Revista Cubana de Ciencias Informáticas
Vol. 10, No. 2, abril-mayo, 2016
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 1-13

Tipo de artículo: Artículo original
Temática: Reconocimiento de patrones
Recibido: 10/11/2015 | Aceptado: 07/03/2016

Limpieza de ruido para clasificación basado en vecindad y cambios de concepto en el tiempo

Noise cleaning for classification based on neighborhood and concept changes over time

Jorge Luis Toro Pozo[1*], **Damaris Pascual González**[2], **Fernando Daniel Vázquez Mesa**[3]

[1]Universidad de Las Tunas. Av. Carlos J. Finlay SN, Rpto. Santos, Las Tunas
[2,3]Universidad de Oriente. Av. Patricio Lumumba SN, Santiago de Cuba

[*]Autor para correspondencia: jorgitoltp@gmail.com

Resumen
En la minería de datos y reconocimiento de patrones, un importante campo lo constituye la clasificación. La clasificación es necesaria en muchos procesos del mundo de hoy. Muchos son los estudios y métodos propuestos con el fin de hacer que los clasificadores sean cada vez más efectivos. Sin embargo, la mayoría de ellos consideran la perfección en los conjuntos de entrenamiento, sin tener en cuenta que podría haber, dentro de estos conjuntos de entrenamiento, objetos con etiquetas de clases erróneas, producto tanto de errores humanos como de previos procesos de clasificación. Al proceso de eliminar estos objetos mal clasificados, se denomina limpieza de ruido. Obviamente, la limpieza de ruido influye considerablemente en la correcta clasificación de nuevas muestras. En esta investigación, se presenta un nuevo algoritmo de limpieza de ruido en flujos de datos para clasificación, basado en criterios de vecindad. Además, considera cambios en la distribución de los datos que pueden ocurrir en el transcurso del tiempo. Se evaluó, mediante varios experimentos, el efecto de la aplicación del método en la construcción automática de conjuntos de entrenamiento usando bases de datos del repositorio UCI y dos sintéticas. Los resultados obtenidos demuestran la eficacia de la estrategia de limpieza de ruido y su influencia en la correcta clasificación de nuevas muestras.

Palabras claves: Limpieza de ruido, aprendizaje semi-supervisado, cambios de concepto

Abstract
An important field within data mining and pattern recognition is classification. Classification is necessary in a number nowadays-world processes. Several works and methods have been proposed with the goal to achieve classifiers to be more effective each time. However, most of them consider the training sets to be perfectly clustered, without having into account that incorrectly classified data might be in them. The process of removing incorrectly classified objects is called noise cleaning. Obviously, noise cleaning influences considerably in classification of new samples. In this work, we present a neighborhood-based algorithm for noise cleaning on data stream for classification. In addition, it considers the data distribution changes that may occur on the time. It was measured, by several experiments, the effect of the method on automatic building of training sets by using databases from UCI repository and two synthetic ones. The obtained results show prove the efficacy of the proposed noise cleaning strategy and its influence on the right classification of new samples.

Keywords: *Noise cleaning, semi-supervised learning, concept drift*

Introducción

En el mundo actual, varias son las esferas en las que es necesario realizar un proceso de clasificación. Para realizar el proceso de clasificación se necesita un conjunto de muestras etiquetadas (prototipos) lo suficientemente representativas, que sean capaces de emitir un juicio correcto acerca de la clase a la cual pertenece un nuevo objeto. Este conjunto de muestras etiquetadas se conoce en la literatura como conjunto de entrenamiento (Training Set, TS).

Los algoritmos de clasificación semi-supervisada o de aprendizaje semi-supervisado (Chapelle et al., 2006; Kalish et al., 2011; Liu et al., 2009; Rohban and Rabiee, 2012; Settles, 2010; Zhou and Goldman, 2004) tienen como única información *a priori* pocas muestras de las clases presentes y cuentan con un conjunto numeroso de objetos no etiquetados que serán utilizados también en el proceso de clasificación. En procesos de aprendizaje semi-supervisado, se pueden cometer errores que más tarde ocasionarán a su vez fallos en la clasificación de nuevos objetos, ya que aprender de datos clasificados incorrectamente afecta la funcionalidad de los algoritmos de clasificación, lo que demuestra la necesidad de aplicar estrategias de eliminación de objetos erróneamente clasificados en las bases de datos.

Muchos algoritmos de detección de ruido trabajan sobre conjuntos de datos estáticos (algoritmos de edición), éstos tienden a obtener un conjunto de prototipos eliminando valores atípicos (*outliers* en la literatura en inglés), y no tienen en cuenta los cambios que se pueden ocasionar con el transcurso del tiempo (García et al., 2012; Segata et al., 2010; Vázquez et al., 2005; Wilson and Martinez, 2000). Sin embargo, se deben tener en cuenta los cambios en la distribución de los datos que pueden ocurrir en el transcurso del tiempo, dando lugar a lo que se denomina *cambios de concepto* (conocido por *concept drift* en inglés) (ver (Jagadeesh et al., 2011; Klinkenberg, 2004)).

En (Zhu et al., 2008) aparece un método para eliminar ruido en un flujo de datos, utilizando técnicas estadísticas como el margen de varianza máxima, y hace una comparación entre las técnicas de Filtrado Local (FL), Global (FG) y, Local y Global (FLyG). Este método tiene tres limitaciones fundamentales: 1) necesita introducir un parámetro α que indica el número de objetos que considera como ruidosos en la base de datos, esto en general, es un problema ya que es imposible conocer a priori cuán contaminados están los datos, 2) se necesita evaluar la función que caracteriza el principio del margen de varianza máxima varias veces, lo que hace el proceso costoso y 3) los mejores resultados de su algoritmo se obtienen con el Filtrado FLyG, por lo que hace es necesario desarrollar tanto el filtrado local como el global.

Por ello, el problema a investigar en este trabajo es la insuficiencia en la calidad de la clasificación debido a la presencia de objetos mal etiquetados (ruido) en los conjuntos de entrenamiento. El objetivo de esta investigación es la creación de un algoritmo para detección y eliminación de ruido basado en criterios de vecindad y que tiene en cuenta cambios de concepto en el tiempo. Además, nuestra hipótesis radica en el perfeccionamiento de los métodos de clasificación basados en aprendizaje semi-supervisado con el uso del método de limpieza de ruido propuesto.

En el presente trabajo, se muestra una nueva estrategia para la detección de ruido en flujos de datos mediante criterios de vecindad para eliminar las limitaciones del método propuesto en (Zhu et al., 2008) empleando un conjunto de dos clasificadores (en la literatura científica en inglés suele llamarse ensemble). Además, se hace una propuesta de un esquema de aprendizaje semi-supervisado que utiliza en la etapa del filtrado de las muestras, el método de limpieza de ruido propuesto.

Materiales y métodos

Los métodos de limpieza de ruido en general, usan clasificadores entrenados de una porción de los datos de entrenamiento, para justificar las muestras excluidas (Jeatrakul et al., 2010; Jagadeesh et al., 2011; Segata et al., 2010; Li et al., 2007). Esto puede ser posible para datos estáticos, pero en flujos de datos, es necesario tener en cuenta que ellos están sujetos a cambios en las diferentes distribuciones, por lo que es necesario definir estrategias para lidiar con esta problemática. Se pueden efectuar tres variantes para filtrar el flujo de datos: 1) El Filtrado Local (FL) realiza la limpieza de los datos localmente dentro de cada bloque, sin necesitar ningún otro bloque de datos, 2) Filtrado Global (FG) que utiliza clasificadores entrenados desde múltiples bloques para identificar el ruido y/o 3) Filtrado Local y Global (FLyG) que tiene en cuenta los objetos ruidosos según cada una de las dos estrategias anteriores.

El flujo de datos se modela a través de los bloques de objetos etiquetados que se denotan $F_i(i = 1, 2, \ldots, H)$. Para todos los objetos de cada bloque se aplica una regla de clasificación, y se verifica si la etiqueta asignada al objeto coincide con la etiqueta que tiene originalmente, en caso que esto no ocurra, el objeto se considera ruidoso y es eliminado. Luego, un problema de clasificación en general puede ser descrito en la siguiente forma:

Definición 1 (Problema de clasificación) *Sean* $(X, \Theta) = \{(x_1, \theta_1), (x_2, \theta_2), \ldots, (x_N, \theta_N)\}$ *un conjunto de muestras etiquetadas (conjunto de entrenamiento) y* x *un nuevo objeto que se quiere asignar a una de las* M *clases* $C_1, C_2, \ldots, C_M$ *donde* $\theta_i \in \{C_1, C_2, \ldots, C_M\}$ $\forall i \in \{1, 2, \ldots, N\}$. *Si* $p(c_j|x)$ *son las probabilidades a posteriori de cada una de las clases, se debe asignar a* x *la etiqueta* c *que maximice el valor de la probabilidad*

anteriormente descrita, i.e.:

$$p(c_i|x) \geq p(c_j|x)\ \forall j \in \{1, 2, \ldots, N\}. \tag{1}$$

Una de las técnicas más empleadas para manejar los cambios de concepto son los conjuntos de clasificadores, mediante los cuales las salidas de varios clasificadores se combinan para tomar una decisión final. En nuestra estrategia se utiliza un producto de dos funciones p_1 y p_2 que representan estrategias de clasificación diferentes y luego se normaliza como se expresa a continuación:

$$p(c_i|x) = \frac{p_1(c_i|x) \cdot p_2(c_i|x)}{\sum_{j=1}^{M} p_1(c_j|x) \cdot p_2(c_j|x)}. \tag{2}$$

Dado un conjunto de entrenamiento E y x un objeto a clasificar, denotemos por U_x^k al conjunto de los k elementos de E más cercanos de acuerdo a la distancia Euclidiana a x (es también conocido como k-vecindad de x en E) unido al conjunto de los elementos de E que tienen a x incluido en su k-vecindad, entonces se define:

$$p_1(c_i|x) = \frac{\left|\{u \in U_x^k \mid \theta_u = c_i\}\right|}{|U_x^k|} \tag{3}$$

donde θ_u representa la clase de u y $|A|$ es el cardinal del conjunto A. Por otro lado, para definir la probabilidad p_2 se tuvo en cuenta la cercanía de x a las clases presentes, para ello se utilizó la siguiente fórmula:

$$p_2(c_i|x) = \frac{1}{\epsilon + d(x, C_i)}, \tag{4}$$

donde ϵ es número real positivo pequeño y $d(x, C_i) = \min_{y \in C_i} d(x, y)$.

Es precisamente en el filtrado global, donde se consideran los cambios de concepto en el tiempo, que significa que se decanten o ignoren todos los bloques anteriores a uno dado. La idea de esta propuesta se basa en el hecho que si hay distribuciones de los datos muy antiguas, es aconsejable no considerarlas, porque podría provocar criterios falsos acerca de la situación actual. Así, en un filtrado global se utiliza un parámetro β para determinar el número de bloques anteriores a F_i que formarán parte del conjunto de entrenamiento E:

$$E = \bigcup_{j=i-\beta}^{i-1} A_j. \tag{5}$$

El conjunto de entrenamiento se constituye con los elementos de los β bloques anteriores a F_i que ya han sido aceptados como no ruidosos (Fórmula 5). En la Figura 1 se resume el método de limpieza de ruido propuesto.

Algoritmo 1.	Limpieza de Ruido (LR)
Entrada:	Flujo de datos de las muestras etiquetadas F_i para $i = 1, \ldots, H$ Número de vecinos k
Salida:	Conjuntos de elementos aceptados (uno por cada bloque) A_i
Método:	Para $i = 1$ hasta H hacer: 1.1 Cargar el bloque F_i 1.2 Inicializar los conjuntos de elementos ruidosos y aceptados en vacío, i.e. $N_i = A_i = \emptyset$ 1.3 Para cada elemento x de F_i hacer: 1.3.1 Definir el conjunto U_x^k 1.3.2 Hallar la etiqueta c_i que maximice $p(c_i\|x)$ 1.3.3 Si $c_i = \theta_x$ entonces x se considera aceptado 1.3.3.1 $A_i = A_i \cup \{x\}$ 1.3.4 En caso contrario se considera ruido 1.3.4.1 $N_i = N_i \cup \{x\}$

Figura 1. Resumen del algoritmo de limpieza de ruido

Aprendizaje semi-supervisado con limpieza de ruido

En aprendizaje semi-supervisado, se tiene un conjunto pequeño E de muestras correctamente etiquetadas y un conjunto grande de objetos sin clase que necesitan ser etiquetados para luego ser utilizados como conjunto de entrenamiento, con el objetivo de clasificar nuevas muestras. Se considera que los objetos sin etiqueta llegan formando una secuencia de bloques $G_1, G_2, \ldots, G_H$ y con algún clasificador, se asignan etiquetas a los objetos, modelando de esta forma un flujo de datos $F_1, F_2, \ldots, F_H$.

Entre los elementos etiquetados de cada bloque existen algunos ruidosos debido a errores en la clasificación, lo que puede ocasionar la aparición de cambios de concepto. Una manera de detectar estos cambios de concepto es mediante la aplicación del método de detección de ruido utilizando únicamente los dos resultados más recientes como se explicó en la sección anterior. El esquema de aprendizaje semi-supervisado se muestra en la Figura 2. Con esta nueva propuesta, constituye el conjunto de entrenamiento actual el conjunto A_i obtenido, a diferencia de otros esquemas de aprendizaje semi-supervisado (Vázquez et al., 2008).

Por tanto, en dependencia de la funcionalidad del clasificador empleado en el paso 1.1, y de la aplicación del algoritmo de detección de ruido en el paso 1.2, así será la calidad del conjunto de entrenamiento A_i obtenido en cada etapa. Para desarrollar el paso 1.2 la primera vez, se selecciona un conjunto inicial A_0 de datos bien etiquetados que constituyen la experiencia existente acerca de la distribución de las clases, que sirve como conjunto de entrenamiento para etiquetar los objetos de F_1 y luego, decidir cuáles de ellos fueron mal

Revista Cubana de Ciencias Informáticas
Vol. 10, No. 2, abril-mayo, 2016
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 1-13

Algoritmo 2.	Aprendizaje semi-supervisado con LR
Entrada:	Conjunto de entrenamiento E Flujo de datos de muestras etiquetadas: Bloques G_i para $i = 1, \ldots, H$ Número de vecinos k
Salida:	Conjuntos de elementos aceptados (uno por cada bloque) A_i
Método:	Para $i = 1$ hasta H hacer: 1.1 Clasificar los elementos del bloque G_i obteniendo el bloque de objetos etiquetados F_i 1.2 Aplicar la estrategia de limpieza de ruido al conjunto F_i y se construye el nuevo conjunto A_i de objetos no ruidosos 1.3 El conjunto A_i es el conjunto de entrenamiento construido en cada etapa

Figura 2. Algoritmo de aprendizaje con limpieza de ruido

etiquetados. La segunda vez, el conjunto de entrenamiento será la unión de A_0 y A_1, para, desde entonces, utilizar los dos conjuntos de aceptados anteriores al bloque que se evalúa.

Resultados y discusión

En este epígrafe se muestran los resultados obtenidos de la experimentación realizada para verificar la efectividad del método propuesto. Para ello se utilizaron 8 bases de datos del repositorio UCI (Newman and Asuncion, 2007) y otras dos sintéticas creadas por los autores que fueron denominadas G4 y G6. G4 está formada por 4 modos gaussianos con poco solapamiento ya que estos concentran la mayor parte de sus puntos cerca de la media, por tanto, los puntos comunes a los demás modos son pocos en comparación con los que se encuentran en un radio dado alrededor de la media. G6 está compuesta por 6 modos gaussianos y en este caso sí existe un alto índice de solapamiento ya que uno de ellos tiene otros tres modos distribuidos cerca de su media. Son modos gaussianos con medias muy cercanas y por tanto muy solapados. En la Tabla 1 se exponen las principales características de estas colecciones de datos.

Se utilizó como medida de calidad *precisión* definida por $Pr = \frac{|R \cap \bar{R}|}{|R|}$, donde $\bar{R}$ es el conjunto de los objetos ruidosos detectados por el algoritmo y R es el conjunto de los ruidosos reales. Para simular el flujo de datos, cada una de las bases de datos fue dividida en 10 bloques de manera aleatoria, manteniendo la distribución de probabilidades de las clases, y de cada bloque del flujo de datos se seleccionó de manera aleatoria un porcentaje $\alpha \in \{10, 20, 30, 40, 50\}$ de objetos a los que se les alteró su correcta etiqueta de clase para simular la existencia de objetos ruidosos en la base de datos. Con cada valor de α se generaron cinco conjuntos diferentes de objetos

Tabla 1. Características básicas de las bases de datos usadas

Base de datos	**# Clases**	**# Instancias**	**# Atributos**	**Tipo**
Cancer	2	683	9	Reales
German	2	1000	24	
Diabetes	2	768	8	
Page	5	5473	10	
Phoneme	2	5404	5	
Wave	3	5000	21	
Pendigit	10	10992	16	
Spam	2	4601	57	
G4	4	4000	2	Sintéticas
G6	6	6000	2	

mal etiquetados. Los resultados son el promedio de las cinco ejecuciones realizadas del proceso indicado. El conjunto U_x^k para cada x se construyó tomando los valores de $k = 1, 3$.

En la Tabla 2 se muestran los porcentajes de precisión en la detección del ruido que se obtuvo para cada una de las bases de datos, es decir, el porcentaje de objetos verdaderamente ruidosos que el algoritmo detectó como ruidosos.

Nótese que en todos los casos el mayor porcentaje de aciertos se obtuvo con el Filtrado Global (en negrita), o sea, con el Filtrado FG se detecta el mayor porcentaje de objetos ruidosos, tanto cuando se considera un vecino como si se utilizan los tres vecinos más cercanos del objeto en análisis. Este es un resultado importante ya que disminuye el costo de la detección de ruido, pues no sería necesario realizar simultáneamente para cada bloque del flujo de datos un filtrado local y un filtrado global.

Obsérvese también, que cuando se tienen en cuenta los 3 vecinos de cada objeto (FG-3 o FLyG-3), los porcentajes son superiores, ya que se está utilizando una vecindad más amplia, lo cual garantiza una mayor precisión en la detección de objetos mal etiquetados.

Es de destacar, que sobre las bases de datos Cancer, Page, Pendigit y G4, con un 10 % de datos mal etiquetados, considerando tres vecinos, se hace una limpieza de al menos el 90 % de los objetos ruidosos con el filtrado FG. Para las bases de datos: Diabetes, Wave, Spam y G6, con un 10 % de objetos ruidosos se detecta un 80 % o más de los mismos. Sólo en el caso de la base de datos German, se obtuvieron porcentajes de detección de ruido inferiores.

Revista Cubana de Ciencias Informáticas
Vol. 10, No. 2, abril-mayo, 2016
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 1-13

Tabla 2. Precisión en la detección de ruido. a) Bases de datos Cancer, German, Diabetes, Page, Phoneme y b) Wave, Pendigit, Spam, G4, G6

BD	Método	10 %	20 %	30 %	40 %	50 %
Cancer	FG-1	**0,900**	**0,857**	**0,700**	**0,549**	**0,363**
	FLyG-1	0,843	0,738	0,536	0,344	0,158
	FG-3	**0,933**	**0,849**	**0,679**	**0,541**	**0,357**
	FLyG-3	0,903	0,772	0,537	0,344	0,154
German	FG-1	**0,722**	**0,633**	**0,579**	**0,505**	**0,431**
	FLyG-1	0,554	0,490	0,408	0,320	0,252
	FG-3	**0,762**	**0,656**	**0,598**	**0,503**	**0,438**
	FLyG-3	0,638	0,521	0,425	0,333	0,268
Diabetes	FG-1	**0,789**	**0,707**	**0,579**	**0,528**	**0,428**
	FLyG-1	0,631	0,548	0,385	0,327	0,233
	FG-3	**0,809**	**0,728**	**0,609**	**0,537**	**0,428**
	FLyG-3	0,686	0,595	0,434	0,363	0,245
Page	FG-1	**0,881**	**0,750**	**0,623**	0,492	0,368
	FLyG-1	0,807	0,611	0,444	0,289	0,174
	FG-3	**0,917**	**0,803**	**0,671**	0,514	0,374
	FLyG-3	0,886	0,708	0,523	0,317	0,177
Phoneme	FG-1	**0,808**	**0,713**	**0,585**	**0,483**	**0,395**
	FLyG-1	0,691	0,550	0,393	0,278	0,196
	FG-3	**0,846**	**0,748**	**0,629**	**0,499**	**0,389**
	FLyG-3	0,760	0,623	0,458	0,310	0,197

(a)

BD	Método	10 %	20 %	30 %	40 %	50 %
Wave	FG-1	**0,825**	**0,726**	**0,634**	**0,532**	**0,423**
	FLyG-1	0,733	0,602	0,461	0,331	0,224
	FG-3	**0,850**	**0,771**	**0,667**	**0,547**	**0,422**
	FLyG-3	0,800	0,688	0,525	0,353	0,218
Pendigit	FG-1	**0,771**	**0,686**	**0,590**	**0,492**	**0,395**
	FLyG-1	0,794	0,600	0,429	0,272	0,153
	FG-3	**0,923**	**0,808**	**0,671**	**0,513**	**0,352**
	FLyG-3	0,888	0,721	0,515	0,316	0,154
Spam	FG-1	**0,771**	**0,686**	**0,590**	**0,492**	**0,395**
	FLyG-1	0,628	0,512	0,403	0,294	0,215
	FG-3	**0,797**	**0,713**	**0,615**	**0,497**	**0,396**
	FLyG-3	0,685	0,560	0,442	0,309	0,219
G4	FG-1	**0,857**	**0,732**	**0,605**	**0,468**	**0,347**
	FLyG-1	0,758	0,594	0,412	0,252	0,141
	FG-3	**0,906**	**0,785**	**0,648**	**0,501**	**0,357**
	FLyG-3	0,877	0,699	0,483	0,297	0,151
G6	FG-1	**0,863**	**0,749**	**0,624**	**0,493**	**0,370**
	FLyG-1	0,780	0,612	0,426	0,284	0,165
	FG-3	**0,896**	**0,795**	**0,661**	**0,513**	**0,373**
	FLyG-3	0,854	0,709	0,492	0,314	0,170

(b)

Para $\alpha = 40$, se detectó alrededor del 50 % de los objetos ruidosos, mientras que para $\alpha = 50$ fueron eliminados alrededor del 40 % de los objetos mal etiquetados, siempre que se aplica el filtrado global, lo que no ocurre con el filtrado FLyG con el cual los porcentajes de detección de ruido son mucho menores.

Es válido aclarar, que si cerca de la mitad de los ejemplos de la base de datos son ruidosos, hay una gran confusión entre los objetos ruidosos y los objetos con una etiquetada de clase correcta, lo que hace extremadamente difícil determinar cuáles son los objetos realmente ruidosos.

Influencia de la limpieza de ruido en aprendizaje semi-supervisado

Se evaluó la influencia de la estrategia de limpieza de ruido en un esquema de aprendizaje semi-supervisado utilizando los conjuntos de objetos aceptados como conjuntos de entrenamiento. Se tomó como conjunto de prueba (*test*) el 10 % de cada base de datos. El criterio de selección de este sub-conjunto fue mediante la selección aleatoria de una muestra del 10 % de cada una de las clases existentes. Este conjunto fue utilizado para determinar el porcentaje de clasificación correcta que los objetos aceptados como no ruidosos (de entre

el restante 90 % dividido en flujos) proporcionan al etiquetar los ejemplos del conjunto de prueba. En este experimento, se emplearon los conjuntos de objetos aceptados del filtrado con las estrategias: FG y FLyG como conjunto de entrenamiento para clasificar el conjunto de prueba y comparamos los resultados obtenidos.

En la Tabla 3 se muestran los resultados de este experimento, con $k = 3$ siendo el valor de mejores resultados en la detección de ruido. Se agregaron, además, dos experimentos cuyos resultados aparecen en las columnas nombradas SF (Sin Filtrado) y FP (Filtrado Perfecto), que significan: todos los bloques antes de ser filtrados, y, todos los bloques luego de haber eliminado el total de los objetos ruidosos, respectivamente. En negrita, marcamos los valores más significativos (mayores) del porcentaje de clasificación correcta. La columna BD significa base de datos, el símbolo α representa el porcentaje de ruido presente. Los resultados indican que en un esquema de aprendizaje, en el que se etiquetan objetos desconocidos, hasta un 20 % de error en el etiquetado puede ser *corregido* o eliminando un porcentaje alto de los objetos ruidosos, y así, los conjuntos de entrenamiento tendrían mayor calidad.

Puede verse, además, que cuando hay un 10 % de objetos ruidosos, sobre las bases de datos: Cancer, German, Diabetes, G4, G6, Page, Wave, Pendigit y Spam, se obtiene un porcentaje de clasificación correcta superior o similar al que se obtiene cuando se realiza un filtrado perfecto. Esto significa, que es útil emplear la estrategia de detección de ruido para construir conjuntos de entrenamiento. Sólo con las bases de datos Page y Phoneme quedaron los porcentajes por debajo de los del filtrado perfecto, aunque sin una marcada diferencia en el caso del filtrado FG.

Cuando existe un 20 % de objetos ruidosos en las bases de datos, también los resultados alcanzados con el método propuesto para la detección de ruido son buenos. Por ejemplo, sobre las bases de datos: German, Diabetes, G6 y Wave, los porcentajes de clasificación correcta son superiores o similares a los obtenidos con un filtrado perfecto. Para el resto de las bases de datos, los porcentajes se pueden considerar adecuados por su significado, ya que al realizar un filtrado se logra eliminar objetos ruidosos y disminuir el tamaño del conjunto de entrenamiento. Se pueden destacar los resultados que se han obtenido con las bases de datos: Cancer, G4, G6, Page, Pendigit, para las cuales, el porcentaje de clasificación correcta que proporcionan es igual o superior al 90 % cuando hay un 20 % o menos de error en las etiquetas de los objetos que forman los bloques. Esto garantiza que la estrategia de detección de ruido, es capaz de filtrar los bloques del flujo de datos de manera que los objetos aceptados como no ruidosos puedan ser empleados para clasificar objetos nuevos.

Obsérvese además, la diferencia de los porcentajes obtenidos después de la limpieza con relación a los obtenidos si no se aplica nuestra estrategia. Para la mayoría de las bases de datos, el porcentaje de clasificación correcta que se obtiene del conjunto de entrenamiento con ruido (sin aplicar el método de filtrado que aquí se propone) es por lo menos un 10 % menor que cuando se utilizan los bloques filtrados.

Tabla 3. Porcentaje de clasificación correcta de los conjuntos de entrenamiento ($k = 3$), a) Bases de datos Cancer, German, Diabetes, G4, G6 y b) Page, Phoneme, Wave, Pendigit, Spam

BD	α	SF	FG	FLyG	FP
Cancer	10	0,871	0,964	**0,966**	0,956
	20	0,761	0,924	**0,940**	0,939
	30	0,680	0,867	0,918	**0,944**
	40	0,571	0,710	0,765	**0,951**
	50	0,503	0,528	0,524	**0,946**
German	10	0,597	0,662	**0,672**	0,637
	20	0,533	0,636	**0,637**	0,627
	30	0,500	0,595	0,603	**0,638**
	40	0,463	0,536	0,532	**0,638**
	50	0,418	0,432	0,428	**0,616**
Diabetes	10	0,616	**0,701**	0,694	0,665
	20	0,564	0,651	**0,660**	0,649
	30	0,520	0,609	0,620	**0,679**
	40	0,461	0,540	0,544	**0,668**
	50	0,408	0,421	0,433	**0,644**
G4	10	0,887	0,977	**0,987**	**0,987**
	20	0,791	0,931	0,965	**0,986**
	30	0,694	0,838	0,896	**0,987**
	40	0,600	0,693	0,746	**0,984**
	50	0,494	0,503	0,498	**0,989**
G6	10	0,829	0,930	**0,938**	0,919
	20	0,742	0,890	**0,924**	0,918
	30	0,650	0,805	0,860	**0,919**
	40	0,560	0,657	0,706	**0,920**
	50	0,474	0,480	0,481	**0,928**

(a)

BD	α	SF	FG	FLyG	FP
Page	10	0,853	0,935	0,935	**0,938**
	20	0,754	0,895	0,915	**0,936**
	30	0,667	0,806	0,855	**0,935**
	40	0,565	0,664	0,700	**0,929**
	50	0,480	0,484	0,479	**0,935**
Phoneme	10	0,744	0,808	0,801	**0,813**
	20	0,666	0,774	0,781	**0,802**
	30	0,604	0,712	0,734	**0,804**
	40	0,525	0,595	0,620	**0,799**
	50	0,454	0,467	0,470	**0,792**
Wave	10	0,705	0,798	**0,804**	0,773
	20	0,629	0,764	**0,786**	0,762
	30	0,557	0,699	0,746	**0,759**
	40	0,493	0,592	0,637	**0,759**
	50	0,419	0,453	0,459	**0,763**
Pendigit	10	0,876	0,966	0,966	**0,976**
	20	0,783	0,928	0,948	**0,974**
	30	0,687	0,837	0,882	**0,971**
	40	0,588	0,684	0,728	**0,968**
	50	0,489	0,485	0,476	**0,965**
Spam	10	0,665	**0,725**	0,714	0,717
	20	0,606	0,699	0,697	**0,714**
	30	0,544	0,626	0,642	**0,701**
	40	0,488	0,544	0,559	**0,695**
	50	0,429	0,448	0,452	**0,687**

(b)

Por ejemplo, sobre la base de datos cáncer, con un 10 % de objetos ruidosos, sin aplicar limpieza de ruido, el porcentaje de clasificación correcta que proporciona el conjunto de entrenamiento es de un 87 %. Sin embargo, después de haber detectado objetos ruidosos, el porcentaje de clasificación correcta aumenta hasta más de un 96 %.

Los resultados, demuestran, además, que la estrategia de tener en cuenta los cambios de concepto proporciona la construcción de conjuntos de entrenamiento adecuados sin necesidad de utilizar todos los objetos del flujo

de datos. El hecho de obtener buenos resultados cuando se tienen en cuenta los cambios de concepto, además de la utilidad en sí que tiene este problema en la actualidad, es importante ya que se puede ir eliminando información no relevante en el contexto actual. Desde el punto de vista computacional es conveniente, ya que para realizar una clasificación, no es necesario utilizar todos los objetos que ya han sido procesados, sino los de la última generación.

También se puede mencionar el hecho de que cuando hay un porcentaje de error de 40 % o 50 % se detecta menor cantidad de objetos ruidosos, causado por la incertidumbre en la veracidad de las etiquetas de clase existe en este caso, pues habría casi el mismo número de objetos bien etiquetados que mal etiquetados. Obviamente, esto influye en los porcentajes de clasificación correcta.

Conclusiones

En este trabajo se ha mostrado una nueva estrategia para la detección y limpieza de ruido en flujos de datos, empleando criterios de vecindad. En la nueva estrategia se utiliza un *conjunto* de dos clasificadores, para combinar los resultados que cada uno aporta en la etapa de clasificación. Este método se enfoca en el problema de la presencia de cambios de concepto en el tiempo. El método propuesto detecta automáticamente todos los objetos que considera ruidosos, no se limita a un porcentaje α (este valor sólo se utiliza para simular la existencia de objetos ruidosos en el flujo de datos). Se emplea una estrategia muy simple (vecinos más cercanos). Se realizaron los experimentos siguiendo los esquemas de los filtrados: FG y FLyG debido a que con el filtrado local (FL) no se tiene en cuenta los cambios de concepto en el tiempo.

Como medida para establecer la calidad del proceso de limpieza de ruido se utilizó la precisión, analizándose el porcentaje de objetos ruidosos que el algoritmo detecta y la calidad de los bloques luego del proceso de limpieza, para ser utilizados como conjuntos de entrenamiento en la clasificación de nuevas muestras. De las dos estrategias de filtrado, los resultados en el procesamiento de los patrones demuestran que el filtrado FG es suficiente para detectar los objetos ruidosos. Esto es importante ya que así el proceso es menos costoso debido a que no hay que realizar el filtrado local.

Entre los valores del parámetro k, para detección de ruido el más efectivo resultó $k = 3$, lo cual demuestra que para detectar los objetos ruidosos es más conveniente verificar las etiquetas de otros objetos que rodean al que se está analizando, no sólo su vecino más cercano. Este hecho se observa en la Tabla 2 ya que FG-3 y FLyG-3 tienen siempre porcentajes de detección de ruido más altos que FG-1 y FLyG-1 respectivamente.

Otra cuestión a destacar es que para tener en cuenta los cambios de concepto, sólo se emplearon dos bloques anteriores al analizado en cada etapa, esto contribuye con un ahorro computacional importante, además del

hecho en sí que es tener en cuenta nada más los resultados más actuales para detectar nuevos objetos ruidosos o para clasificar objetos correctamente, desechando información fuera del contexto actual. Los porcentajes alcanzados en cuanto al filtrado de objetos ruidosos, demuestran la validez del método aplicado, ya que se detecta un 80 % o más de individuos mal etiquetados cuando hay hasta un 20 % de error de clasificación. La importancia de este hecho está en la posibilidad de emplear el método en esquemas de aprendizaje semi-supervisado.

En cuanto a la calidad como conjuntos de entrenamiento de los bloques filtrados, los resultados en los casos de menos de un 30 % de ruido son positivos. Los mejores resultados se obtienen cuando hay un 10 % de ruido, ya que los porcentajes son superiores a los que se obtienen con el filtrado perfecto y se demuestra que con un menor número de objetos se obtienen porcentajes de clasificación satisfactorios.

Como una aplicación de los resultados obtenidos en el esquema de detección de ruido en flujos de datos, se propuso un algoritmo de aprendizaje semi-supervisado para desechar los objetos ruidosos producto de la etapa de clasificación.

Referencias

Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien. *Semi-supervised learning*. MIT press Cambridge, 2006.

Salvador García, Joaquín Derrac, José Ramón Cano, and Francisco Herrera. Prototype selection for nearest neighbor classification: Taxonomy and empirical study. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(3): 417–435, 2012.

R. P. Jagadeesh, Chandra Bose, Wil M. P. van der Aalst, Indre Zliobaite, and Mykola Pechenizkiy. Handling concept drift in process mining. In *Advanced Information Systems Engineering - 23rd International Conference, CAiSE 2011, London, UK, June 20-24, 2011. Proceedings*, pages 391–405, 2011.

Piyasak Jeatrakul, Kok Wai Wong, and Chun Che Fung. Data cleaning for classification using misclassification analysis. *JACIII*, 14(3):297–302, 2010.

Charles W. Kalish, Timothy T. Rogers, Jonathan Lang, and Xiaojin Zhu. Can semi-supervised learning explain incorrect beliefs about categories? *Cognition*, 120(1):106–118, 2011.

Ralf Klinkenberg. Learning drifting concepts: Example selection vs. example weighting. *Intelligent Data Analysis*, 8(3):281–300, 2004.

Yunlei Li, Lodewyk F. A. Wessels, Dick de Ridder, and Marcel J. T. Reinders. Classification in the presence of class noise using a probabilistic kernel fisher method. *Pattern Recognition*, 40(12):3349–3357, 2007.

Qiuhua Liu, Xuejun Liao, Hui Li, Jason R. Stack, and Lawrence Carin. Semisupervised multitask learning. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(6):1074–1086, 2009.

David Newman and Arthur Asuncion. University of California Irvine UCI- Machine Learning repository, 2007.

Mohammad H. Rohban and Hamid R. Rabiee. Supervised neighborhood graph construction for semi-supervised classification. *Pattern Recognition*, 45(4):1363–1372, 2012.

Nicola Segata, Enrico Blanzieri, Sarah Jane Delany, and Padraig Cunningham. Noise reduction for instance-based learning with a local maximal margin approach. *Journal of Intelligent Information Systems*, 35(2): 301–331, 2010.

Burr Settles. Active learning literature survey. *University of Wisconsin, Madison*, 52(55-66):11, 2010.

Fernando Vázquez, J. Salvador Sánchez, and Filiberto Pla. A stochastic approach to wilson's editing algorithm. In *Pattern Recognition and Image Analysis*, pages 35–42. Springer, 2005.

Fernando Vázquez, José Salvador Sánchez, and Filiberto Pla. Learning and forgetting with local information of new objects. In *Progress in Pattern Recognition, Image Analysis and Applications, 13th Iberoamerican Congress on Pattern Recognition, CIARP 2008, Havana, Cuba, September 9-12, 2008. Proceedings*, pages 261–268, 2008.

D. Randall Wilson and Tony R. Martinez. Reduction techniques for instance-based learning algorithms. *Machine Learning*, 38(3):257–286, 2000.

Yan Zhou and Sally A. Goldman. Democratic co-learning. In *16th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2004), 15-17 November 2004, Boca Raton, FL, USA*, pages 594–602, 2004.

Xingquan Zhu, Peng Zhang, Xindong Wu, Dan He, Chengqi Zhang, and Yong Shi. Cleansing noisy data streams. In *Proceedings of the 8th IEEE International Conference on Data Mining (ICDM 2008), December 15-19, 2008, Pisa, Italy*, pages 1139–1144, 2008.