntos de datos reales [27, 31].

Estos conjuntos de datos fueron seleccionados porque han sido ampliamente usados para mostrar la efectividad de algoritmos de aprendizaje en datos con cambio de concepto.

Tabla 6. Precisión ± desviación estándar en *elec2*

	Perceptrón	Naïve Bayes	VFDT
HDDM	43.78±1.85	**85.80±0.78**	66.56±1.14
DDM	43.66±1.81	84.75±0.94	71.53±2.32
EDDM	**44.61±2.38**	85.73±0.75	62.75±1.01
ADWIN2	43.12±1.60	82.10±1.15	69.44±4.44
SD	42.49±1.55	76.75±2.63	**74.15±5.17**

Tabla 7. Precisión ± desviación estándar en *spam_corpus2*

	Perceptrón	Naïve Bayes	VFDT
HDDM	96.50±0.74	88.90±2.47	**81.31±5.74**
DDM	**96.84±0.74**	84.13±5.70	79.94±6.33
EDDM	96.52±0.84	87.91±2.80	79.64±6.57
ADWIN2	96.58±0.72	**89.17±2.10**	78.48±6.86
SD	96.65±0.80	87.02±3.80	72.93±8.38

El conjunto de datos de predicción de electricidad (*elec2*) consiste en 52,312 experiencias coleccionadas cada 30 minutos entre el 7 de mayo de 1996 y el 5 de diciembre de 1998 [27]. La tarea es predecir si el precio de la electricidad subirá o bajará basado en cinco atributos numéricos: el día de la semana, el período del día basado en los 30 minutos, la demanda de electricidad en el distrito al sureste de Australia (*New South Wales*), la demanda en Victoria y la cantidad de electricidad a ser transferida entre estas dos. Alrededor del 39% de las experiencias tiene valores perdidos para ambos la demanda en Victoria y la cantidad de electricidad transferida.

El segundo conjunto de datos (*spam_corpus2*) es una colección de correos "no deseados" (*Spam Assassin collection*) y contienen tanto correos spam como legítimos. Al contrario de los datos sintéticos, no se conoce con seguridad la presencia o tipos de cambio [31].

Las Tablas 6 y 7 muestran que HDDM en los conjuntos de datos reales considerados también tuvo un buen comportamiento. Por ejemplo, la combinación HDDM-Naïve Bayes fue el clasificador más preciso en *elec2* y la combinación HDDM-VFDT fue la más precisa en *spam_corpus2* (en comparación con otros detectores de cambio sin variar el clasificador).

Como se puede observar en la Tabla 6, el clasificador VFDT nuevamente fue más preciso sin detector de cambio de concepto, lo que sugiere que en la inducción de árboles de decisión no es tan efectiva la estrategia utilizada en el presente artículo para el aprendizaje en flujos de datos con cambio de concepto. Así, nuevos enfoques en la inducción de árboles de decisión en flujos de datos no estacionarios pueden ser necesarios.

6. Conclusiones

En este artículo se presenta un método para la detección de cambio de concepto en el dominio del aprendizaje incremental. Dicho método no depende del algoritmo de aprendizaje, y consecuentemente puede aplicarse a cualquier clasificador para la manipulación de cambio de concepto. Adicionalmente, el mismo tiene complejidad temporal y espacial constante.

Otra característica relevante del método propuesto es la ausencia de parámetros relacionados con conocimientos a priori de la estructura del cambio, este es capaz de adaptar el tamaño de la ventana que contiene valores reales en respuesta a la tasa de cambio estimada. Así, asumiendo solo valores reales regidos por variables aleatorias independientes y acotadas, se ofrecen garantías probabilísticas para la detección de cambios significativos en la media de dicho flujo de valores en términos de tasa de falsos positivos y falsos negativos.

Experimentos preliminares con conocidos conjuntos de datos reales y sintéticos muestran que el detector de cambio mejora la capacidad de aprendizaje de algoritmos ejecutados en flujos de datos con cambio de concepto; manteniendo un buen comportamiento en presencia de ruido y en períodos donde el concepto objetivo es estable. Esto muestra que el mencionado detector adapta efectivamente su comportamiento a las características del problema a tratar.

Se espera continuar con la presente investigación considerando más conjuntos de datos reales y sintéticos, otras funciones de pérdida y explotando características específicas de algoritmos para adaptar más eficientemente el aprendizaje al cambio de concepto. Específicamente, se ha empezado a integrar el método propuesto con algoritmos de inducción de árboles de decisión.

Anexo A

Sean $n+m$ variables aleatorias independientes $X_1, \dots, X_n, X_{n+1}, \dots, X_{n+m}$ acotadas en el intervalo $[a, b]$. Sean $Y_i = mX_i$ para $1 \leq i \leq n$ y $Y_i = -nX_i$ para $n+1 \leq i \leq n+m$ también variables aleatorias independientes. Entonces $Y_i \in [ma, mb]$ para $1 \leq i \leq n$ y $Y_i \in [-na, -nb]$ para $n+1 \leq i \leq n+m$.

Considerando entonces las variables aleatorias $Y_1, \dots, Y_n, Y_{n+1}, \dots, Y_m$, y haciendo $\varepsilon = t/(n+m)$ en el Teorema 1 tenemos:

$$P\left\{\sum_{i=1}^{n+m} Y_i - E\left[\sum_{i=1}^{n+m} Y_i\right] > t\right\} \leq e^{\frac{-2t^2}{\sum_{i=1}^{n}(a_i - b_i)^2}} \qquad (4)$$

pero

$$\sum_{i=1}^{n} (a_i - b_i)^2 = nm(n+m)(b-a)^2 \qquad (5)$$

Finalmente, reemplazando $t = \xi nm$ y la ecuación (5) en la ecuación (4), se obtiene el Corolario 1 luego de algunas transformaciones algebraicas. El Corolario 2 se obtiene de una forma análoga, reemplazando $t = \xi n(n+m)$ y la ecuación (5) en la ecuación (4).

Referencias

1. **Agrawal, R., Imielinski, T., & Swami, A. (1993).** Database mining: A performance perspective. *IEEE Transaction on Knowledge and Data Engineering,* 5(6), 914–925.
2. **Aha, D.W., Kibler, D., & Albert, M.K. (1991).** Instance-based learning algorithms. *Machine Learning,* 6(1), 37–66.
3. **Babcock, B., Babu, S., Datar, M., Motwani, R., & Widom, J. (2002).** Models and issues in data stream systems. *Twenty-first ACM SIGMOD-SIGACT-SIGART Symposium on Principles of database systems (PODS'02)*, Madison, Wisconsin, USA, 1–16.
4. **Baena-García, M., del Campo-Ávila, J., Fidalgo, R., Bifet, A., Gavaldà, R., & Morales-Bueno, R. (2006).** Early Drift Detection Method. *Fourth International Workshop on Knowledge Discovery from Data Streams.*
5. **Basseville, M. & Nikiforov, I.V. (1993).** *Detection of Abrupt Changes: Theory and Application.* Englewood Cliffs, NJ: Prentice-Hall.
6. **Beringer, J. & Hüllermeier, E. (2007).** Efficient instance-based learning on data streams. *Intelligent Data Analysis,* 11(6), 627–650.
7. **Bifet, A. & Gavaldà, R. (2007).** Learning from time-changing data with adaptive windowing. *2007 SIAM International Conference on Data Mining*, Minneapolis, Minnesota, 443–448.
8. **Bifet, A. & Gavaldá, R. (2009).** Adaptive learning from evolving data streams. *8th International Symposium on Intelligent Data Analysis: Advances in Intelligent Data Analysis VIII* (IDA '09), Lyon, France, 249–260.
9. **Bifet, A., Holmes, G., Kirkby, R., & Pfahringer, B. (2010).** MOA: Massive Online Analysis. *Journal of Machine Learning Research,* 11(2010), 1601–1604.
10. **Bifet, A., Holmes, G., Pfahringer, B., & Frank, E. (2010).** Fast perceptron decision tree learning from evolving data streams. *14th Pacific-Asia conference on Advances in Knowledge Discovery and Data Mining* - Volume Part II (PAKDD'10), Hyderabad, India, 299–310.
11. **Bifet, A., Holmes, G., Pfahringer, B., Kirkby, R., & Gavaldà, R. (2009).** New ensemble methods for evolving data streams. *15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (KDD '09), Paris, France, 139–148.
12. **del Campo-Ávila, J., Ramos-Jiménez, G., Gama, J., & Morales-Bueno, R. (2008).** Improving the performance of an incremental algorithm driven by error margins. *Intelligent Data Analysis-Knowledge Discovery from Data Streams,* 12(3), 305–318.
13. **Chernoff, H. (1952).** A Measure of Asymptotic Efficiency for Tests of a Hypothesis Based on the Sum of Observations. *Annals of Mathematical Statistics,* 23(4), 493–507.
14. **Cunningham, P., Nowlan, N., Delany, S.J., & Haahr, M. (2003).** A case-based approach to spam

filtering that can track concept drift. *ICCBR'2003 Workshop on Long-Lived CBR Systems, Trondheim, Norway*.

15. **Datar, M., Gionis, A., Indyk, P., & Motwani, R. (2002).** Maintaining stream statistics over sliding windows. SIAM *Journal on Computing,* 31(6), 1794–1813.
16. **Domingos, P. & Hulten, G. (2000).** Mining High-Speed Data Streams. *Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (KDD '00), Boston, MA, USA, 71–80.
17. **Dredze, M., Oates, T., & Piatko, C. (2010).** We're not in Kansas anymore: detecting domain changes in streams. *2010 Conference on Empirical Methods in Natural Language Processing* (*EMNLP'10)*, Massachusetts, USA, 585–595.
18. **Dries, A. & Rückert, U. (2009).** Adaptive concept drift detection. *Statistical Analysis and Data Mining,* 2(5–6), 311–327.
19. **Ferrer-Troyano, F.J., Aguilar, J.S., & Riquelme, J.C. (2005).** Incremental Rule Learning and Border Examples Selection from Numerical Data Streams. *Journal of Universal Computer Science,* 11(8), 1426–1439.
20. **Frías, I., Ortiz, A., Ramos, G., Morales, R., & Caballero, Y. (2010).** Clasificadores y multiclasificadores con cambio de concepto basados en árboles de decisión. *Revista Iberoamericana de Inteligencia Artificial* 14(45), 32–43.
21. **Gama, J. (2010).** *Knowledge Discovery from Data Streams*. Boca Raton, FL: Chapman and Hall/CRC.
22. **Gama, J. & Gaber, M.M. (2007).** *Learning from Data Streams: Processing Techniques in Sensor Networks*. Berlin; New York: Springer.
23. **Gama, J. & Kosina, P. (2011).** Learning decision rules from data streams. *Twenty-Second International Joint Conference on Artificial Intelligence*, Barcelona, Catalonia, Spain, 1255–1260.
24. **Gama, J., Medas, P., Castillo, G., & Rodrigues, P. (2004).** Learning with drift detection. *Advances in Artificial Intelligence, SBIA 2004*, *Lecture Notes in Computer Science,* 3171, 286–295.
25. **Gama, J., Sebastião, R., & Rodrigues, P. (2009).** Issues in Evaluation of Stream Learning Algorithms. *15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Paris, France, 329–338.
26. **Gretton, A., Borgwardt, K., Rasch, M., Schölkopf, B., & Smola, A. (2006).** A Kernel Method for the Two Sample Problem. Twentieth Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada*,* 513–520.
27. **Harries, M.B., Sammut, C., & Horn, K. (1998).** Extracting hidden context. *Machine Learning - Special issue on context sensitivity and concept drift,* 32(2), 101–126.
28. **Hawkins, D.M. & Deng, Q. (2010).** A Nonparametric Change-Point Control Chart. *Journal of Quality Technology,* 42(2), 165–173.
29. **Hoeffding, W. (1963).** Probability inequalities for sums of bounded random variables. *Journal of American Statistical Association,* 58(301), 13–30.
30. **Ikonomovska, E., Gama, J., & Dzeroski, S. (2011).** Learning model trees from evolving data streams. Data Mining and Knowledge Discovery, 23(1), 128–168.
31. **Katakis, I., Tsoumakas, G., & Vlahavas, I. (2008).** An Ensemble of Classifiers for coping with Recurring Contexts in Data Streams. *2008 conference on ECAI 2008: 18th European Conference on Artificial Intelligence*, Patras, Greece, 763–764.
32. **Kawahara, Y. & Sugiyama, M. (2009).** Change-Point Detection in Time-Series Data by Direct Density-Ratio Estimation. *SIAM International Conference on Data Mining (SDM 2009)*, Sparks, Nevada, USA, 389–400.
33. **Kawahara, Y., Yairi, T., & Machida, K. (2007).** Change-Point Detection in Time-Series Data Based on Subspace Identification. *Seventh IEEE International Conference on Data Mining* (ICDM '07), Omaha, NE, USA, 559–564.
34. **Kifer, D., Ben-David, S., & Gehrke, J. (2004).** Detecting Change in Data Streams. *Thirtieth International Conference on Very Large Data Bases* (*VLDB '04)*, Toronto, Ontario, Canada, 30, 180–191.
35. **Klinkenberg, R. (2004).** Learning drifting concepts: example selection vs. example weighting. *Intelligent Data Analysis,* 8(3), 281–300.
36. **Klinkenberg, R. & Joachims, T. (2000).** Detecting Concept Drift with Support Vector Machines. *Seventeenth International Conference on Machine Learning* (*ICML '00)*, Stanford, CA, USA, 487–494.
37. **Kubat, M. & Widmer, G. (1994).** Adapting to drift in continuous domains. *Machine Learning: ECML-95, Lecture Notes in Computer Science*, 912, 307–310.
38. **Pettitt, A.N. (1979).** A Non-Parametric Approach to the Change-Point Problem. *Journal of the Royal*

Statistical Society, Series C (Applied Statistics), 28(2), 126–135.

39. **Ross, G., Tasoulis, D.K., & Adams, N.M. (2011).** Nonparametric Monitoring of Data Streams for Changes in Location and Scale. *Technometrics,* 53(4), 379–389.
40. **Salganicoff, M. (1997).** Tolerating Concept and Sampling Shift in Lazy Learning Using Prediction Error Context Switching. *Artificial Intelligence Review,* 11(1-5), 133–155.
41. **Schlimmer, J.C. & Granger Jr., R.H. (1986).** Incremental learning from noisy data. *Machine Learning* 1(3), 317–354.
42. **Scholz, M. & Klinkenberg, R. (2007).** Boosting classifiers for drifting concepts. *Intelligent Data Analysis - Knowlegde Discovery from Data Streams,* 11(1), 3–28.
43. **Yamanishi, K. & Takeuchi, J.I. (2002).** A Unifying Framework for Detecting Outliers and Change Points from Non-Stationary Time Series Data. *Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (KDD '02), Edmonton, AB, Canada, 676–681.
44. **Zhou, C., Zou, C., Zhang, Y., & Wang, Z. (2009).** Nonparametric control chart based on change-point model. *Statistical Papers,* 50(1), 13–28.
45. **Žliobaitė, I. (2009).** Learning under Concept Drift: an Overview. Lithuania: Vilnius University.

Isvani Frías-Blanco received the MSc degree in computer science from the Universidad de Oriente, Cuba, in 2005. He is an assistant professor in the Universidad de las Ciencias Informáticas, La Habana, Cuba. His research interests include machine learning and data mining.

José del Campo-Ávila received the PhD degree in software engineering and artificial intelligence from the University of Málaga in 2007. His research interests include incremental learning, mining, data streams for classification, and multiple classifier systems, among others. He is an assistant professor in the Department of Lenguajes y Ciencias de la Computación at the University of Málaga and a member of the (IA)2 research group.

Gonzalo Ramos-Jiménez received the degree in computer science engineering and the degree in psychology from the University of Málaga, Málaga, Spain. In 2001 received the PhD degree in computer science from the same university. He is a full professor in the Department of Lenguajes y Ciencias de la Computación at the University of Málaga. His research interests include machine learning and data mining, among others. He is also interested in cognitive science, and he is a member of EATCS (European Association for Theoretical Computer Science), member of AEPIA (Asociación Española de Inteligencia Artificial) and member of the (IA)2 research group.

Rafael Morales-Bueno received the PhD degree in computer science from the University of Málaga, Málaga, Spain, in 1991. He is a full professor in the Department of Lenguajes y Ciencias de la Computación at the University of Málaga. He is a relevant member in the (IA)2 research group. His research interests include computational learning, machine learning and data mining, among others. He is a member of EATCS (European Association for Theoretical Computer Science).

Agustín Ortiz-Díaz received the MSc degree in computer science from the Universidad de Oriente, Cuba, in 2003. He is an assistant professor in the Universidad de Granma, Granma, Cuba. His research interests include machine learning and data mining.

Yailé Caballero-Mota received the MSc degree in computer science from the Universidad Central de las Villas, Santa Clara, Cuba, and the PhD degree in computer science, in 2007, from the same university. She is a full professor in the Department of Computer Science, University of Camagüey, Camagüey, Cuba. Her research interests include artificial neural networks, machine learning and data mining.

Artículo recibido el 22/04/2013, aceptado el 30/06/2013.