Rev. Fac. Ing. Univ. Antioquia N. º71 pp. 37-47, junio, 2014

Detección de ruido en aprendizaje semi-supervisado con el uso de flujos de datos

Noise detection in semi-supervised learning with the use of data streams

Damaris Pascual González[1], Fernando D. Vázquez Mesa[1], J. Salvador Sánchez[2], Filiberto Pla[2]*

[1]Facultad de Ciencias Económicas y Empresariales, Universidad de Oriente, Av. Patricio Lumumba s/n.CP. 90500. Santiago de Cuba, Cuba.

[2]Departamento de Lenguajes y Sistemas Informáticos, Universitat Jaume I. CP. 12071. Castellón, España.

(Recibido el 19 de febrero de 2013. Aceptado el 13 de marzo de 2014)

Resumen

A menudo, es necesario construir conjuntos de entrenamiento. Si disponemos solamente de un número reducido de objetos etiquetados y de un conjunto numeroso de objetos no etiquetados, podemos construir el conjunto de entrenamiento simulando un flujo de datos no etiquetados de los cuales es necesario aprender para poder incorporarlos al conjunto de entrenamiento. Con el objetivo de prevenir que se deterioren los conjuntos de entrenamiento que se obtienen, en este trabajo se propone un esquema que tiene en cuenta el *concept drift*, ya que en muchas situaciones la distribución de las clases puede cambiar con el tiempo. Para clasificar los objetos no etiquetados hemos empleado un *ensemble* de clasificadores y proponemos una estrategia para detectar el ruido.

----------*Palabras clave: Concept drift,* flujo de datos, datos no etiquetados, limpieza de ruido

Abstract

Often, it is necessary to construct training sets. If we have only a small number of tagged objects and a large group of unlabeled objects, we can build the training set simulating a data stream of unlabelled objects from which it is necessary to learn and to incorporate them to the training set later. In order to prevent deterioration of the training set obtained, in this work we propose a scheme that takes into account the *concept drift*, since in many situations

* Autor de correspondencia: teléfono: + 53 + 22 + 635330, fax: + 53 + 22 + 632689, correo electrónico: dpascual@eco.uo.edu.cu (D. Pascual)

the distribution of classes may change over time. To classify the unlabelled objects we have used an ensemble of classifiers and we propose a strategy to detect the noise after the classification process.

----------*Keywords:* Concept drift, data streams, unlabeled data, noise cleaning

Introducción

Los conceptos del mundo real muchas veces no son estables, sino que cambian con el tiempo. Un ejemplo típico son las reglas de predicción del tiempo que pueden variar radicalmente con la temporada. Otro ejemplo son los patrones de las preferencias de compra de los clientes que pueden cambiar, dependiendo del día de la semana, la disponibilidad de alternativas, la tasa de inflación, etc.

A menudo, la causa del cambio está oculta y/o no se conoce a priori. Este problema se conoce con el nombre de contexto oculto (*hidden context*), haciendo que la tarea que deben llevar a cabo los algoritmos de aprendizaje sea mucho más compleja. Los cambios en el contexto que están ocultos pueden inducir, de alguna manera, un cambio radical en el concepto final, dando lugar a lo que se denomina cambio de concepto (*concept drift*) [1].

Un problema difícil en el manejo del *concept drift* es saber distinguir entre éste y el posible ruido presente en los objetos mal etiquetados. Algunos algoritmos pueden reaccionar de forma exagerada al ruido, erróneamente interpretar como ruido el *concept drift*, mientras que otros pueden ser muy robustos al ruido. Un algoritmo de aprendizaje ideal debe combinar la robustez al ruido y la sensibilidad al *concept drift* [2].

Los sistemas de clasificación supervisados dependen de una muestra de entrenamiento que sea lo suficientemente representativa del problema que se pretende resolver. Este conjunto de entrenamiento debe ser preparado con anterioridad por un experto humano, quien elige un conjunto de objetos representativos y los atributos que logren distinguirlos. Por otro lado, si en el entorno donde el clasificador ha sido entrenado aparece el *concept drift*, el clasificador deberá ser entrenado nuevamente y será necesario recurrir al experto humano para que reconstruya el conjunto de entrenamiento [3]. En general, este proceso resulta complicado y muy costoso, de manera que no siempre es posible lograr un conjunto de entrenamiento suficientemente bueno. En la práctica, resulta mucho más sencillo obtener muestras no etiquetadas por lo que se hace necesario diseñar métodos de aprendizaje que permitan utilizar tanto muestras etiquetadas como no etiquetadas en el proceso de construcción de un conjunto de entrenamiento, teniendo en cuenta además el *concept drift* [4]

En la literatura científica, este tipo de aprendizaje recibe el nombre de aprendizaje semi-supervisado o parcialmente supervisado [5]. El aprendizaje semi-supervisado trata este problema usando una cantidad grande de objetos sin etiquetar, junto con un conjunto (probablemente pequeño) de objetos etiquetados, para construir clasificadores mejores. Su principal ventaja es que requiere un menor esfuerzo humano y, además, tiene en cuenta los cambios producidos por el *concept drift* que pueden ocasionar la incorporación de objetos mal etiquetados.

Los modelos generativos son quizás los métodos de aprendizaje semi-supervisados más antiguos referenciados en la literatura consultada [6-10]. El auto-entrenamiento es otro enfoque en el aprendizaje semi-supervisado que podemos encontrar en los trabajos de [11-15].

Otra técnica de aprendizaje semi-supervisado es el *co-training* propuesto por [16, 17], cuya continuación se puede ver en los trabajos de [9] y [18, 19]. Las Máquinas de Vector Soporte Transductivas (TSVMs) constituyen una extensión de las Máquinas de Vector Soporte estándar [20],

las cuales trabajan tanto con objetos etiquetados como con muestras no etiquetadas [21, 22].

Trabajos más recientes pueden verse en [23] que emplea técnicas de aprendizaje activo [24, 25] o que utiliza un algoritmo de propagación de etiquetas semi-supervisado basado en grafos.

En el presente trabajo, presentamos dos algoritmos de aprendizaje semi-supervisado para construir un conjunto de entrenamiento que tiene en cuenta el *concept drift*. En el primero, todos los conjuntos de datos son etiquetados utilizando el conocimiento base (CB) como conjunto de entrenamiento, mientras que los objetos etiquetados y filtrados se van utilizando para la detección de los objetos ruidosos en los nuevos subconjuntos que van llegando. En el segundo algoritmo, los objetos etiquetados y filtrados van enriqueciendo el conjunto de entrenamiento para etiquetar a los nuevos subconjuntos que van llegando, mientras que para detectar el ruido se emplea un conjunto fijo de datos bien etiquetados (CR).

Este trabajo esta estructurado de la siguiente forma. Inicialmente, se presenta la estrategia utilizada para detectar los objetos mal clasificados; a continuación, se muestran los dos esquemas de aprendizaje semi-supervisado. Los resultados experimentales que avalan nuestras dos estrategias son descritos en la sección de experimentos y, finalmente, se comentan las principales conclusiones.

Detección de ruido

En problemas de aprendizaje semi-supervisado, se dispone de un conjunto pequeño de datos etiquetados y un conjunto (posiblemente) grande de datos no etiquetados, los cuales pueden ser coleccionados con el tiempo con el objetivo de enriquecer el conjunto de entrenamiento inicial. Una de las estrategias empleadas para realizar esta tarea consiste en clasificar los datos no etiquetados e incorporarlos a la muestra de entrenamiento. De esta manera, el conjunto de entrenamiento se irá nutriendo de los cambios que estos objetos pudieran provocar al mismo.

Una de las dificultades de esta estrategia es que algunos ejemplos pueden haber sido mal etiquetados, lo que obviamente podría producir errores en la clasificación de los nuevos objetos que irán llegando al sistema en el futuro. Otra dificultad añadida en este proceso, se refiere al hecho de que debemos distinguir entre el ruido y el *concept drift*, lo que supone una tarea de gran complejidad, pero sumamente importante [26]. Para detectar los objetos ruidosos en un conjunto de datos etiquetados, emplearemos una combinación (*ensemble*) de clasificadores, formada por dos clasificadores muy simples: la regla del vecino más cercano (NN) y el basado en centroides. El motivo de utilizar estos dos clasificadores viene dado por su simplicidad y por el buen comportamiento que generalmente presentan en un gran número de problemas reales.

Dado un conjunto de entrenamiento dividido en p clases c_1, c_2,..., c_p, se hallan los centroides de cada una de las clases, los cuales denotaremos por $Cent_i$ ($i = 1,\ldots, p$). El clasificador basado en centroides que proponemos puede ser expresado por medio de probabilidades a posteriori en la forma de la ecuación (1):

$$p_{cent}(c_i / x) = \frac{f_i(x)}{\sum_{j=1}^{p} f_j(x)} \qquad (1)$$

que se obtiene al normalizar la ecuación (1a)

$$f_i(x) = \frac{d(x, NCent)}{d(x, Cent_i)} \qquad (1a)$$

donde *NCent* denota el centroide más cercano a x, es decir, $d(x, NCent) = \min_{1 \le i \le p}\{d(x, Cent_i)\}$. De este modo, cuando $Cent_i$ coincide con *NCent*, $f_i(x)$ será igual a 1, mientras que en cualquier otro caso será menor que 1. Así, el objeto será asignado a la clase de su centroide más cercano.

Hacemos la aclaración que si $d(x, Cent_i)$ fuese cero para alguna i, asignaremos el valor 1 a la función $f_i(x)$ relacionada con dicha clase.

Cuando utilizamos el clasificador no paramétrico del vecino más cercano, la probabilidad a posteriori será igual a 1 si el vecino más cercano de x pertenece

a dicha clase y será igual a cero en caso contrario. Como el objetivo es detectar el ruido, nos interesa, además del vecino más cercano de *x* en el conjunto de entrenamiento, conocer el entorno que rodea al punto *x*, por lo que buscamos también a todos los objetos del conjunto de entrenamiento (*TS*, *Training Set*) que tienen a *x* como vecino más cercano y tenemos en cuenta las etiquetas de éstos, tal como se expresa en la ecuación (2).

$$p_v(c_i / x) = \frac{g_i(x)}{\sum_{j=1}^{p} g_j(x)} \qquad (2)$$

que se obtiene al normalizar la ecuación (2a)

$$g_i(x) = \begin{cases} \frac{1}{N_x + 1}\left(1 + \sum_{\substack{x \in V_{x'} \\ x' \in TS}} 1_{\{\theta_{x'} = c_i\}}\right) & si \quad \theta_v = c_i \\ \frac{1}{N_x} \sum_{\substack{x \in V_{x'} \\ x' \in TS}} 1_{\{\theta_{x'} = c_i\}} & en\ otro\ caso \end{cases} \qquad (2a)$$

donde θ_v es la clase del vecino más cercano de *x*, N_x es el número de objetos que tienen a *x* como vecino más cercano y $V_{x'}$ es la vecindad de *x'* (sólo contiene al vecino más cercano de *x'*). Por tanto, si tenemos en cuenta tanto al vecino más cercano de *x* como a todos los objetos que tienen a *x* como su vecino más cercano, para cada *x*, $g_i(x)$ es la razón entre el número de objetos que cumplen una de las dos condiciones: o es el vecino más cercano de *x* o tiene a *x* como vecino más cercano que pertenecen a la clase c_i, sobre el total de esos objetos. Al igual que para la ecuación 1, es necesario normalizar para que p_v denote una probabilidad a posteriori.

Finalmente, para analizar la pertenencia o no de un objeto a alguna de las clases, utilizamos, para cada clase c_i, una función igual al producto de las dos funciones antes mencionadas a través de la expresión (3).

De este modo, si *x* pertenece a una zona en la que la mayoría de los puntos son de la clase c_i y el centroide más cercano es el de la clase c_i, es de esperar que *x* pertenezca a la clase c_i. El número *e* es un valor positivo pequeño que se utiliza para evitar que sea cero la función (3) cuando $p_v(c_i / x)$ es igual a cero. Las probabilidades a posteriori que nos permitirán clasificar a los puntos *x* se obtienen normalizando la ecuación (3) de acuerdo a las clases.

Ahora, nuestra estrategia para detectar el ruido será: para cada *x*, una vez evaluada la expresión (3), si la etiqueta de la clase c_i cuya probabilidad es la máxima es diferente de la etiqueta original de *x*, se considerará que éste es un objeto ruidoso.

Aprendizaje semi-supervisado

Muchos problemas de la vida real se caracterizan por constar de una gran cantidad de datos, pero la mayoría de ellos sin etiquetas de clase, por lo que surge la necesidad de crear sistemas automáticos capaces de aprender también de datos no etiquetados sin requerir la participación de un experto para etiquetarlos, puesto que ello haría que el proceso fuera mucho más costoso.

En el presente trabajo, proponemos dos métodos de aprendizaje semi-supervisado [5] capaces de construir un conjunto de entrenamiento que tenga presente el *concept drift*, partiendo de un conjunto pequeño de datos etiquetados y un conjunto grande de datos no etiquetados. La idea que se sigue en general es, dado un conjunto pequeño de datos etiquetados, utilizar algún clasificador no paramétrico para asignar una clase a los datos no etiquetados y, posteriormente, aplicar una estrategia de detección de objetos mal etiquetados (ruido). Con ello, al unir los objetos del conjunto de entrenamiento inicial con los objetos considerados como no ruidosos por el sistema, se persigue mejorar la calidad del conjunto resultante y que en éste se tenga en

$$\overline{p}(c_i / x) = \begin{cases} p_v(c_i / x) * p_{cent}(c_i / x) & si \quad p_v(c_i / x) > 0 \\ \varepsilon * p_{cent}(c_i / x) & en\ otro\ caso \end{cases} \qquad (3)$$

cuenta los posibles cambios que introducen los nuevos objetos que se han ido incorporando, es decir, que se tenga en cuenta el *concept drift*.

En nuestra estrategia, contamos con un pequeño conjunto de entrenamiento inicial al que llamamos conocimiento base (CB) y un conjunto grande de objetos no etiquetados que vamos a dividir en M bloques para simular un flujo de datos en el proceso de construcción del conjunto de entrenamiento, es decir, consideraremos que cada cierto tiempo disponemos de un conjunto de datos no etiquetados con los que queremos enriquecer el conocimiento existente hasta ese momento. A los diferentes conjuntos de datos no etiquetados los denotaremos por G_i, i=1,..., M y llamaremos CA (conocimiento actual) a cada uno de los conjuntos de entrenamiento obtenidos en cada etapa. Además, en el algoritmo, denotaremos por $Acep_i$ al subconjunto de objetos de G_i que queda después de aplicar la estrategia de detección de ruido. Ya que en el conjunto de entrenamiento inicial se supone que los objetos están bien etiquetados, ese conjunto se representa como $Acep_0$.

En la primera variante de nuestra estrategia para determinar cuáles son los objetos mal clasificados, consideramos los conjuntos de objetos ya etiquetados como parte de un flujo de datos formado por los subconjuntos G_i que van llegando, con la diferencia que ahora los objetos no están etiquetados, pero solucionamos esto aplicando una regla de clasificación que asignará a cada objeto una etiqueta que, posteriormente, será analizada en la etapa de filtrado. Para diferenciar el conjunto G_i de objetos sin etiquetas del mismo conjunto pero con sus elementos etiquetados, a este último lo llamamos G_i-*class*. Como estamos utilizando un esquema similar al del flujo de datos, también suponemos que son subconjuntos que van llegando con el tiempo, por lo que la distribución de los datos no siempre es la misma, lo que puede ocasionar la aparición del *concept drift.* Por tanto, sólo consideraremos en el filtrado los dos últimos subconjuntos de datos aceptados como no ruidosos anteriores al que en ese momento se está analizando y son rechazados los conjuntos de objetos aceptados resultantes del filtrado en las primeras etapas del aprendizaje.

Algoritmo 1:

1- Hacemos CA = CB (CA denota el conocimiento actual, que representa el conjunto de entrenamiento que se construye, TS).

2- Inicializar $Acep_0$ = CB

3- Para cada i = 1,..., M:

3.1-Se clasifican los objetos del bloque G_i con CB como conjunto de entrenamiento y se obtiene el conjunto G_i-*class*

3.2-Hacer CR_i = $Acep_{i-2}$ È $Acep_{i-1}$. (CR_i conocimiento para detectar el ruido en G_i-*class*), los subíndices i-1 e i-2 denotan, que se están considerando solamente los objetos aceptados de los dos bloques anteriores al bloque i-ésimo.

3.3-Se realiza el filtrado de las muestras de G_i-*class* utilizando como conjunto de entrenamiento en la estrategia de detección de ruido a CR_i, obteniendo como resultado el conjunto $Acep_i$

3.4-CA = CA È $Acep_i$ para obtener el nuevo conocimiento actual CA.

3.5-Conjunto de entrenamiento TS = CA

Es importante señalar en el paso 3.2 (estrategia para detectar ruido) que, como sólo se consideran los objetos aceptados de los dos bloques anteriores al que se está analizando, cuando se aplica la estrategia de detección de ruido en el primer bloque G_1-*class*, sólo hay un conjunto de objetos aceptados que es $Acep_0$ y, por tanto, únicamente se dispone de ese conjunto pero no de dos anteriores; en otras palabras, CR_1 = $Acep_0$ = CB. Cuando se analiza el segundo bloque G_2-*class*, tenemos $Acep_0$ y los objetos del conjunto $Acep_1$, que son los dos anteriores hasta ese momento, es decir, CR_2 = $Acep_0$ È $Acep_1$. A partir del tercer bloque G_3-*class*, ya comenzamos a tomar los dos anteriores, es decir, CR_3 = $Acep_1$ È $Acep_2$, desechando entonces a los objetos de $Acep_0$.

La segunda variante empleada se diferencia de la primera en el hecho de que, en lugar de tener inicialmente un único conjunto de datos etiquetados, tendremos dos: a uno de ellos lo denominaremos igual que en la estrategia anterior, CB, y al otro lo denominaremos CR (conocimiento para detectar el ruido). La diferencia fundamental radica en que el conocimiento base inicial con el que se clasifican cada uno de los bloques del flujo va variando con el tiempo, nutriéndose de muestras que, después de ser etiquetadas y filtradas, son incorporadas al conjunto de entrenamiento. Con la incorporación de estas muestras, logramos la presencia de posibles cambios que puedan producirse en los datos. El segundo conjunto que aparece en esta etapa (CR), lo utilizaremos para filtrar los conjuntos de datos que se van obteniendo en el aprendizaje, considerándose además que todos los objetos de este conjunto estarán bien etiquetados y que este conjunto permanecerá fijo a lo largo de todo el proceso.

Algoritmo 2:

1- Hacemos CA = CB (CA denota el conocimiento actual que representa el conjunto de entrenamiento actual).

2- Para cada $i = 1,..., M$:

2.1-Se clasifican los objetos de G_i con CA como conjunto de entrenamiento y se obtiene el conjunto G_i-*class*

2.2-Se realiza el filtrado de las muestras de G_i-*class* teniendo en cuenta el conjunto fijo de objetos CR, el conjunto resultante se denotará por $Acep_i$.

2.3-CA = CA È $Acep_i$ para obtener el nuevo conocimiento actual CA, y se vuelve al segundo paso.

2.4-Conjunto de entrenamiento TS = CA

Haciendo un resumen, podemos decir que, en el caso de la primera variante, todos los conjuntos de datos son etiquetados utilizando el CB como conjunto de entrenamiento, mientras que los objetos etiquetados y filtrados se van utilizando para la detección de los objetos ruidosos en los nuevos subconjuntos que van llegando. En el caso de la segunda variante, los objetos etiquetados y filtrados van enriqueciendo el conjunto de entrenamiento para etiquetar a los nuevos subconjuntos que van llegando, mientras que para detectar el ruido se emplea un conjunto fijo de datos bien etiquetados CR.

Resultados experimentales

Los experimentos fueron realizados con algunas bases de datos sintéticas y otras tomadas del UCI Machine Learning Database Repository. Como clasificador supervisado para etiquetar los datos no etiquetados en el paso 3.1 (algoritmo 1) y 2.1 (algoritmo 2), tomamos el basado en la regla *k*-NN con $k = 3$. Para medir la calidad de los conjuntos de entrenamiento que se van construyendo mediante los dos algoritmos propuestos, se seleccionó un conjunto de prueba (*Test*) y éste fue clasificado utilizando los conjuntos de entrenamiento construidos en cada una de las fases de los algoritmos, obteniendo el porcentaje de aciertos o de objetos bien etiquetados con la regla de clasificación NN.

En todos los experimentos realizados, se dividieron las bases de datos en varios bloques de manera aleatoria (mostramos aquí el experimento para 5 bloques) y se repitieron 10 veces los experimentos. Los resultados que se muestran en cada una de las figuras representan el promedio de los porcentajes obtenidos para cada una de las etapas con los diferentes conjuntos de muestras.

En las siguientes figuras, mostramos las curvas que se obtienen con los porcentajes de clasificación correcta en cada etapa. Además, agregamos una tercera curva a la que llamamos "Sin proceso" en la que el conjunto *Test* se clasifica con las uniones de los conjuntos G_i que van llegando con las etiquetas de clases originales de cada uno de estos objetos.

Evaluamos nuestras dos estrategias sobre 3 bases de datos sintéticas formadas por modos

gaussianos con diferentes grados de solapamiento. En la figura 1, a la izquierda se muestra la base de datos G3-I, formada por tres modos gaussianos muy solapados. En el centro, se observa la base de datos G3-II, la cual está formada también por tres modos gaussianos, dos de ellos con igual media y diferentes covarianzas. A la derecha, la base de datos G6 está formada por seis modos gaussianos, 4 de ellos muy solapados.

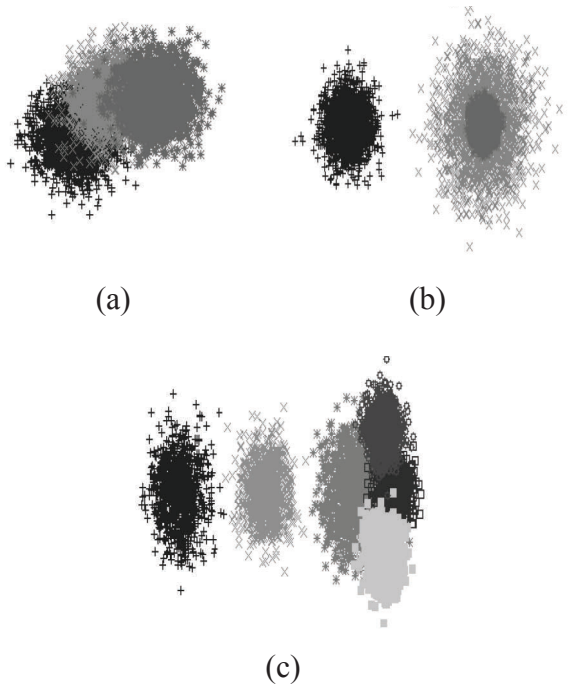


Figura 1 Bases de datos sintéticas. a) G3-I, b) G3-II, c) G6

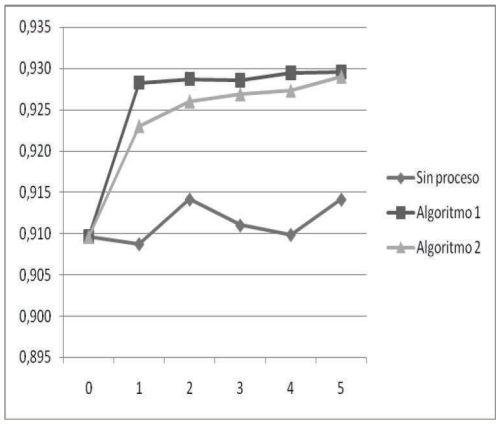
En la tabla 1, hacemos un resumen de las características de las bases de datos tomadas del UCI Repository para probar ambas estrategias.

En la figura 2, se muestran los resultados obtenidos con las bases de datos gaussianas. La característica fundamental de las mismas es el solapamiento que existe entre las clases, lo que origina que, a medida que se agregan puntos al conjunto inicial, no se observa siempre un ascenso de los porcentajes de clasificación correcta al utilizar la regla NN cuando tenemos en cuenta todos los objetos de la base de datos etiquetados correctamente, específicamente para la curva "Sin Proceso". Obsérvese que, para la base de datos G6, el comportamiento de la curva "Sin proceso" es siempre decreciente.

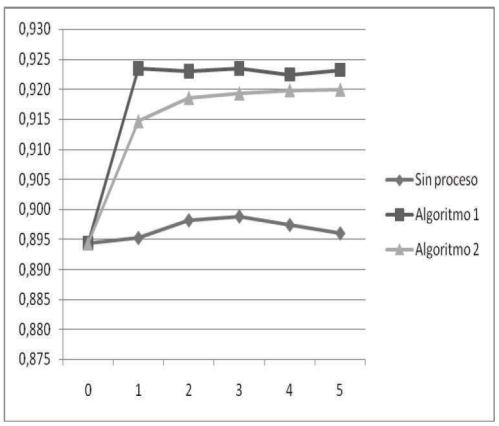
Sin embargo, cuando se aplican las dos estrategias propuestas, las curvas de ambos algoritmos están por encima de la curva "Sin proceso", lo que significa que al rechazar los objetos mal etiquetados se obtienen mejores porcentajes de clasificación correcta, de manera que los sucesivos conjuntos de entrenamiento que se van construyendo tienen mejor calidad que los conjuntos de entrenamiento obtenidos de etiquetar correctamente los datos. Además, se debe tener en cuenta que la cantidad de puntos necesarios para etiquetar el Test es menor, mientras que los porcentajes de acierto son mayores que cuando se utilizan todos los puntos de cada uno de los conjuntos G_i.

Tabla 1 Descripción de las bases de datos tomadas del UCI *Repository*

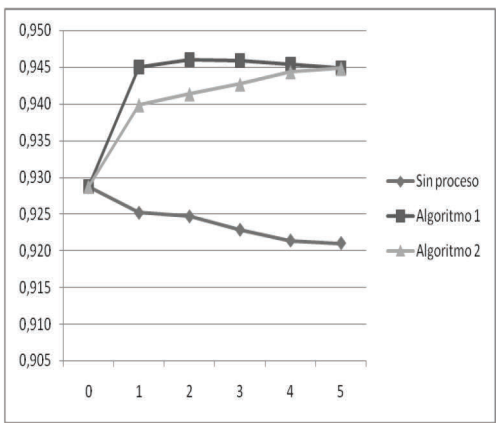
Bases de datos	*# de objetos*	*# de características*	*# de clases*
Australian	690	14	2
German	1000	24	2
Magic	19020	10	2
Cancer	699	9	2
Diabetes	768	8	2
Wave	5000	21	3



(a)



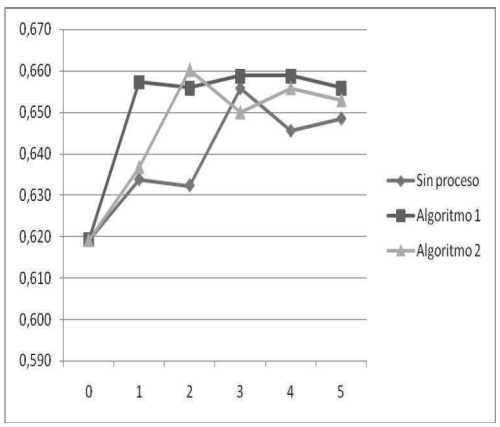
(b)



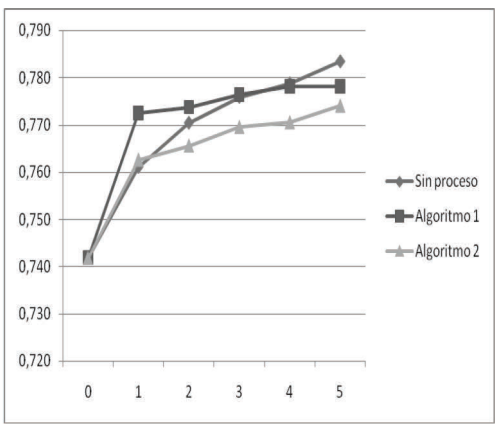
(c)

Figura 2 Resultados del aprendizaje sobre bases de datos sintéticas: a) G3-I, b) G3-II, c) G6

Sobre las bases de datos del UCI Repository, se puede observar en la figura 3 los resultados obtenidos con las bases de datos Australian y Magic. En el caso de Australian, con el algoritmo 1, los porcentajes de acierto son superiores que los obtenidos con todos los datos bien etiquetados, demostrando que la estrategia de eliminar ruido beneficia los conjuntos de entrenamiento que se van obteniendo. Con el algoritmo 2, sólo en una de las etapas eso no sucede. En general, se puede decir que con nuestra estrategia se alcanzan mejores conjuntos de entrenamiento, a pesar de que éstos tienen una menor cantidad de objetos, lo que constituye una mayor eficiencia en el proceso de cómputo.



(a)



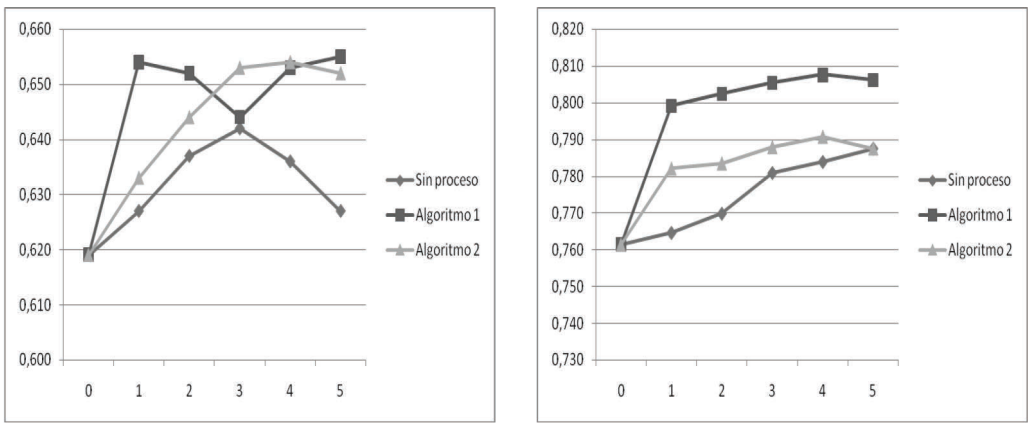
(b)

Figura 3 Resultados sobre las bases de datos a) Australian, b) Magic

Al observar los resultados sobre la base de datos Magic, cuando aplicamos el Algoritmo 1, se obtuvieron porcentajes de clasificación correcta superiores a los del algoritmo 2, y sólo en el último bloque, la curva "Sin proceso" está por encima. De todas formas, nótese el crecimiento de la curva con el algoritmo 2, lo que significa que hay un aprendizaje constante a pesar de ser datos originalmente no etiquetados y del rechazo de varios de los objetos en el proceso.

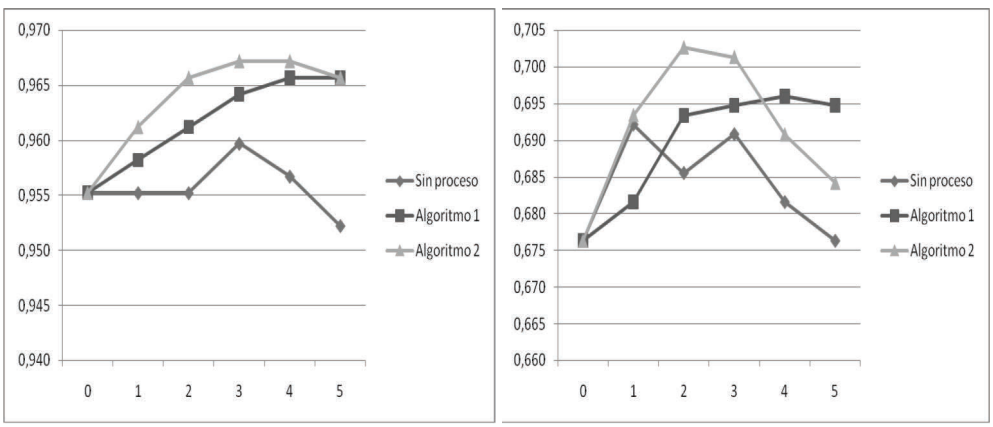
Sobre la base de datos German, obsérvese en la figura 4a que ambos algoritmos constituyen una mejor manera de construir el conjunto de entrenamiento que si agregamos al conjunto original todos los objetos de cada bloque bien etiquetados. Aunque para uno de los bloques se observa un descenso del porcentaje de aciertos con el algoritmo 1, el resultado es mejor que el de la curva "Sin proceso". El comportamiento de la curva del algoritmo 2 parece demostrar que es el mejor método para construir un nuevo conjunto de entrenamiento.

Sobre Wave (Figura 4b) con ambos algoritmos se obtuvieron porcentajes de clasificación superiores a los de la curva "Sin proceso". Nótese que la estrategia del algoritmo 1 constituye una mejor variante para esta base de datos.



(a) (b)

Figura 4 Resultados sobre las bases de datos a) German, b) Wave



(a) (b)

Figura 5 a) Cáncer, b) Diabetes

Los gráficos de la figura 5 muestran los resultados sobre las bases de datos Cancer y Diabetes. En el caso de Cancer, las dos curvas de aprendizaje demuestran la calidad del proceso, es decir, es mejor construir el conjunto de entrenamiento aplicando cualquiera de las estrategias que agregando todos los objetos posibles etiquetados correctamente. Obsérvese que, al agregar los dos últimos bloques completos de objetos correctamente etiquetados, se perjudica la clasificación al obtenerse una curva descendente.

En la figura 5b, se nota que los porcentajes de clasificación correcta que proporciona la primera estrategia van creciendo, lo que garantiza que para esta base de datos esa es la mejor técnica para formar el conjunto de entrenamiento. En el caso de la segunda variante aquí propuesta, después de un crecimiento, se observa un retroceso, pero ese mismo comportamiento se nota también con la curva "Sin proceso".

Conclusiones

En este trabajo, hemos presentado dos nuevas estrategias para construir un conjunto de entrenamiento que tenga en cuenta el *concept drift* cuando disponemos de un pequeño conjunto de datos etiquetados y una cantidad grande de datos sin etiquetar.

En el primer algoritmo, se emplea un esquema de detección de ruido no estático, ya que se tienen en cuenta en cada etapa los dos últimos resultados logrados con la limpieza de los datos, bajo la suposición de que con el transcurso del tiempo puede haber cambios en las distribuciones de probabilidad de los datos y también cambios de conceptos.

El segundo algoritmo utiliza un conjunto adicional de datos bien etiquetados dado a priori para detectar el ruido dinámicamente, mientras que el clasificador para etiquetar los datos no etiquetados se construye enriqueciéndolo en cada etapa del proceso con los datos aceptados como no ruidosos. De esta forma, estos conjuntos de entrenamiento tendrán en cuenta los posibles cambios que pueden ocasionar la incorporación de los nuevos objetos.

Los resultados experimentales indican que con ambas estrategias se logra obtener un buen conjunto de entrenamiento cuando únicamente se dispone de un conjunto pequeño de datos etiquetados. La exactitud y el rendimiento de ambos algoritmos en cuanto a los porcentajes de clasificación son superiores al porcentaje que se obtiene si en cada etapa se incorporaran todos los objetos con sus etiquetas verdaderas de clases, lo que además es más costoso.

Según los porcentajes alcanzados por cada uno de los conjuntos de entrenamiento construidos con ambos algoritmos, se observa que los relacionados con el primer algoritmo son generalmente superiores a los de la segunda variante, lo que garantiza mejores conjuntos de entrenamiento.

Los métodos han sido probados sobre varios conjuntos de datos, en particular con datos con un alto grado de solapamiento, demostrando que al rechazar varios puntos con la estrategia para detectar el ruido, los porcentajes de clasificación correcta que proporcionan son superiores.

Referencias

1. R. Bose, P. van der Aalst, I. Žliobaitė, M. Pechenizkiy. *Handling concept drift in process mining.* Proceedings of the 23rd International Conference on Advanced Information Systems Engineering. London, UK. 2011. pp. 391-405.

2. G. Widmer, M. Kubat. "Learning in the presence of concept drift and hidden contexts". *Machine Learning.* Vol. 23. 1996. pp. 69-101.

3. R. Elwell, R. Polikar, "Incremental learning of concept drift in nonstationary environments". *IEEE Transactions on Neural Networks*.Vol. 22. 2011. pp. 1517-1531.

4. G. Ross, N. Adams, D. Tasoulis, D. Hand. "Exponentially weighted moving average charts for detecting concept drift". *Pattern Recognition Letters*. Vol. 33. 2012. pp. 191-198.

5. O. Chapelle, A. Zien, B. Schölkopf, *Semi-supervised Learning*. 1st ed. Ed. MIT Press. Cambridge, MA, USA. 2006. pp. 3-5.

6. V. Castelli, T. Cover, "On the Exponential Value of Labelled Samples". *Pattern Recognition Letters*. Vol. 16. 1995. pp. 105-111.

7. V. Castelli, T. Cover. "The Relative Value of Labeled and Unlabeled Samples in Pattern Recognition With an Unknown Mixing Parameter". *IEEE Transactions on Information Theory*. Vol. 42. 1996. pp. 2101-2117.

8. J. Ratsaby, S. Venkatesh. *Learning From a Mixture of Labelled and Unlabelled Examples With Parametric Side Information.* Proceedings of the 8th Annual Conference on Computational Learning Theory. Santa Cruz, USA. 1995. pp. 412-417.

9. K. Nigam, R. Ghani. *Analyzing the Effective and Applicability of Co-training*. Proceedings of the 9th International Conference on Information and Knowledge Management. McLean, VA, USA. 2000. pp. 86-93.

10. F. Cozman, I. Cohen, M. Cirelo. *Semi-supervised Learning of Mixture Models*. Proceedings of the 20th International Conference on Machine Learning. Washington DC., USA, 2003. pp. 99-106.

11. D. Yarowsky. *Unsupervised Word Sense Disambiguation Rivaling Supervised Methods*. Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics. Cambridge. MA, USA. 1995. pp. 189-196.

12. E. Riloff, J. Wiebe, T. *Wilson Learning Subjective Nouns Using Extraction Pattern Bootstrapping.* Proceedings of the 7th Conference on Natural Language Learning. Edmonton, Canada. 2003. pp. 25-32.

13. B. Maeireizo, D. Litman, R. Hwa. *Co-training for Predicting Emotions With Spoken Dialogue Data.* Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics. Barcelona, Spain. 2004. pp. 203-206.

14. C. Rosenberg, M. Hebert, H. Schneiderman. *Semi-Supervised Self-training of Object Detection Models.* Proceedings of the 7th IEEE Workshop on Applications of Computer Vision. Breckenridge, USA. 2005. pp. 29-36.

15. Y. Jin, Y. Ma, L. Zhao. *A Modified Self-training Semi-supervised SVM Algorithm*. Proceedings of the International Conference on Communication Systems and Network Technologies. Gujarat, India. 2012. pp. 224-228.

16. A. Blum, T. Mitchell. *Combining Labelled and Unlabelled Data With Co-training*. Proceedings of the Workshop on Computational Learning Theory. New York, USA. 1998. pp. 92-100.

17. T. Mitchell. *The Role of Unlabeled Data in Supervised Learning*. Proceeding of the 6th International Colloquium on Cognitive Science. San Sebastian, Spain. 1999. pp. 1-8.

18. S. Goldman, Y. Zhou. *Enhancing Supervised Learning With Unlabelled Data*. Proceedings of the 17th International Conference on Machine Learning. Stanford, USA. 2000. pp 327-334.

19. Y. Zhou, S. Goldman. *Democratic Co-learning*. Proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligence. Boca Raton, FL, USA. 2004. pp. 594-602.

20. V. Vapnik. *Statistical Learning Theory*. 1st ed. Ed. Wiley. New York, USA. 1998. pp. 434-436.

21. A. Demirez, K. Bennett. "Optimization Approaches to Semi Supervised Learning". M. Ferris, O. Mangasarian, J. Pang (Eds.). *Applications and Algorithms of Complementarity*. 1st. ed. Ed. Kluwer Academic Publishers. Boston, USA. 2000. pp. 121-141.

22. Y. Shi, Y. Tian, G. Kou, Y. Peng, J. Li. "Unsupervised and Semi-supervised Support Vector Machines". *Optimization Based Data Mining: Theory and Applications*. 1st ed. Ed. Springer. London, UK. 2011. pp. 61-79.

23. B. Settles. *Active Learning Literature Survey*. Computer Sciences Technical Report 1648. University of Wisconsin-Madison. Wisconsin, USA. 2009. pp. 1-44.

24. F. Gu, D. Liu, X. Wang. *Semi-Supervised Weighted Distance Metric Learning for kNN Classification*. Proceedings of the International Conference on Computer, Mechatronics, Control and Electronic Engineering. Changchun, China. 2010. pp. 406-409.

25. B. Ni, S. Yan, A. Kassim. "Learning a Propagable Graph for Semisupervised Learning: Classification and regression". *IEEE Transactions on Knowledge and Data Engineering*. Vol. 24. 2012. pp. 114-126.

26. C. Kalish, T. Rogers, J. Lang, X. Zhu."Can Semi-Supervised Learning Explain Incorrect Beliefs About Categories?". *Cognition*. Vol. 120. 2011. pp. 106-118.