Mejora del desempeño de SVM usando un GA y la creación de puntos artificiales para conjuntos de datos no balanceados

José Hernández Santiago[1,2], Jair Cervantes Canales[2], Carlos Hiram Moreno Montiel[1], Beatriz Hernández Santiago[1]

[1] Tecnológico de Estudios Superiores de Chimalhuacán, Edo. de México, México

[2] Posgrado e Investigación Universidad Autónoma del Estado de México, Edo. de México, México

{jhernandezs, jcervantesc}@uaemex.mx, {josehernandez, carlosmoreno}@teschi.edu.mx

Resumen. En el mundo real los conjuntos de datos son regularmente no balanceados representando un problema crucial en Aprendizaje de Máquinas, ya que provoca una baja precisión en la mayoría de las técnicas de clasificación. Las SVM han reportado una excelente capacidad de generalización en los últimos años; sin embargo, al trabajar con conjuntos de datos no balanceados presentan un desempeño pobre debido a que el hiperplano obtenido queda sesgado hacia la clase mayoritaria. En este artículo, se presenta un nuevo algoritmo capaz de generar puntos artificiales dentro de la clase minoritaria y en la frontera entre clases a partir de los Vectores Soporte positivos, estos nuevos puntos son agregados al conjunto de entrenamiento a fin de disminuir el desbalance entre clases, mejorando el desempeño de las SVM en la mayoría de las pruebas.

Palabras clave: Máquinas de vectores soporte, conjuntos no balanceados, puntos artificiales, vectores soporte, sensitividad, especificidad, algoritmo genético.

Improved Performance of SVM Using a GA and the Creation of Artificial Points for Unbalanced Data Sets

Abstract. Real world data sets are regularly unbalanced, which is a crucial problem in Machine Learning, it causes a low accuracy in most classification techniques. SVM have reported excellent generalization capability in recent years; however, working with unbalanced data sets have poor performance due to the retrieved hyperplane is biased towards the majority class. This article presents a new algorithm capable of generating artificial points within the minority class and on the border between classes from the positive Support Vectors, these new points are added to the training data set in order to reduce the imbalance between classes, improving the performance of SVM in the majority of tests.

Keywords: Support vector machines, unbalanced data sets, artificial points, support vectors, sensitivity, specificity, genetic algorithm.

1. Introducción

Un conjunto de datos esta no balanceado cuando contiene un gran número de patrones de muestra de un tipo (clase mayoritaria) y un número muy reducido de patrones de muestra opuestos a los anteriores (clase minoritaria). En el mundo existen aplicaciones que presentan un desbalance muy acentuado en su conjunto de entrenamiento, por ejemplo en problemas de detección de fraudes, donde el radio de desbalance puede ir de 100 a 1 hasta 100,000 a 1 [1], otros ejemplos son la clasificación de secuencias de proteína [2, 3], diagnóstico médico [4, 5], detección de intrusos y clasificación de texto [6, 7].

Experimentos recientes [8, 9, 10] han mostrado que el desempeño de la mayoría de los métodos de clasificación son afectados cuando son aplicados sobre conjuntos no balanceados, siendo más evidente cuando el radio de desbalance es muy alto, debido a que los clasificadores en general están diseñados para reducir el error promedio global sin importar la distribución de las clases.

Las Máquinas de Vectores Soporte (SVM) son actualmente una de las técnicas de clasificación más importantes [11, 3, 12, 13] debido a que tiene un mejor desempeño frente a otros métodos como redes neuronales artificiales [14, 15], árboles de decisión y clasificadores Bayesianos [2, 16]. Una SVM busca maximizar el margen de separación entre los híperplanos de cada clase, otorgándole un gran poder de generalización, propiedad que puede ser explicada por la teoría de aprendizaje estadístico [17]; sin embargo, en el caso de conjuntos no balanceados, el hiperplano de separación se ve sesgado hacia la clase mayoritaria, provocando un impacto negativo en la precisión de clasificación puesto que la clase minoritaria puede ser considerada como ruido y por consiguiente ignorada por el clasificador.

El desarrollo de nuevas técnicas para reforzar el desempeño de clasificadores como las SVM sobre conjuntos no balanceados es importante en el área de reconocimiento de patrones, minería de datos y aprendizaje de máquinas. Para mejorarlas surgen técnicas como bajo muestreo (*undersampling*) que balancea el conjunto al reducir la clase mayoritaria. Por su parte la técnica de sobre muestreo (*oversampling*) duplica la clase minoritaria tantas veces como sea necesario hasta equilibrar el tamaño de las clases [8]. La desventaja de estas técnicas es que al eliminar datos de forma aleatoria para la clase mayoritaria, se podrían estar eliminando datos importantes sobre la frontera de decisión, causando que el hiperplano de separación no sea el óptimo al usar una SVM; por el contrario, si se duplican los datos de la clase minoritaria, el tiempo de entrenamiento se incrementaría debido a que la complejidad de la SVM es $O(n^2)$ [2], además de que no aportan nada al entrenamiento puesto que estos puntos no son diferentes a los ya existentes.

Chawla et al. [18] propuso *Synthetic Minority Over sampling Technique* (SMOTE) que genera puntos sintéticos que son incluidos en la clase minoritaria. SMOTE toma un punto de la clase minoritaria y produce una nueva versión de este al desplazarlo hacia su vecino más cercano una distancia aleatoria para cada dimensión. Esta técnica

no incluye una selección de datos, opera con todo el conjunto de entrada y de acuerdo a los resultados es mejor que *undersampling* y *oversampling*.

Una combinación de SMOTE y oversampling fue propuesta en [8], introduciendo un esquema de penalización del error dependiendo de la clase, decrementando el costo para la clase mayoritaria e incrementándolo para la clase minoritaria; logrando una mayor densidad en la distribución de la clase minoritaria y colocando el hiperplano de separación más cerca de la clase mayoritaria. En [19] diferentes criterios de penalización son usados para producir efectos similares en la separación del hiperplano. Otras propuestas inspiradas en SMOTE pueden encontrarse en [20, 21, 22, 23]. Otro enfoque se basa en aplicar undersampling sobre conjuntos no balanceados seleccionando las muestras por medio de un algoritmo genético y resultando mejor que un simple muestreo aleatorio [24].

Para el caso de SVM con conjuntos no balanceados, en [25] logran mejorar su desempeño al generan datos sintéticos a partir de los vectores soporte (SV); mientras que en [26] estos SV son desplazados un épsilon. Ambos algoritmos trabajan en el espacio de características; sin embargo, sus desplazamientos son aleatorios obteniendo precisiones sesgadas hacia la sensitividad o la especificidad.

En este artículo se presenta una nueva técnica de muestreo a fin de mejorar el desempeño de las SVM sobre conjuntos no balanceados. Inicialmente se entrena la SVM usando el conjunto de entrenamiento completo con el objetivo de obtener los Vectores Soporte (SV) que serán usados como base para crear nuevos puntos artificiales y poblar la clase minoritaria. Este método incluye un algoritmo genético para encontrar una buena combinación entre parámetros de la SVM y puntos artificiales creados dentro de la clase minoritaria así como en la frontera entre clases.

En la metodología se explica cómo el método propuesto crea puntos artificiales de forma dirigida partiendo de los vectores soporte, que son los puntos más importantes, en lugar de usar todo el conjunto de datos de entrada como lo harían otras técnicas reportadas en la literatura; permitiendo crear puntos tanto dentro de la clase minoritaria como en la frontera entre clases. Para evitar introducir ruido con las nuevas muestras, el algoritmo es capaz de distinguir entre los SV más cercanos a la clase minoritaria. En la sección de resultados se puede observar que el método de clasificación propuesto mantiene el equilibrio entre sensitividad y especificidad al mismo tiempo que las mejora mientras que otras técnicas presentan sesgo hacia alguna de las dos métricas mencionadas.

2. Preliminares

2.1. Máquinas de vectores soporte (SVM)

Las SVM fueron inspiradas en los resultados de la teoría de aprendizaje estadístico desarrollado por Vapnik en los 70's [17]. Este clasificador permite encontrar un hiperplano capaz de separar linealmente dos clases, proyectando el espacio de entrada original a un espacio de características altamente dimensional donde maximiza el margen entre clases.

Las SVM permiten estimar una función de clasificación óptima empleando datos de entrenamiento etiquetados como X_{tr}, de esta forma, la función f clasificará

correctamente datos no vistos antes por el clasificador (datos de prueba). Considerando el caso más simple de clasificación binaria, asumimos que el conjunto X_{tr} es dado como:

$$(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n), \tag{1}$$

i.e. $X_{tr} = \{x_i, y_i\}_{i=1}^{n}$ donde $x_i \in R^d$ y $y_i \in R(+1, -1)$ corresponde a la etiqueta de clasificación de la muestra x_i. La función de clasificación puede ser escrita como:

$$y_i = sign\left(\sum_{i=1}^{n} \alpha_i y_i K\langle x_i \cdot x_j\rangle + b\right), \tag{2}$$

donde $x = [x_1, x_2, \dots, x_n]$ son los datos de entrada. Un nuevo objeto *x* puede ser clasificado usando (2). El vector x_i es mostrado en la forma de producto punto. Las α'_i son multiplicadores de Lagrange y b es el bias obtenido al entrenar la SVM.

2.2. Métricas para evaluar precisión en conjuntos no balanceados

Comúnmente, la precisión (*accuracy*) es la medida empleada para evaluar empíricamente el desempeño de un clasificador; sin embargo, para clasificación con datos no balanceados, esta métrica puede llevarnos a conclusiones erróneas debido a que la clase minoritaria tiene un impacto muy pequeño en la precisión. En conjuntos de datos con una distribución muy sesgada, la métrica de la precisión total no es suficiente debido a que en un conjunto descompensado de 99 a 1, un clasificador que etiquete todos los datos de prueba como negativos obtendrá una precisión del 99%, siendo inútil como clasificador para detectar los ejemplos positivos inusuales. La comunidad médica y la comunidad de aprendizaje de máquinas emplean dos métricas, la sensitividad (3) y la especificidad (4) para evaluar el desempeño de un clasificador sobre grandes conjuntos de datos altamente no balanceados.

$$S_n^{false} = \frac{T_N}{T_N + F_P}, \tag{3}$$

S_n^{true} es la proporción de verdaderos positivos i.e.,

$$S_n^{true} = \frac{T_P}{T_P + F_N}, \tag{4}$$

donde T_P es el número de patrones de clase +1 reales pronosticados como positivos (verdaderos positivos), T_N es el número de patrones de clase -1 reales pronosticadas como negativos (verdaderos negativos), F_P es el número de patrones de clase -1 reales pronosticados como positivos (falsos positivos) y F_N es el número de patrones de clase +1 reales que son pronosticados como negativos (falsos negativos).

2.3. Receiver operating characteristic (ROC)

La gráfica proporcionada por el método *Receiver Operating Characteristic* (ROC) es ampliamente usado para analizar el desempeño de clasificadores binarios y al medir

el área bajo la curva ROC (AUC) se obtiene una representación numérica de que tan separables son las clases analizadas [27]. Las ventajas más importantes del análisis con ROC es que no es necesario especificar los costos por errores de clasificación y proporciona una forma visual para analizar el desempeño del clasificador.

3. Metodología

3.1. Creación de puntos artificiales

El primer paso del método propuesto consiste en identificar las clases minoritaria y mayoritaria del conjunto de entrenamiento a partir del conjunto no balanceado original. La clase minoritaria contiene *t* muestras positivas etiquetadas como X_t^+, mientras que las muestras negativas X_t^- pertenecen a la clase mayoritaria. Si el conjunto negativo es muy grande, se aplica una técnica de bajo muestreo para evitar un alto costo computacional. El nuevo conjunto formado por X_t^+ y X_t^-posteriormente es empleado para entrenar una SVM, obteniendo un hiperplano preliminar $H_1(X_t^+, X_t^-)$ y su vectores soporte (SV).

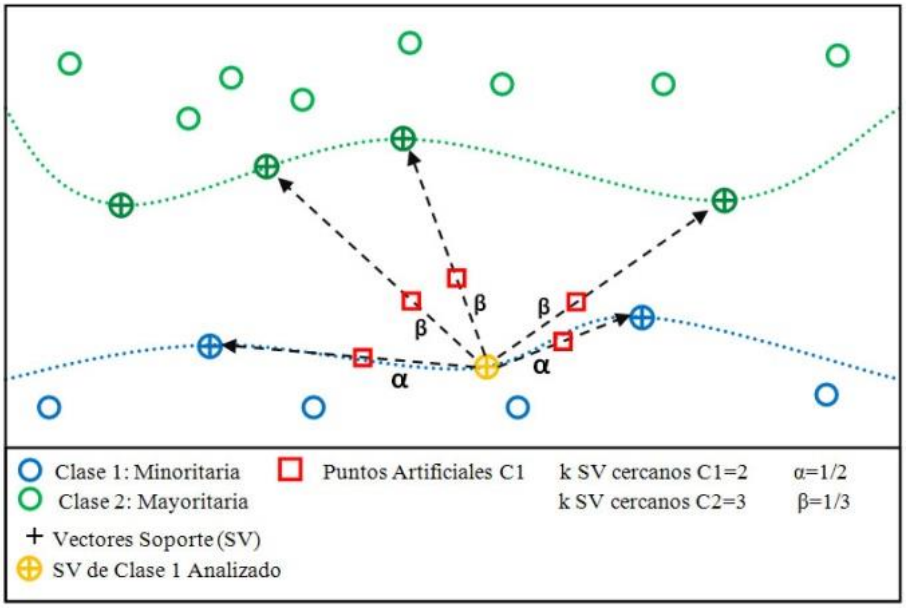


Fig. 1. Creación de puntos artificiales por el método propuesto

El segundo paso implica etiquetar los *SV* de acuerdo a la clase a la que pertenezcan, obteniendo SV^+ y SV^-; después son utilizados los SV^+ como referencia para crear nuevos *puntos artificiales* como se muestra en la Fig. 1, primero dentro de la clase minoritaria y después en su frontera con la clase mayoritaria. Para poblar la clase minoritaria es necesario elegir el número *k* de puntos que se crearán por cada SV^+ y el

desplazamiento α que moverá el SV_t^+ a una nueva posición. También hay que encontrar los $k\ SV^+$ más cercanos por cada SV^+ para finalmente aplicar (5).

$$X_{tk}'^{+} = SV_t^+ + \alpha * |SV_t^+ - SV_k^+| \quad \text{para cada } k\ SV_t^+ \text{ más cercano.} \tag{5}$$

Después de una forma similar, se poblará la frontera con puntos artificiales positivos pero ahora utilizando los $k\ SV^-$ más cercanos para cada SV^+; la proporción de desplazamiento β desplazará el SV_t^+ original en dirección de su vector soporte negativo más cercano de la siguiente forma:

$$X_{tk}''^{+} = SV_t^+ + \beta * |SV_t^+ - SV_k^-| \quad \text{para cada } k\ SV_t^- \text{ más cercano.} \tag{6}$$

La métrica usada para evaluar la distancia es la euclidiana y tanto α como β están en rangos entre 0 y 0.9, de esta forma se evita que un nuevo punto artificial pueda quedar localizado en el mismo lugar que el SV_k, ya que introduciría ruido.

3.2. Mejora de los parámetros usando un algoritmo genético

Con el fin de mejorar los parámetros de la SVM tales como tipo de *kernel*, costo *C*, *gamma* para el *kernel RBF*, así como los parámetros del método propuesto, $k\ SV^+$ más cercanos, $k\ SV^-$ más cercanos y los valores normalizados entre 0 y 0.9 para el desplazamiento α y β; se utilizó un algoritmo genético (GA) cuyo cromosoma contenía las variables anteriormente citadas. En la Tabla 1 se presenta el tamaño en bits ocupado por cada variable.

Tabla 1. Variables para el cromosoma

Variable	Rango	Precisión decimal	Tamaño en bits
gamma	[0.001, 1.000]	3	10
C	[0.001, 1.000]	3	10
Tipo de kernel	{1-lineal, 2-RBF}	0	1
k SV^+	{0,1,2}	0	2
α	[0.01, 0.90]	2	7
k SV^-	{0,1}	0	1
β	[0.01, 0.90]	2	7

El algoritmo genético encuentra una solución en un tiempo razonable y gracias a sus operadores de cruza y mutación puede realizar una búsqueda explotativa y explorativa respectivamente por lo que es mejor que utilizar una búsqueda por malla. Cada individuo dentro de la población tendrá un cromosoma como se ejemplifica en la Fig. 2 y su calidad como solución al problema estará en función de que tan próximas están las precisiones AUC, S_n^{true} y S_n^{false} respecto a la solución óptima que ocurre cuando todas valen 1.0. Para obtener los valores de cada métrica se requiere decodificar el cromosoma y obtener el fenotipo, asignando un valor a cada parámetro. Una vez que se tienen todos los valores se crean nuevos *puntos artificiales* y se entrena una SVM con los parámetros obtenidos.

El tamaño de cada gen se determinó con la fórmula (7), siendo *n* la precisión deseada.

$$nbits = log2[(limiteSuperior - limiteInferior)x10^n] + 0.5. \quad (7)$$

Para aplicar el mapeo de números binarios a reales se usó la fórmula (8), donde *nbits* es el tamaño de la palabra calculada anteriormente, *x'* es el valor obtenido de la conversión de la cadena binaria a decimal y *x* es el valor real obtenido de la transformación de *x'*.

$$x = limiteInferior + \frac{x'[limiteSuperior - limiteInferior]}{2^{nbits} - 1}. \quad (8)$$

3.3. Selección del modelo

El entrenamiento con SVM involucra el ajuste de varios parámetros que tienen un crucial efecto sobre el desempeño del clasificador entrenado. El modelo propuesto emplea una función de base radial (RBF) para entrenar la SVM, definida como:

$$K(x_i - x_j) = exp\left(-\gamma\|x_i - x_j\|^2\right), \gamma > 0. \quad (9)$$

El parámetro *C* regula el punto medio entre error de entrenamiento y complejidad, mientras que γ es un parámetro del *kernel*. La obtención de buenos parámetros se logró usando un algoritmo genético, fijando el tamaño del cromosoma en 38 bits, con una población de 24 individuos y seleccionando los padres en cada generación por sobrante estocástico con reemplazo. Los operadores usados son cruza de dos puntos y mutación uniforme con una probabilidad fija de 0.25, eligiendo el mejor individuo de dos corridas.

Para asegurar la convergencia, se aplicó el enfoque elitista, manteniendo intacto el material genético del mejor individuo en la siguiente generación y adicionalmente se utilizó una codificación Gray para disminuir las debilidades de la cruza de dos puntos.

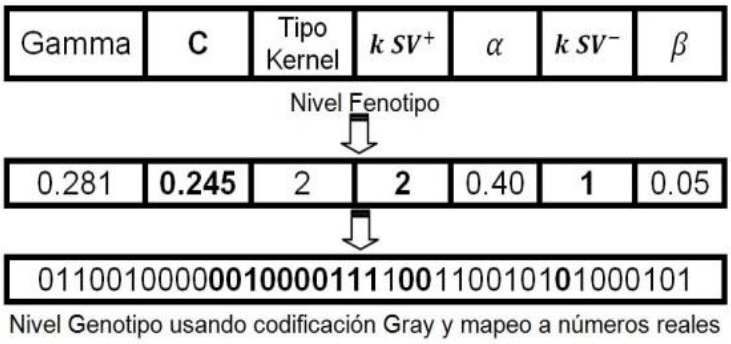
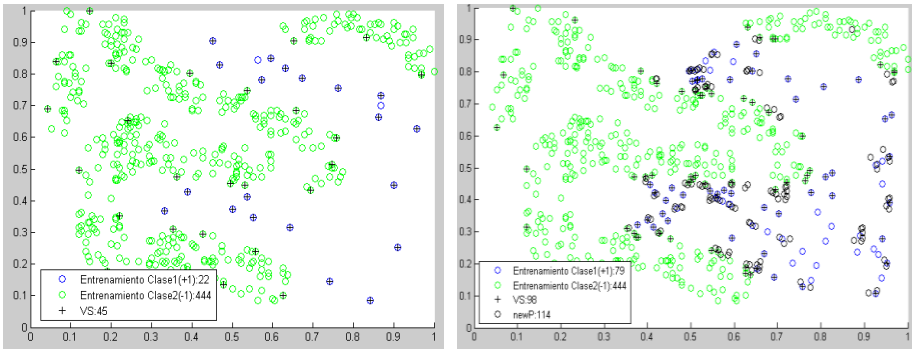


Fig. 2. Ejemplo de la estructura del cromosoma para el conjunto four class



a) Conjunto de entrenamiento no balanceado.

b) Conjunto de entrenamiento balanceado con puntos artificiales por el método propuesto.

Fig. 3. Distribución del conjunto four class, clase 1 en verde y clase 2 en azul

4. Resultados

En las pruebas realizadas se empleó el conjunto de datos KEEL Dataset, que es un conjunto comúnmente empleado para evaluar el desempeño de clasificadores sobre conjuntos de datos no balanceados; se encuentra disponible en http://sci2s.ugr.es/keel/datasets.php y los radios de desbalance van desde 1:1.4 para el conjunto *liver disorders* hasta 1:41.4 para *yeast 6*, que puede interpretarse como un patrón de muestra positivo por cada 41.4 patrones de muestra negativos correspondiente a este último conjunto.

Para realizar los experimentos se prepararon los conjuntos de entrenamiento y prueba, seleccionándolos aleatoriamente y destinando 80% y 20% respectivamente a partir del conjunto original. A continuación se presentan los resultados en dos secciones, primero usando las técnicas clásicas y después aplicando el método propuesto.

4.1. Desempeño de la SVM usando técnicas clásicas

En la Tabla 2 se presentan las precisiones AUC, S_n^{true} y S_n^{false}, donde cada valor corresponde al promedio de 10 pruebas para cada método, incluyendo la precisión alcanzada con el conjunto original. Puede notarse que las precisiones están sesgadas hacia S_n^{false} en el conjunto original, mientras que al aplicar *undersampling* u *oversampling* la precisión promedio no supera el 95% para las cuatro métricas y cuando alguna lo hace, la técnica sesga la precisión hacia una de ellas. En SMOTE, se usaron como parámetros un valor de 400 para *N* y 10 *k vecinos*. Esta técnica también tiene la debilidad de sesgar su precisión, clasificando la mayoría de patrones de prueba como positivos cuando está sesgada a S_n^{true} o clasificando la mayoría como negativos cuando está sesgada a S_n^{false}.

También se puede concluir que aunque el área bajo la curva ROC sea un valor alto para las técnicas analizadas, no puede usarse para discriminar si el clasificador es bueno detectando patrones positivos o si la técnica para balancear el conjunto es buena [27].

Tabla 2. Precisión para los conjuntos de datos no balanceados

Conjunto de datos	No balanceado			Undersampling			Oversampling			SMOTE		
	AUC	S_n^T	S_n^F	AUC	S_n^T	S_n^F	AUC	S_n^T	S_n^F	AUC	S_n^T	S_n^F
liver disorders	0.75	0.48	0.85	0.74	0.68	0.69	0.75	0.64	0.75	0.71	0.89	0.28
four class	0.87	0.51	0.97	0.87	0.78	0.78	0.88	0.81	0.80	0.83	0.91	0.71
glass 1	0.79	0.08	0.99	0.77	0.84	0.46	0.79	0.80	0.57	0.75	0.91	0.31
diabetes	0.81	0.56	0.86	0.81	0.74	0.71	0.81	0.69	0.76	0.79	0.83	0.62
glass 0	0.85	0.31	0.92	0.83	1.00	0.44	0.83	0.99	0.47	0.81	0.99	0.45
vehicle 2	0.99	0.90	0.98	0.99	0.96	0.91	0.99	0.82	0.98	0.99	0.97	0.93
vehicle 3	0.80	0.13	0.97	0.79	0.76	0.67	0.81	0.57	0.82	0.82	0.88	0.66
ecoli 1	0.95	0.69	0.96	0.94	0.92	0.84	0.94	0.89	0.86	0.95	0.91	0.84
ecoli 2	0.96	0.81	0.98	0.96	0.93	0.91	0.96	0.92	0.94	0.96	0.93	0.94
glass 6	0.93	0.70	0.98	0.96	0.80	0.95	0.94	0.80	0.98	0.96	0.80	0.98
yeast 3	0.98	0.66	0.98	0.98	0.91	0.93	0.97	0.65	0.98	0.98	0.86	0.96
ecoli 3	0.92	0.44	0.98	0.95	0.93	0.83	0.94	0.89	0.88	0.94	0.83	0.92
glass 2	0.66	0.00	1.00	0.64	0.97	0.31	0.64	0.87	0.37	0.64	0.00	1.00
cleveland 0 vs 4	0.98	0.15	1.00	0.94	0.95	0.70	0.97	0.20	0.99	0.98	0.60	0.99
glass 4	0.97	0.05	1.00	0.93	0.95	0.80	0.97	0.95	0.94	0.97	0.95	0.95
ecoli 4	1.00	0.75	1.00	1.00	1.00	0.93	0.99	0.90	0.98	1.00	0.93	0.99
page blocks 1-3 vs 4	1.00	0.50	1.00	0.99	0.98	0.90	1.00	0.90	0.98	1.00	0.94	1.00
glass 5	0.97	0.00	1.00	0.92	0.90	0.83	0.97	0.80	0.93	0.97	0.70	0.99
yeast 4	0.81	0.00	1.00	0.84	0.75	0.87	0.87	0.71	0.88	0.86	0.54	0.97
yeast 5	0.99	0.16	1.00	0.99	1.00	0.91	0.99	1.00	0.94	0.99	0.93	0.97
yeast 6	0.90	0.00	1.00	0.92	0.86	0.88	0.94	0.81	0.92	0.94	0.70	0.98

4.2. Desempeño de la SVM usando el método propuesto

En la Tabla 3 se presentan las precisiones AUC, S_n^{true} y S_n^{false} para el método propuesto con su respectiva desviación estándar, siendo cada valor el promedio de 10 pruebas.

Para balancear el conjunto de entrenamiento, en cada prueba se anexaron los puntos artificiales creados por el método propuesto y después para la prueba los parámetros *kernel RBF*, *C* y *gamma* requeridos por la SVM fueron calculados usando un algoritmo genético con codificación Gray.

Se observa que las precisiones mejoraron notablemente frente al conjunto original y no presentan gran sesgo, salvo para el conjunto *glass 0*, *glass 1* y *vehicle 3*; sin

embargo en comparación, estas precisiones son mejores frente a las otras técnicas analizadas.

Tabla 3. Precisión para los conjuntos de datos no balanceados usando el método propuesto

Conjunto de datos	Método propuesto (promedio)			Método propuesto (desviación estándar)			Puntos artificiales (clase positiva)
	AUC	S_n^T	S_n^F	AUC	S_n^T	S_n^F	
liver disorders	**0.93**	**0.89**	**0.81**	0.03	0.03	0.07	1326
four class	**1.00**	**1.00**	**1.00**	0.00	0.00	0.00	624
glass 1	**0.89**	**0.87**	**0.77**	0.04	0.05	0.04	54
diabetes	**0.87**	**0.85**	**0.80**	0.04	0.05	0.02	338
glass 0	**0.87**	**0.96**	**0.64**	0.04	0.05	0.05	65
vehicle 2	**1.00**	**1.00**	**0.98**	0.00	0.00	0.01	500
vehicle 3	**0.93**	**0.93**	**0.85**	0.02	0.02	0.03	668
ecoli 1	**0.96**	**0.95**	**0.89**	0.04	0.05	0.05	111
ecoli 2	**0.97**	**0.96**	**0.96**	0.04	0.05	0.03	48
glass 6	**0.98**	**0.90**	**0.99**	0.04	0.11	0.03	38
yeast 3	**0.98**	**0.98**	**0.96**	0.01	0.02	0.01	642
ecoli 3	**0.97**	**0.94**	**0.93**	0.02	0.07	0.03	135
glass 2	**0.99**	**1.00**	**0.97**	0.03	0.00	0.06	84
cleveland 0 vs 4	**1.00**	**1.00**	**1.00**	0.00	0.00	0.00	90
glass 4	**1.00**	**1.00**	**0.99**	0.00	0.00	0.01	88
ecoli 4	**1.00**	**1.00**	**0.99**	0.00	0.00	0.01	108
page blocks 1-3 vs 4	**1.00**	**1.00**	**1.00**	0.00	0.00	0.00	108
glass 5	**1.00**	**1.00**	**1.00**	0.00	0.00	0.00	24
yeast 4	**0.98**	**0.99**	**0.95**	0.02	0.03	0.04	62
yeast 5	**1.00**	**1.00**	**0.99**	0.00	0.00	0.01	238
yeast 6	**0.98**	**0.91**	**0.99**	0.04	0.07	0.01	84

Por último, la desviación estándar no superó el 0.04 para el AUC, el 0.07 para S_n^{true}, salvo *glass 6*, mientras que S_n^{false} tuvo un máximo de 0.07 por lo que sugiere que es un método estable.

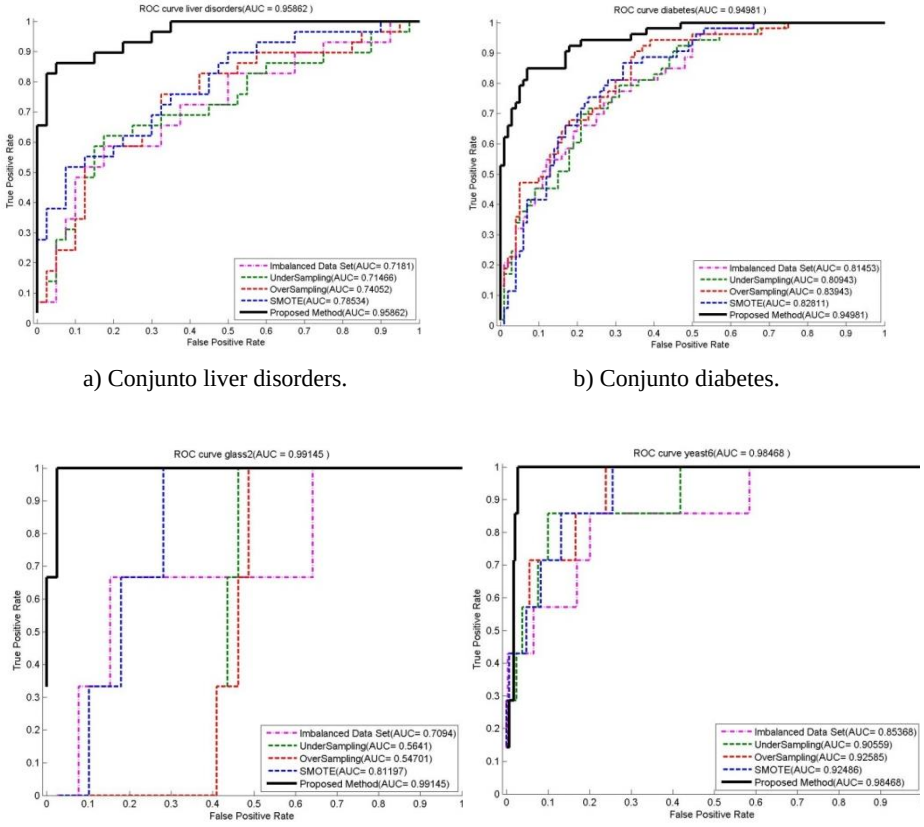
5. Conclusión

Las Máquinas de Vectores Soporte son una herramienta de clasificación que posee un buen desempeño sobre conjuntos balanceados; sin embargo, al trabajar en conjuntos desbalanceados, su desempeño es severamente afectado, ya que por la naturaleza de su entrenamiento el hiperplano obtenido se ve sesgado hacia la clase mayoritaria.

En este artículo, se presentó un nuevo método que mejora el desempeño de las SVM sobre conjuntos no balanceados, reduciendo el efecto del radio de desbalance al crear nuevos puntos artificiales que son agregados al conjunto de entrenamiento, al mismo

tiempo mejora significativamente el desempeño de las SVM en conjuntos con desbalance.

El método propuesto es diferente a otros métodos reportados en la literatura, ya que la creación de puntos es inteligente en el sentido de no utilizar como base todo el conjunto de datos de entrada, sino solo los Vectores Soporte, ya que son los puntos más importantes; sus dos fases pueden crear puntos tanto dentro de la clase minoritaria como en la frontera y al distinguir entre los SV más cercanos se puede evitar introducir ruido con los nuevos puntos.



a) Conjunto liver disorders. b) Conjunto diabetes.

c) Conjunto glass 2. d) Conjunto yeast 6.

Fig. 4. Gráficas ROC para una de las diez pruebas realizadas a los conjuntos a) liver disorders, b)diabetes, c) glass 2 y d) yeast 6

De acuerdo con los resultados, el método propuesto presentó una mejora en el desempeño de la SVM con precisiones superiores a las de las otras técnicas analizadas, disminuyendo el sesgo del hiperplano de separación al proveer el entrenamiento con más ejemplos de muestra positivos para la clase minoritaria y obteniendo resultados más notables cuando es aplicado sobre conjuntos de datos cuyo radio de desbalance es mayor a 10.

El método propuesto es estable, ya que su desviación estándar sobre la precisión no supera el 0.07, obteniendo una precisión S_n^{true} mayor o igual a 93% en 15 de los 22 conjuntos de datos no balanceados analizados y manteniendo una precisión alta para todas las precisiones, a diferencia de las demás técnicas, que presentan claramente una precisión sesgada hacia una de las métricas.

El desempeño en los conjuntos de datos probados es bueno; sin embargo, determinar la mejor combinación de parámetros, tanto para la SVM como para la creación de puntos artificiales es una tarea costosa computacionalmente, por lo que se incluyó un algoritmo genético dentro del algoritmo propuesto para obtener una buena combinación en un tiempo aceptable y mantener una buena precisión.

Referencias

1. Provost, F., Fawcett, T.: Robust Classification for Imprecise Environments. Machine Learning, pp. 203–231(2001)
2. Cervantes, J., Xiaoou, L., Wen, Y.: Splice site detection in DNA sequences using a fast classification algorithm. In: Proceedings of IEEE International Conference on System, Man and Cybernetics, pp. 2762–2767 (2009)
3. Dror, G., Sorek, R., Shamir, R.: Accurate identification of alternatively spliced exons using support vector machine. Bioinformatics, Vol. 21, No. 7, pp. 897–901 (2005)
4. Kononenko, I.: Machine learning for medical diagnosis: history, state of the art and perspective. Artificial Intelligence in Medicine, Vol. 23, pp. 89–109 (2001)
5. Grzymala-Busse, J.W., Stefanowski, J., Wilk, S.: A comparison of two approaches to data mining from imbalanced data. Journal of Intelligent Manufacturing, Vol. 16, pp.565–573 (2005)
6. Sebastiani, F.: Machine learning in automated text categorization. ACM Computing Surveys, Vol. 34, pp.1–47 (2002)
7. Tan, S.: Neighbor-weighted k-nearest neighbor for unbalanced text corpus. Expert Systems with Applications, Vol. 28, pp. 667–671 (2005)
8. Akbani, R., Kwek, S., Japkowicz, N.: Applying Support Vector Machines to Imbalanced Datasets. Boulicaut, J.-F., Esposito, F., Giannotti, F., Pedreschi, D. (Eds.): Machine Learning (ECML). Springer-Verlag Berlin Heidelberg, pp. 39–50 (2004)
9. Zeng, Z.Q., Gao, J.: Improving SVM Classification with Imbalance Data Set. In: Leung, C.S., Lee, M., Chan, J.H. (Eds.): Proceedings of the 16th International Conference on Neural Information Processing. Springer-Verlag, pp. 389–398 (2009)
10. Tezel, S.K., Latecki, L.J.: Improving SVM Classification on Imbalanced Data Sets in Distance Spaces. In: Ninth IEEE International Conference on Data Mining, pp. 259–267 (2009)
11. Bazzani, A., Bevilacqua, A., Bollini, D., Brancaccio, R., Campanini, R., Lanconelli, N., Riccardi, A., Romani, D., Zamboni, G.: Automatic detection of clustered microcalcifications in digital mammograms using an SVM classifier. In: Proceedings of European Symposium on Artificial Neural Networks, pp. 195–200 (2000)
12. Kong, W., Tham, L., Wong, K.Y., Tan, P.: Support Vector Machine Approach for Cancer Detection Using Amplified Fragment Length Polymorphism (AFLP) Screening Method. In: Proceedings of 2nd Asia-Pacific Bioinformatics Conference, Conferences in Research and Practice in Information Technology, pp. 63–66 (2004)
13. Platt, J.C.: Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines. Technical Report MSR-TR-98-14 (1998)
14. Arbach, L., Reinhardt, J.M., Bennett, D.L., Fallouh, G.: Mammographic Masses Classification: Comparison between Backpropagation Neural Network (BNN), K Nearest

Neighbors (KNN) and Human Readers. Electrical and Computer Engineering, Vol. 3, pp. 1441–1444 (2003)
15. Makal, S., Ozyilmaz, L., Palavaroglu, S.: Neural Network Based Determination of Splice Junctions by ROC Analysis. World Academy of Science, Engineering and Technology, No. 43, pp. 613–615 (2008)
16. Ya, Z., Chao-Hsien, Ch., Yixin, Ch., Hongyuan, Z., Xiang, J.: Splice site prediction using support vector machines with a Bayes kernel. Expert Systems with Applications, Vol. 30, No. 1, pp. 73–81 (2006)
17. Vapnik, V.N.: The Nature of Statistical Learning Theory. Springer-Verlag (1995)
18. Chawla, N., Bowyer, K., Hall, L., Kegelmeyer, W.: SMOTE: Synthetic Minority Over-sampling Technique. Journal of Artificial Intelligence Research, pp. 321–357 (2002)
19. Veropoulos, K., Campbell, C., Cristianini, N.: Controlling the Sensitivity of Support Vector Machines. In: Proceedings of the International Joint Conference on AI, pp. 55–60 (1999)
20. Hart, P.E.: The condensed nearest neighbor rule. IEEE Transactions on Information Theory, Vol. 14, pp. 515–516 (1968)
21. Han, H., Wang, W.-Y. , Mao, B.-H.: Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning. In: Huang, D.S., Zhang, X.-P., Huang, G.-B. (Eds.): Proceedings of the 1thInternational Conference on Intelligent Computing, Springer-Verlag, pp. 878–887 (2005)
22. Hu, S., Liang, Y., Ma, L., He, Y.: MSMOTE: Improving Classification Performance When Training Data is Imbalanced. In: Proceedings of the 2nd International Workshop on Computer Science and Engineering, Vol. 2, pp.13–17 (2009)
23. Hongyu, G., Herna, L.V.: Learning from imbalanced data sets with boosting and data generation: the DataBoost-IM approach. ACM SIG KDD Explorations Newsletter, Vol. 6, pp. 30–39 (2004)
24. Zou, S., Huang, Y., Wang, Y., Wang, J., Zhou, Ch.: SVM Learning from Imbalanced Data by GA Sampling for Protein Domain Prediction. In: Proceedings of The 9th International Conference for Young Computer Scientist, pp. 982–987 (2008)
25. Hernández, J., Cervantes, J., Trueba, A.: Mejorando la Clasificación de Datos No-Balanceados con SVM Generando Datos Sintéticos. Ciencia y Tecnología en Computación e Informática. CONACI, pp. 121–130 (2011)
26. Hernández, J., Cervantes, J., López, A., García, F.: Enhancing the Performance of SVM on Skewed Data Sets by Exciting Support Vectors. Pavón, J., Duque, N.D., Fuentes, R. (Eds.): Advances in Artificial Intelligence. Springer Berlin Heidelberg, pp. 101–110 (2012)
27. Fawcett, T.: An introduction to ROC analysis. Pattern Recognition Letters, Vol. 27, pp. 861–874 (2006)