DESARROLLO DE ALGORITMOS PARA MEJORAR EL DESEMPEÑO DE SVM EN CONJUNTOS DE DATOS NO BALANCEADOS

T E S I S

Que para obtener el grado de:

Maestro en Ciencias de la Computación

Presenta:

José Hernández Santiago

Tutor Académico:

Dr. Jair Cervantes Canales

Texcoco, Estado de México, México, (2012)

2012

Universidad Autónoma del Estado de México
UAEM

CENTRO UNIVERSITARIO XICO VALLE DE CHALCO

COPIA

DICTÁMEN DE AUTORIZACIÓN Y OBTENCIÓN DE GRADO DE MAESTRÍA

Texcoco, Méx. , a 9 de 11 de 2012

CENTRO UNIVERSITARIO UAEM TEXCOCO

Título del proyecto:

DESARROLLO DE ALGORITMOS PARA MEJORAR EL DESEMPEÑO DE SVM EN CONJUNTOS DE DATOS NO BALANCEADOS

Tesista:

Hernández Santiago José

CENTRO UNIVERSITARIO UAEM VALLE DE MÉXICO

Dictamen: **No. de revisión:** 3

- ☐ **Rechazado**
- ☐ **Sujeto a modificaciones**
- ☐ **Aceptado, condicionado**
- ☒ **Aceptado**

Observaciones generales:

Aceptado para la impresión
Aceptado para la defensa de grado

TITULACION RECIBIDO
Por
Texcoco, Méx., a 9 de 11 de 12

Tutor Adjunto	**Tutor Académico**	**Tutor Adjunto**
M. en C. José Sergio Ruiz Castilla	**Dr. Jair Cervantes Canales**	**Dr. Adrian Trueba Espinosa**

Agradecimientos

Agradezco a mi Tutor académico y mis revisores de tesis, quienes me guiaron en esta investigación

Agradezco a la UAEM por la beca proporcionada para realizar mis estudios de posgrado y a COMECyT por la beca de Titulación.

Dedicatoria

Quiero dedicar esta investigación a mi mamá Lucy y mis hermanas Beátriz y Alva, quienes son la razón para mejorar cada día.

Índice general

Índice de figuras

Índice de Tablas

Resumen

En los últimos años ha crecido el interés en el aprendizaje de máquinas para abordar problemas donde el discernimiento humano se torna difícil, sobre todo en tareas de clasificación. En la literatura se han reportado diversas técnicas para resolver tareas de clasificación, Clasificadores de Vecinos Cercanos, Clasificadores Bayesianos, Redes Neuronales Artificiales, Árboles de Decisión y recientemente las Máquinas de Vectores Soporte (SVM). Estas técnicas tienen un buen desempeño sobre problemas específicos pero son las SVM la técnica más utilizada recientemente debido a su buena capacidad de generalización, sin embargo, al igual que los clasificadores clásicos, fueron diseñadas para conjuntos con tamaños similares entre clases y cuando se abordan problemas donde el tamaño de una clase es muy pequeño (que contiene los ejemplos positivos) con respecto a otra (que contiene los ejemplos negativos), las SVM ven disminuido su desempeño, ya que aprenderá muy bien los ejemplos negativos pero su precisión será muy pobre cuando se requiera clasificar ejemplos positivos, que regularmente son los más importantes. En el mundo real los problemas de clasificación a menudo presentan esta diferencia entre la cantidad de muestras existentes para cada clase, haciendo más difícil la tarea de aprendizaje para los algoritmos de inteligencia artificial. Algunos ejemplos sobre conjuntos no balanceados son la detección de fraudes, donde puede haber un fraude entre mil transacciones correctas, clasificación de secuencias de proteína, clasificación de texto y diagnóstico médico, por mencionar algunos.

Con el fin de mejorar la precisión de los clasificadores, se han reportado en la literatura varias técnicas para homogeneizar el balance entre clases directamente sobre el conjunto de entrada. Técnicas como Bajo-muestreo y Sobre-muestreo, que reducen la clase mayoritaria e incrementan la clase minoritaria respectivamente,

otras técnicas como SMOTE, crean nuevos puntos para poblar la clase minoritaria, sin embargo, estas técnicas trabajan sobre el espacio de entrada y no distinguen entre los puntos importantes (identificados como Vectores Soporte por las SVM).

Es claro que el desarrollo de técnicas que balanceen las clases y mejoren su desempeño en conjuntos de datos no balanceados es crucial hoy en día. En esta Tesis, se proponen dos nuevos algoritmos para mejorar el desempeño de las SVM sobre conjuntos de datos no balanceados. El primer algoritmo excita los Vectores Soporte (SV), generando nuevos SV positivos que estarán desplazados un épsilon respecto a los SV positivos originales. El objetivo de este primer método es reducir el sesgo que el hiperplano tiene hacia la clase mayoritaria. El segundo algoritmo genera puntos artificiales primero dentro de la clase minoritaria y después en la frontera entre clases a partir de los SV positivos, estos nuevos puntos son agregados al conjunto de entrenamiento con el fin de disminuir el desbalance entre clases.

La obtención de buenos parámetros es una de las etapas más importantes para mejorar la precisión de las SVM, por esto, el primer algoritmo propuesto incluye una búsqueda en malla ya que son pocos los parámetros que se requieren optimizar, mientras que el segundo algoritmo, necesita optimizar siete parámetros y fue necesario implementar un Algoritmo Genético. Los resultados experimentales obtenidos sobre diversos conjuntos no balanceados muestran que los dos algoritmos mejoran el desempeño de las SVM en la mayoría de las pruebas frente a técnicas clásicas.

Capítulo 1

Introducción

1.1. Estado del Arte

La tarea de clasificación es un problema ampliamente estudiado en la literatura. Los problemas donde se requiere decidir si nuevas muestras pertenecen a una clase o no, pueden verse como problemas de clasificación de conjuntos binarios, donde sólo existen dos clases. Regularmente estos datos son etiquetados como "+1" las muestras positivas o que pertenecen a la clase y "-1" las muestras negativas o que no pertenecen a la clase.

La solución deberá proveer un clasificador que permita determinar si las nuevas muestras son positivas o negativas y la precisión dependerá entre otras cosas de su poder de generalización al aprender sobre un conjunto de entrenamiento.

Los clasificadores convencionales como clasificadores de Vecinos Cercanos (Nearest Neighbor) (Arbach et al., 2003) (Tan, 2005) (Hart, 1968), clasificadores Bayesianos (Zhang et al., 2006) (Cervantes et al., 2009), Redes Neuronales Artificiales (Sotolongo and Guzmán, 2001) (Peterson et al., 2005) (Makal. and

Ozyilmaz, 2007) (Makal et al., 2008), Árboles de Decisión (Segovia-Juarez et al., 2007)y las Máquinas de Vectores Soporte (SVM)(Bazzani et al., 2000) (Bobadilla et al., 2003) (Marangoni et al., 2003) (Kuan-ming and Chih-jen, 2003) permiten distinguir entre clases binarias, pero, requieren que el conjunto con el que entrenan sea balanceado, esto es, que contengan muestras positivas y negativas de tamaños similares; sin embargo, en el mundo real, los problemas a menudo presentan conjuntos de datos no balanceados, es decir, contiene un gran número de muestras para la primera clase (clase mayoritaria) y un número muy reducido de muestras para la segunda clase (clase minoritaria).

Las Máquinas de Vectores Soporte (SVM) son actualmente una de las técnicas de clasificación más importantes (Kong et al., 2004) (Dror et al., 2005)debido a que tienen un mejor desempeño frente a los métodos anteriormente citados. El poder de generalización de las SVM es una de sus características más importantes, la razón de esta propiedad puede ser explicada por la teoría de aprendizaje estadístico (Vapnik, 1995)relacionado con maximizar el margen de separación entre los hiperplanos de cada clase, otra ventaja de las SVM radica en que obtienen un subconjunto de Vectores Soporte durante la fase de aprendizaje, subconjunto que a menudo es sólo una pequeña parte del conjunto de datos original (Chih-Chia and Pao-Ta, 2007), sin embargo, para encontrar el hiperplano de separación las SVM necesitan resolver un problema de programación cuadrática (QP), que involucra una matriz de densidad $N \times N$, donde N es el número de puntos en el conjunto de datos. Esto provoca que la complejidad del entrenamiento de las SVM sea altamente dependiente del tamaño del conjunto de datos, requiriendo grandes cantidades de tiempo computacional y memoria para conjuntos de datos muy grandes (Xiaoou et al., 2010). Otra gran desventaja ha sido mencionada en investigaciones recientes (Akbani et al., 2004) (Zeng and Gao, 2009) (Köknar-Tezel and Latecki, 2009) donde se ha mostrado

que el desempeño de las SVM es afectado cuando son utilizadas en conjuntos no balanceados, haciendo más evidente esto cuando el radio de desbalance es muy alto, este problema se debe a que los clasificadores en general están diseñados para reducir el error promedio global sin importar la distribución de las clases y por ello la frontera de decisión es desplazada hacia la clase mayoritaria. En particular, las SVM a pesar de encontrar el máximo margen de separación, cuando se entrenan con conjuntos con alto radio de desbalance, el hiperplano de separación que obtienen se sesga hacia la clase mayoritaria, provocando un impacto negativo en la precisión de clasificación puesto que la clase minoritaria, al tener muy pocas muestras, puede ser considerada como ruido y por consiguiente ignorada por el clasificador.

En el mundo real, existen aplicaciones que presentan un desbalance muy acentuado, por ejemplo en problemas de detección de fraudes, donde el radio de desbalance puede ir de 100 a 1 hasta 100,000 a 1 (Provost and Fawcett, 2001), interpretándose como un fraude frente a cien mil que no lo son, otros ejemplos son la clasificación de secuencias de proteína (Sonnenburg et al., 2005) (Dror et al., 2005) (Cervantes et al., 2009) (Segovia-Juarez et al., 2007) donde sólo unas fracciones de la secuencia completa de ADN, denominados exones, codifican en proteína, otros ejemplos son el diagnóstico médico (Kononenko, 2001) (Grzymala-Busse et al., 2005), la detección de intrusos y la clasificación de texto (Sebastiani, 2002) (Tan, 2005).

El desarrollo de nuevas técnicas para mejorar el desempeño de las SVM en conjuntos no balanceados es un reto importante en el área de reconocimiento de patrones, minería de datos y aprendizaje de máquinas. Para abordar este problema algunos autores han propuesto métodos para reducir el efecto negativo de los conjuntos desbalanceados. Técnicas como Bajo-muestreo (Under-sampling) balancean el conjunto, reduciendo la clase mayoritaria hasta obtener un subconjunto, selec-

cionado de forma aleatoria, que tenga la misma cantidad de muestras que la clase minoritaria; por otro lado, la técnica de Sobre-muestreo (Over-sampling) duplica la clase minoritaria tantas veces como sea necesario hasta equilibrar el tamaño entre clases (Akbani et al., 2004). Una desventaja de usar Bajo-muestreo sucede al eliminar datos de forma aleatoria de la clase mayoritaria, ya que se podría estar eliminando datos importantes en la frontera de decisión, provocando que el hiperplano de separación no sea óptimo al usar una SVM, afectando la sensitividad y la especificidad. Por otro lado, al utilizar Sobre-muestreo y duplicar datos de la clase minoritaria, el tiempo de entrenamiento se incrementará significativamente.

Chawla propuso *Synthetic Minority Over Sampling Technique* (SMOTE), técnica que genera puntos sintéticos para después incluirlos como miembros de la clase minoritaria con el objetivo de reducir el desbalance entre clases. Para crear los puntos sintéticos, SMOTE toma un punto de la clase minoritaria y produce una nueva versión del dato al desplazarlo hacia su vecino más cercano una distancia aleatoria para cada dimensión. Una desventaja de esta técnica es que aunque trabaja sólo con la clase minoritaria, de todos los puntos sintéticos creados por cada muestra positiva, sólo se elegirá un subgrupo seleccionado al azar. De acuerdo a los resultados presentados en (Chawla et al., 2002) esta técnica es mejor que Under-sampling y Over-sampling, sin embargo, estas tres operan sobre el espacio de entrada.

Una combinación de SMOTE y Over-sampling fue propuesta en (Akbani et al., 2004), este método introduce un esquema de penalización del error dependiendo de la clase, decrementando el costo de la clase mayoritaria e incrementándolo en la clase minoritaria, tal combinación logra una mayor densidad en la distribución de la clase minoritaria y coloca el hiperplano de separación más cerca de la clase mayoritaria. En (Veropoulos et al., 1999) diferentes criterios de penalización son usados para producir efectos similares en la separación del hiperplano. Otras

propuestas inspiradas en SMOTE pueden encontrarse en (Hart, 1968) (Guo, 2004) (Han et al., 2005) (Hu et al., 2009). Otro enfoque se basa en aplicar Under-sampling sobre conjuntos no balanceados pero con una selección por medio de un algoritmo genético, resultando en mejores resultados que un simple muestreo aleatorio (Zou et al., 2008).

1.2. Planteamiento del problema

Las aplicaciones de clasificación en el mundo real a menudo se presentan con conjuntos de datos no balanceados, ocasionando que incluso los clasificadores como las SVM, vean sesgado su desempeño durante el entrenamiento debido a que son entrenadas con más muestras negativas que positivas, favoreciendo a la clase mayoritaria y presentando una baja precisión al tratar de clasificar los datos importantes etiquetados como muestras positivas.

La solución está en equilibrar el tamaño de los conjuntos de datos, sin embargo, en muchos casos las muestras son difíciles de obtener, debido a que estos eventos ocurren con poca frecuencia o presentan una distribución muy reducida frente a otra muy elevada, siendo necesario balancear el tamaño del conjunto de entrenamiento con las muestras disponibles a fin de reducir el sesgo en la hipótesis final.

La solución ofrecida por el bajo-muestreo permite equilibrar el tamaño entre clases al seleccionar aleatoriamente un subconjunto de muestras negativas, sin embargo, este tipo de selección podría dejar fuera muestras importantes, otra alternativa es replicar las muestras positivas (Sobre-muestreo), pero al no estar introduciendo nuevas muestras, podría provocar un sobre entrenamiento en el clasificador.

El problema principal yace en encontrar nuevas muestras positivas que ayuden a reforzar el entrenamiento y evitar el sesgo en la regla de clasificación. La técnica SMOTE (Chawla et al., 2002), introduce nuevos puntos sintéticos en el conjunto de entrenamiento sin embargo, utiliza todo el conjunto original para generar nuevos puntos, por lo que no asegura que los nuevos puntos sean relevantes, otro problema de esta técnica es que los nuevos puntos podrían introducir ruido debido a que mientras más vecinos utilice, los puntos sintéticos generados podrían quedar localizados dentro del espacio de la clase contraria, SMOTE funciona bien para patrones bidimensionales pero presenta una precisión de clasificación sesgada hacia una de las dos clases cuando el número de características es elevado.

Otro problema que se presenta al trabajar con conjuntos de datos no balanceados es que debe definirse una métrica capaz de evaluar la precisión real del clasificador, que depende de los verdaderos positivos, verdaderos negativos, asimismo deberá ser capaz de mostrar el sesgo de la precisión. En estos casos la métrica *accuracy* no tiene buen desempeño y puede llevarnos a conclusiones erróneas debido a que la clase minoritaria tiene un impacto pequeño en la precisión en comparación con la clase mayoritaria. Por ejemplo, en un desbalance con 1 muestra positiva por cada 99 muestras negativas, un clasificador que etiquete todas las muestras como negativas obtendría una precisión de 99 % para las muestras negativas pero una precisión de 0 % al intentar clasificar las muestras positivas. Es por ello que la evaluación del desempeño de clasificación juega un rol muy importante en el diseño de un algoritmo y el uso de una métrica apropiada es tan importante como la selección de parámetros.

1.3. Justificación

En la actualidad la mayoría de los conjuntos de datos son no balanceados, presentes en aplicaciones como detección de fraudes, donde un fraude puede ocurrir frente a otras mil transacciones que no lo son, otro ejemplo es el filtrado de correo *spam*, donde el correo deseado regularmente se presenta en menor proporción frente al correo no deseado. En bioinformática, el problema de clasificación de secuencias de ADN que codifican en proteína es otro claro ejemplo de desbalance ya que de las miles de secuencias que contiene el ADN solo una pequeña fracción codifica en proteína.

Las aplicaciones médicas para diagnosticar enfermedades también trabajan con conjuntos no balanceados, en estas, los casos de personas diagnosticadas como enfermas se presenta en menor proporción que aquellas que no lo están, haciendo indispensable una buena precisión de clasificación para diagnosticar correctamente y proveerlos del tratamiento apropiado. Una mala precisión significaría un gran costo en algunas situaciones, como un mal diagnóstico de cáncer o algún síndrome.

El balanceo de clases es entonces un problema muy importante en aprendizaje de máquinas, ya que si el conjunto con el que se entrena el clasificador esta no balanceado, la precisión de clasificación se verá afectada, sesgando las predicciones hacia una de las dos clases.

Las SVM ofrecen la posibilidad de trabajar con un espacio altamente dimensional, diferente al espacio de entrada que es donde operan las técnicas clásicas como Bajo-muestreo, Sobre-muestreo y SMOTE; además proveen de un conjunto reducido de puntos llamados Vectores Soporte, que son los más importantes y que sirven para generar el hiperplano de separación. Resulta claro que para mejorar el desempeño de las SVM, deberían aprovecharse las cualidades antes mencionadas,

por lo que los métodos que se presentarán en esta tesis, operarán en el espacio de características a fin de reducir el desbalance entre las clases o reducir el sesgo del hiperplano de separación basándose en los SV.

El diseñar métodos que puedan superar los inconvenientes del desbalance entre clases, es un reto importante y puede beneficiar a la comunidad científica al proveer de una nueva técnica de balanceo de clases al mismo tiempo que mejora el desempeño de clasificación de las SVM.

1.4. Objetivos y Metas

1.4.1. Objetivo General

Implementar dos algoritmos basado en los Vectores Soporte para mejorar el desempeño de clasificación de las SVM sobre conjuntos de datos no balanceados.

1.4.2. Objetivos Específicos

1. Estudiar el estado del arte sobre el problema de clasificación con conjuntos no balanceados.

2. Diseñar algoritmos para disminuir el sesgo de clasificación.

3. Realizar simulaciones y validar los algoritmos propuestos.

4. Obtener parámetros óptimos.

5. Estudiar los resultados.

1.5. Hipótesis

Es posible reducir el sesgo de clasificación de la SVM, ocasionado por el desbalance entre clases, al crear nuevos puntos artificiales positivos tomando como referencia los Vectores Soporte.

1.6. Organización de la tesis

La presente tesis está organizada de la siguiente forma: en el primer Capítulo, como anteriormente se mostró, presenta una introducción, abordando el problema del desbalance de clases en los conjuntos de datos y como esto afecta el desempeño de clasificación de las SVM, en el segundo Capítulo se incluyen los fundamentos necesarios para abordar la metodología del Capítulo tres, Sección donde se proponen dos nuevos métodos para reducir el sesgo del hiperplano de clasificación de las SVM primero a través de la excitación de SV y después a través de la creación de puntos artificiales positivos. Los resultados experimentales son mostrados en el cuarto Capítulo, primero usando técnicas clásicas y después usando los dos métodos propuestos. Adicionalmente se incluyen dos ejemplos ilustrativos, donde dos conjuntos no balanceados, uno con radio de desbalance pequeño y otro con alto radio de desbalance, son clasificados usando los métodos propuestos. Al final se discuten los resultados obtenidos de las pruebas y se presentan las conclusiones sobre este trabajo de investigación.

Capítulo 2

Preliminares

En este Capítulo se exponen los fundamentos necesarios para abordar el problema de clasificación de conjuntos binarios no balanceados usando SVM así como los conceptos usados en los métodos propuestos, que seran presentados en Capítulos posteriores.

2.1. Tipos de entrenamiento

El entrenamiento es una de las herramientas que el aprendizaje de máquinas proporciona para agilizar el aprendizaje. Este proceso consiste en ir ajustando los pesos (w) gradualmente hasta que el *vector de salida resultante* coincida con el *vector de salida deseado.*

2.1.1. Entrenamiento supervisado

El entrenamiento supervisado parte de un *vector de entrada* del cual se conoce su *vector de salida deseada* o al menos la aproximación a este. A los *vectores*

de entrada y salida deseada se les denomina *par de entrenamiento*. Este proceso consiste en aplicar el *vector de entrada* al modelo del clasificador. La diferencia o cambió existente entre el *vector de salida resultante* y el *vector de salida deseada* se reduce a través de diversos algoritmos existentes. El objetivo es continuar probando diversos vectores de entrada y ajustar el *vector de pesos*, hasta que la diferencia con la salida deseada sea mínima.

2.1.2. Entrenamiento no supervisado

En el entrenamiento no supervisado se desconoce la salida, únicamente se proporciona un *vector de entrada*. Lo que se busca es generar después de varios *vectores de entrada*, salidas consistentes, es decir que los *pesos* se vayan ajustando poco a poco a través del reconocimiento de patrones, regularidades, propiedades estáticas, etc. Así, las entradas similares producirán el mismo tipo de salida. Otra forma de explicar esto, es que este proceso extrae propiedades estáticas del conjunto de entrenamiento.

2.2. *Support Vector Machines*

Las máquinas de vectores soporte (SVM) son uno de los métodos de clasificación más usados para el modelado y clasificación de datos, recientemente clasificado junto con los *métodos kernel*. La ventaja de las SVM es su excelente capacidad de generalización, que concierne en la capacidad de clasificar correctamente ejemplos que no están dentro de los rasgos de espacio usados en el entrenamiento (Nixon and Aguado, 2008). Este método de clasificación es ampliamente usado en la bioinformática (entre otras disciplinas) debido a su exactitud, habilidad para tratar con

grandes cantidades de datos y la flexibilidad en la modelación de diversas fuentes de información.

Cuando se entrena una SVM se necesita realizar algunos pasos: procesar los datos, definir el método de *kernel* usar y finalmente preparar los parámetros de la SVM.

2.2.1. Clasificador lineal

Los datos para un problema de aprendizaje de dos clases consta de objetos etiquetados con una de las dos etiquetas correspondientes a las clases disponibles, por conveniencia se asume que las etiquetas "+1" son ejemplos positivos y "-1" son ejemplos negativos (Ben-Hur and Weston, 2010).

Teniendo el vector x con componentes x_i. Estos componentes denotan los $i-ésimos$ vectores de un conjunto de datos $\{(x_i, y_i)\}_i^n = 1, donde\ y_i$ es la etiqueta asociada a x_i. Los objetos x_i son llamados modelos o ejemplos. Se asume que los ejemplos pertenecen a un conjunto X. Inicialmente se toman los ejemplos como vectores, pero una vez introducidos en el modelo esta afirmación podría ser diferente pudiendo ser entonces conjuntos u objetos discretos.

El concepto principal para definir un clasificador es el producto punto (*dot product*) entre dos vectores, también conocido como producto escalar o producto interno, definido como $w^T x = \sum_i w_i x_i$.Un clasificador lineal está basado en la función:

$$f(x) = w^T x + b \tag{2.1}$$

El vector w es conocido como el vector de peso, y b es el bias. Considerando el caso $b = 0$, el conjunto de puntos de x tal que $w^T x = 0$ son todos los puntos que son

perpendiculares a w y pasan por el origen en el plano de dos dimensiones, un plano de tres dimensiones, y en un hiperplano. La tendencia b traslada el hiperplano fuera del origen. El hiperplano:

$$\{x : f(x) = w^T x + b = 0\} \tag{2.2}$$

Divide el espacio en dos partes, el signo de la función discriminativa $f(x)$ denota el lado del hiperplano en donde se encuentra el punto. El límite entre las regiones positiva y negativa del plano son llamadas el límite de decisión del clasificador.

2.2.2. Clasificador no lineal

En diversas aplicaciones los clasificadores no lineales proveen una mejor exactitud. Aún así los clasificadores lineales tienen ventajas, una de ellas es que cuentan con algoritmos de entrenamiento simples. Podemos fácilmente observar el comportamiento de ambos tipos de clasificadores mapeando los datos de un conjunto X usando la función no lineal $\phi : X \longrightarrow F$. En el espacio F la función discriminante es:

$$f(x) = w^T \phi(x) + b \tag{2.3}$$

Considerando el caso de un espacio bidimensional:

$$\phi(x) = (x_i^2, \sqrt{2}x_1 x_2, x_2^2)^T \tag{2.4}$$

Representando un vector en términos de monomios de segundo grado. En este caso

$$w^T \phi(x) = w_1 x_1^2 + \sqrt{2} w_2 x_1 x_2 + w_3 x_2^2 \tag{2.5}$$

Resultando en un límite de decisión para el clasificador, $f(x) = w^T x + b = 0$, que es una sección cónica.

La aproximación al introducir rasgos no lineales no es fácil de escalar debido al número de rasgos de entrada. Cuando se mapea el ejemplo anterior la dimensionalidad de las características del espacio F es cuadrática en comparación de la dimensionalidad del espacio original. Esta complejidad cuadrática es factible con un espacio dimensional reducido; pero cuando se trata de datos con miles de dimensiones, la complejidad cuadrática en el número de dimensiones no es aceptable. Los métodos *kernel* resuelven este caso evitando el paso del mapeo de datos de grandes dimensiones.

Debido a que el peso del vector puede ser expresado como una combinación de ejemplos lineales de entrenamiento.

$$w = \sum_{i=1}^{n} \alpha_i x_i \tag{2.6}$$

entonces

$$f(x) = \sum_{i=1}^{n} \alpha_i^T x + b \tag{2.7}$$

En el espacio de características F se forma la siguiente expresión:

$$f(x) = \sum_{i=1}^{n} \alpha_i \phi(x_i)^T \phi(x) + b \tag{2.8}$$

La representación en términos de α_i es conocida como la representación del límite de decisión. Como se indicó anteriormente el espacio F puede ser de gran dimensión haciendo este truco poco práctico a no ser que la función *kernel* $k(x, x')$ se defina

como:

$$k(x, x') = \phi(x)^T \phi(x') \tag{2.9}$$

Pudiendo procesarlo eficientemente. En términos de la función de *kernel* la función de decisión es:

$$f(x) = \sum_{i=1}^{n} \alpha_i (x.x_i) + b \tag{2.10}$$

2.2.3. Margen geométrico

Para un hiperplano dado entendemos por x_+ o x_- como el punto más cercano entre los ejemplos negativos o positivos. La norma de un vector w, denotado por $\|w\|$, es su longitud y está dada por $\sqrt{w^T w}$. Un vector unitario $\widehat{w}$ en dirección de w está dado por $w/\|w\|$ teniendo $\|w\| = 1$. De consideraciones geométricas simple consideramos un hiperplano f con respecto a un conjunto de datos D de la forma:

$$m_D(f) = \frac{1}{2} \widehat{w}^T (x_+ - x_-) \tag{2.11}$$

donde $\widehat{w}$ es un vector unitario en dirección de w, asumiendo que x_+ y x_- son equidistantes del límite de decisión.

$$f(x) = w^T x_+ + b = a \tag{2.12}$$

$$f(x) = w^T x_- + b = -a \tag{2.13}$$

para una constante $a > 0$. Para hacer el margen geométrico significativo ajustamos el valor de la función de decisión a los puntos más cercanos del hiperplano, y establecemos $a = 1$.

Adicionando las dos ecuaciones y dividiéndolas por $\|w\|$ obtenemos:

$$m_D(f) = \frac{1}{2}\widehat{w}^T(x_+ - x_-) = \frac{1}{\|w\|} \tag{2.14}$$

2.2.4. Clasificador de margen máximo

Los clasificadores de margen máximo poseen una función discriminante que maximiza el margen geométrico $\frac{1}{\|w\|}$ que es equivalente a minimizar $\|w\|^2$. Tomando lo anterior

$$\begin{aligned} &minimizar\ \frac{1}{2}\|w\|^2 \\ sujeto\ a\ \ :\ \ &y_i(w^T x_i + b) \geq 1 \qquad i = 1, ..., n \end{aligned} \tag{2.15}$$

Las implicantes en esta formulación aseguran que el máximo clasificador clasificará cada ejemplo correctamente, esto es posible asumiendo que los conjuntos de datos son linealmente separables. En la práctica en ocasiones los conjuntos no son linealmente separables; y aunque esta se pudiera separar, el margen más adecuado puede ser alcanzado permitiendo que el clasificador no clasifique algunos puntos. Para permitir errores reemplazamos las coacciones desiguales de la ecuación anterior

$$y_i(w^T x_i + b) \geq 1 - \xi_i \qquad i = 1, ..., n \tag{2.16}$$

donde $\xi_i \geq 0$ son variables de poca utilidad que permiten a un ejemplo estar dentro del margen de $0 \leq \xi_i \leq 1$ (también llamado margen de error) o no ser clasificados ($\xi_i > 1$). Por lo tanto la variable no es clasificada cuando su valor es mayor a 1, $\sum_i \xi_i$ es el conjunto de ejemplos que no son clasificados. El objetivo de maximizar el margen, minimizando $\frac{1}{2}\|w\|^2$ es demostrado en la ecuación $C\sum_i \xi_i$

para penalizar el error de margen y los elementos no clasificados.

$$\begin{aligned} & \underset{w,b}{minimizar} && \frac{1}{2}\|w\|^2 + C\sum_i \xi_i \\ sujeto\ a\ \ : \ \ & y_i(w^T x_i + b) \geq 1 - \xi_i && \xi_i \geq 0 \end{aligned} \tag{2.17}$$

La constante C establece la importancia relativa de maximizar el margen y minimizar la cantidad de ejemplos de poca utilidad. Esta formulación es conocida como margen blando SVM (*soft-margin*). Utilizando el método de multiplicadores de Lagrange obtenemos la formulación dual:

$$\begin{aligned} & \underset{\alpha}{min} && \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j x_i^T x_j \\ sujeto\ a\ \ : \ \ & \sum_{i=1}^{n} y_i \alpha_i = 0 && 0 \leq \alpha_i \leq C \end{aligned} \tag{2.18}$$

La formulación dual guía una expansión del vector de peso en términos de los ejemplos de entrada:

$$w = \sum_{i=1}^{n} y_i \alpha_i x_i \tag{2.19}$$

Los ejemplos x_i para cada $a_i > 0$ son los puntos que no se encuentran en el margen cuando es usado el margen blando. Estos son los llamados vectores de soporte. La expansión en términos de los vectores de soporte es algunas veces escasa, y el nivel de escases es mayor en el porcentaje de error del clasificador. El problema de la formulación de la optimización de las SVM depende de la información sólo a través de los productos punto. El producto punto se puede por lo tanto reemplazar por una función *kernel* no lineal, proporcionando así un amplio margen de separación en el espacio de características.

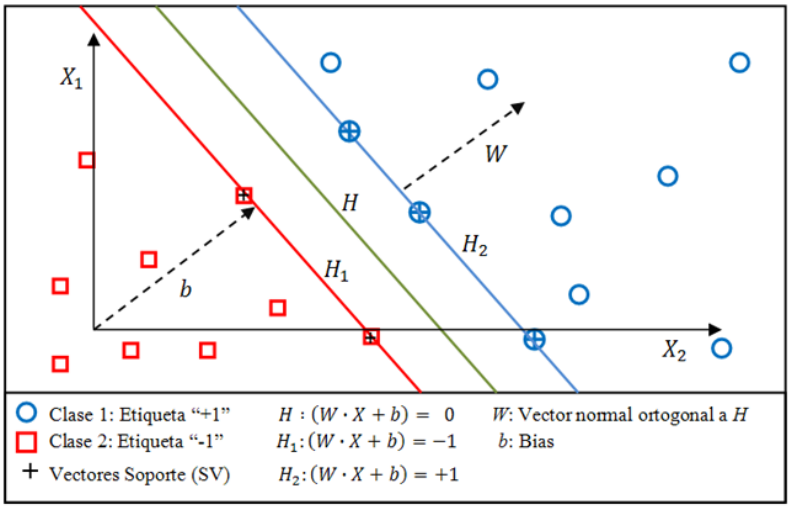


Figura 2.1: Clasificador con margen máximo

2.2.5. Condiciones de Karush-Kuhn-Tucker

El análisis denominado Karush Kuhn Tucker (KKT) demuestra que la gran mayoría de los coeficientes de Lagrange son cero, y que sólo pueden ser distintos de cero para los vectores soporte, puntos que se encuentran exactamente a la distancia marcada por el margen. Al dualizar el modelo de maximización del margen se transforma en un problema de minimización de una función cuadrática convexa sujeta a restricciones lineales.

2.2.6. Aprendizaje con Kernels

El limitado poder computacional de máquinas con aprendizaje lineal fue hecho notar en 1960 por Minsky y Papert. En general, las aplicaciones del mundo real requieren un espacio con más dimensiones para poder crear una hipótesis más allá de lo que pueden ofrecer las funciones lineales. Otra forma de ver este problema es que

frecuentemente la solución no puede ser expresada como una simple combinación lineal de los atributos dados, requiriendo de más características abstractas. Como solución a este problema múltiples capas con funciones lineales han sido propuestas resultando en el desarrollo de redes neuronales artificiales multicapa y algoritmos de aprendizaje como *back-propagation* para entrenar tales sistemas.

La representación como Kernel ofrece una solución alternativa a través de proyectar los datos dentro de un espacio altamente dimensional llamado "espacio de características" con el objetivo de incrementar el poder computacional de las máquinas con aprendizaje lineal.

La complejidad de la función objetivo a ser aprendida depende de la forma en cómo esta es representada y la dificultad de la tarea de aprendizaje puede variar de acuerdo a ello. Entonces una estrategia de preprocesamiento común en aprendizaje de máquinas involucra cambiar la representación de los datos:

$$x = (x_1, ..., x_n) \longmapsto \phi(x) = (\phi_1(x), ..., \phi_N(x)) \tag{2.20}$$

Este paso es equivalente a mapear el espacio de entrada X dentro de un nuevo espacio, $F = \{\phi(x)|x \, \epsilon \, X\}$. Las cantidades introducidas para describir los datos son usualmente llamadas "*rasgos o características*", mientras las cantidades originales son llamadas "*atributos*". La tarea de elegir la representación más conveniente es conocida como "*selección de características*". El espacio X corresponde entonces al *espacio de entrada*, mientras F es llamado *espacio de características*.

La siguiente Figura 2.2 muestra un ejemplo del mapeo de características, desde un espacio de entrada con dos dimensiones hacia un espacio de características con tres dimensiones, donde inicialmente los datos no pueden ser separados por una función lineal dentro del espacio de entrada, pero si es posible dentro del

espacio de características.Existen diferentes aproximaciones para la selección de

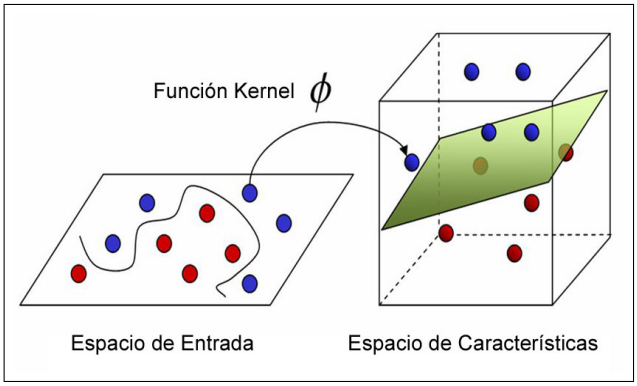


Figura 2.2: Un mapeo al espacio de características puede simplificar la tarea de clasificación

características. Frecuentemente se busca identificar el conjunto de características más pequeño que mantenga la información esencial de los atributos originales. Esto es conocido como *reducción de la dimensionalidad*.

$$x = (x_1, ..., x_n) \longmapsto \phi(x) = (\phi_1(x), ..., \phi_d(x)), d < n \tag{2.21}$$

Otra forma de selección de características se refiere a la detección de *rasgos irrelevantes*. El uso del Análisis de Componentes Principales provee de un mapeo de los datos hacia un espacio de características donde los nuevos rasgos son funciones lineales de los atributos originales y son ordenados por la cantidad de varianza que exhiben los datos en cada dirección.

El conjunto de hipótesis que consideraremos serán funciones del siguiente tipo:

$$f(x) = \sum_{i=1}^{N} w_i \phi(x) + b \tag{2.22}$$

donde $\phi : X \rightarrow F$ es un mapeo no lineal desde el espacio de entrada hacia

el espacio de características. Esto significa que se pueden construir máquinas no lineales en dos pasos: primero aplicando un mapeo no lineal y después usando una máquina lineal para clasificar dentro del espacio de características.

Una propiedad importante de las máquinas con aprendizaje lineal es que estas pueden ser expresadas en una representación dual, esto significa que la hipótesis 2.22 puede ser expresada como una combinación lineal de los puntos de entrenamiento y que la regla de decisión puede ser evaluada usando el producto interno entre el punto de prueba y el punto de entrenamiento:

$$f(x) = \sum_{i=1}^{l} \alpha_i y_i \langle \phi(x_i) \cdot \phi(x) \rangle + b \tag{2.23}$$

Si se cuenta con una forma de calcular el producto interno $\langle \phi(x_i) \cdot \phi(x) \rangle$ directamente en el espacio de características como una función de los puntos de entrada originales, esto hace posible unir los dos pasos necesarios para construir una máquina con aprendizaje no lineal. Este método de cálculo directo es llamado función *kernel*.

Definición 2.2.1 *Un kernel es una función K, tal que para todo x,z* ϵ *X.* $K(x, z) = \langle \phi(x) \cdot \phi(z) \rangle$, *donde* ϕ *es un mapeo desde X hacia un (producto punto) espacio de características F*

La clave de esta aproximación es encontrar una función kernel que pueda ser evaluada eficientemente. Una vez que contamos con tal función, la regla de decisión puede ser evaluada por al menos l evaluaciones del kernel:

$$f(x) = \sum_{i=1}^{l} \alpha_i y_i K(x_i, x) + b \tag{2.24}$$

Una función kernel es un producto interno en el espacio de características, que tiene su equivalente en el espacio de entrada donde K, es una función simétrica positiva definida que cumple las condiciones de Mercer.

Entre los kernels más comunes, se encuentran: la función lineal (Fórmula 2.25), polinomial (Fórmula 2.26), función de base radial (Fórmula 2.27), sigmoidal (Fórmula 2.28), ERBF (Exponential Radial Basis Function), entre otros.

$$K(X_i, X_j) = X_i \cdot X_j \tag{2.25}$$

$$K(X_i, X_j) = (\gamma(X_i \cdot X_j) + \theta)^d \tag{2.26}$$

$$K(X_i, X_j) = \exp(-\gamma||X_i - X_j||^2), \ \ donde\ \gamma > 0, \ \gamma = \frac{1}{2\sigma^2} \tag{2.27}$$

$$K(X_i, X_j) = \tanh(\gamma(X_i \cdot X_j) + \theta) \tag{2.28}$$

En la Figura 2.4 se muestra en tres dimensiones el comportamiento de la función

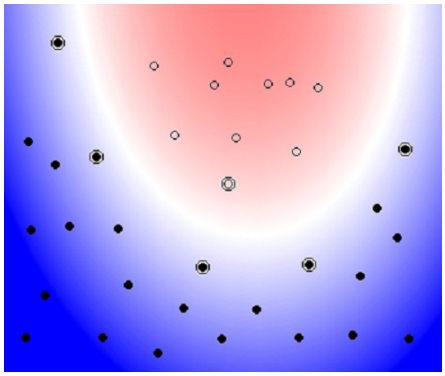


Figura 2.3: Clasificación con Kernel polinomial de grado $d = 2$

de base radial (RBF) con sigma (σ) igual a 0,5.

La salida de la función kernel $K(X_i, X_j)$, depende de la distancia entre el Vector Soporte (X_i) y el punto de prueba (X_j). El SV se ubica en el centro de

la Gráfica y el área de influencia es determinada por σ; cuando este valor es grande, el área de influencia también será grande, permitiendo obtener un hiperplano de separación más liso y regular, reduciendo la cantidad de SV necesarios para definir este hiperplano puesto que cada SV, al tener una fuerte influencia sobre los demás puntos, puede cubrir un espacio grande. Por otro lado cuando σ es muy pequeño, se requieren muchos SV para cubrir un espacio grande, creando un hiperplano de separación menos liso. Esto implica que para lograr una buena generalización hay que proporcionar un buen valor para σ, sin embargo esto dependerá de que tan empalmadas se encuentren las clases.

Un ejemplo de clasificación para conjuntos no linealmente separables se muestra en la Figura 2.5.

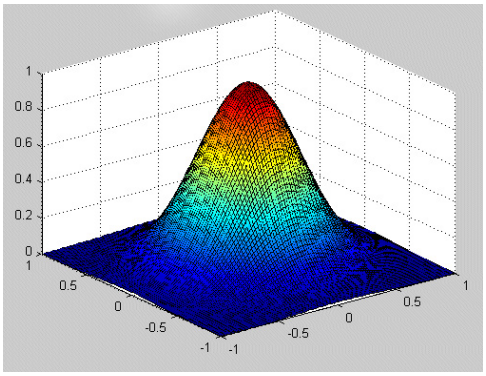


Figura 2.4: Función de Base Radial con $\sigma = 0{,}5$

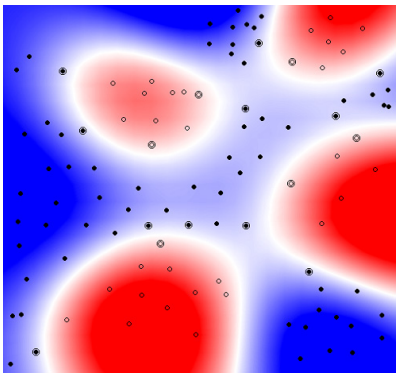


Figura 2.5: Clasificación con Kernel RBF

2.2.7. Condición de Mercer

Existe un mapeo ϕ y una expansión

$$K(x,y) = \sum_i \phi_i(x)\phi_i(y) \tag{2.29}$$

si y sólo si, para algún $g(x)$ tal que

$$\int g(x)^2 dx \text{ es finita} \tag{2.30}$$

entonces

$$\int K(x,y)g(x)g(y)dxdy \geq 0 \tag{2.31}$$

Note que para casos específicos, puede no ser fácil verificar si la condición de Mercer es satisfecha. La ecuación 2.31 debe mantener para cada g con norma finita L_2 (cumpliendo la ecuación 2.30).

2.2.8. Optimización de secuencia mínima (SMO)

El algoritmo de optimización mínima secuencial (SMO por sus siglas en ingles Sequential Minimal Optimisation) es un algoritmo que agiliza el entrenamiento, se deriva tomando la idea del método de descomposición a su extremo y optimizando un subconjunto mínimo de sólo dos puntos en cada iteración. El poder de esta técnica reside en el hecho de que el problema de optimización para dos puntos admite una solución analítica, eliminando la necesidad de usar un programa de optimización cuadrática iterativa como parte del algoritmo.

El requisito es hacer cumplir la condición $\sum_{i=1}^{l} \alpha_i y_i = 0$ durante las iteraciones implicando que el número más pequeño de multiplicadores que pueden ser optimizados en cada paso son 2, por lo que cada vez que un multiplicador es actualizado, por lo menos otro multiplicador necesita ser ajustado con el objetivo de mantener válida la condición.

En cada paso SMO elije dos elementos α_i y α_j para optimizarlos, encuentra los valores óptimos para esos dos parámetros dado que todos los otros son fijos, y por consiguiente actualiza el vector α. La selección de los dos puntos es determinado por una heurística, mientras la optimización de los dos multiplicadores es realizado analíticamente.

A pesar de necesitar más iteraciones para converger, cada iteración usa unas pocas operaciones tal que el algoritmo muestra una alta velocidad de algunos órdenes de magnitud.

SMO divide un problema con gran QP (programación cuadrática) en una serie de pequeños problemas QP, que son resueltos rápidamente y analíticamente, generalmente esto mejora significativamente el tiempo de procesamiento requerido.

SMO ha sido probado en problemas tanto para el mundo real como para

problemas artificiales. De estas pruebas se puede deducir lo siguiente:

1. SMO puede ser usado cuando el usuario no tiene acceso a la programación cuadrática o su implementación es muy tardada.

2. SMO beneficia a las SVM que tienen muchos de sus multiplicadores de Lagrange cerca de la frontera límite.

3. SMO se desempeña bien para SVM lineales debido a que el tiempo de procesamiento es en su mayoría debido a la evaluación de la SVM, y la evaluación de la SVM lineal puede ser expresada como un simple producto punto en lugar de una suma de kernels lineales.

4. SMO se desempeña bien para SVM con datos de entrada esparcidos, incluso para SVM no lineales, debido a que el tiempo de procesamiento usado por el kernel, puede ser reducido, esto acelera directamente a SMO.

5. SMO se desempeña bien para problemas con grandes conjuntos de datos debido a que el escalamiento del tamaño para el conjunto de entrenamiento es mejor que el propuesto por Chunking para la mayoría de los problemas probados hasta ahora.

Mas allá del tiempo de convergencia, otro rasgo importante del algoritmo es que este no necesita almacenar la matriz kernel en la memoria, dado que las operaciones con matrices no están involucradas. Nótese que el SMO no usa una matriz kernel con cache y la introducción de esta podría ser usada para obtener una mayor velocidad a expensas de incrementar la complejidad espacial (Platt, 1998).

2.3. Técnicas de balanceo de clases

2.3.1. *Under-sampling*

El Bajo-muestreo es una técnica de balanceo de clases que opera a nivel de los datos de entrada, se enfoca en lograr el equilibrio a través de la reducción de la clase mayoritaria. Para lograr la reducción se eligen i elementos de todo el conjunto de muestras negativas pertenecientes a la clase mayoritaria, siendo i un valor cercano al tamaño de la clase minoritaria sin embargo, esta medida podría eliminar datos importantes para el aprendizaje debido a que usa una selección aleatoria para reducir el conjunto de entrada. Si comparamos el hiperplano ideal y el aprendido por una SVM con Under-sampling veríamos que se forma un ángulo significativamente grande entre los dos debido al sesgo que no pudo mejorar el Bajo-muestreo, como se mostró experimentalmente en (Akbani et al., 2004).

2.3.2. *Over-sampling*

El Sobre-muestreo es una técnica de balanceo de clases que opera a nivel de los datos de entrada, se enfoca en lograr el equilibrio a través de incrementar el tamaño de la clase minoritaria. Para lograrlo se agregan todas las muestras positivas del conjunto de entrada original y se repite este paso tantas veces hasta que iguale o se acerque sin rebasar el tamaño de la clase mayoritaria. En caso de faltar más muestras para equilibrar las clases, se eligen aleatoriamente las muestras faltantes de la clase minoritaria original sin embargo, esta técnica no genera nuevos puntos, sólo repite los datos de la clase minoritaria ocasionando que no mejore significativamente la precisión en una SVM y adicionalmente incrementa el tiempo de entrenamiento (Akbani et al., 2004).

2.3.3. *Synthetic Minority Over-sampling Technique (SMOTE)*

La técnica SMOTE fue propuesta por en (Chawla et al., 2002), esta aproximación sobre-muestrea la clase minoritaria a través de la creación de "puntos sintéticos". La clase minoritaria es sobre-muestreada tomando cada muestra positiva e introduciendo nuevas muestras sintéticas sobre el segmento de línea que une esta muestra con los k vecinos más cercanos dentro de la clase minoritaria. Dependiendo de la cantidad de sobre-muestreo requerido, son seleccionados aleatoriamente los vecinos del conjunto de k vecinos más cercanos, siendo lo más recomendado un valor de cinco para k.

Por ejemplo, si la cantidad de sobre-muestreo requerida es de 200 %, sólo dos vecinos del conjunto de k vecinos cercanos serán elegidos y una nueva muestra será generada en dirección de cada uno de estos dos vecinos. Los puntos sintéticos son generados de la siguiente forma: Se toma la diferencia entre el vector de atributos perteneciente a la muestra bajo consideración y su vecino más cercano. Se multiplica la diferencia por un número aleatorio entre 0 y 1 y se suma el valor calculado al vector de atributos original. Esto causa la selección de un punto aleatorio sobre el segmento de línea entre dos características específicas. Esta aproximación forza la región de decisión de la clase minoritaria para que se convierta en una más general, convirtiendo la técnica en más útil que simplemente sobre-muestrear los datos positivos. Sin embargo, una de sus debilidades es que trabaja sobre el espacio de entrada y no sobre el de características que es donde optimiza la SVM, además la técnica se aplica sobre todos los elementos de la clase minoritaria y no incluye algún tipo de selección para trabajar sólo con los puntos más importantes.

Los nuevos puntos sintéticos causan que el clasificador genere regiones de decisión más grandes y menos específicas en lugar de regiones de decisión pequeñas

y más específicas. Entonces el entrenamiento es actualizado con regiones más generales para la clase minoritaria.

2.4. Técnicas de validación

2.4.1. Validación cruzada

La técnica de validación cruzada (*k-fold cross validation*) es usada en aprendizaje de máquinas con el objetivo de evitar el sobre-entrenamiento, que sucede cuando el clasificador aprende muy bien los patrones con los que entrenó y los clasifica correctamente, pero será incapaz de clasificar correctamente patrones diferentes.

En la validación cruzada se usa un subconjunto del conjunto de entrenamiento, para lo que es necesario dividir el conjunto de entrenamiento en k partes iguales, una vez dividido, el clasificador entrenará de forma iterativa con el conjunto formado por la unión de los subconjuntos diferentes a la iteración i, por lo que el conjunto de entrenamiento tendrá un tamaño igual a $k-1$, mientras que para la prueba se aplicará el subconjunto con índice igual a i. De esta forma si asignamos un valor $k=3$, el conjunto original se dividirá en 3 subconjuntos, entrenando la primera iteración con los subconjuntos $2,3$ y probando con el subconjunto 1, en la segunda iteración entrenará con $1,3$ y probará con el subconjunto 2 y finalmente en la tercera iteración el clasificador entrenará con $1,2$ y probará con el subconjunto 3. La precisión alcanzada finalmente será el promedio de las k pruebas.

2.5. Técnicas de validación de desempeño

2.5.1. Matriz de Confusión

El desempeño de los algoritmos para aprendizaje de máquinas son típicamente evaluadas por una matriz de confusión como se ilustra en la Tabla 2.1. Los clasificadores pueden producir una etiqueta de clase discreta, indicando la clase pronosticada para la instancia evaluada.

Dado un clasificador y una instancia, existen cuatro posibles resultados. Si la instancia originalmente tiene etiqueta positiva (+1) y es clasificada como positiva, entonces es contado como un verdadero positivo pero si esta misma instancia es clasificada como negativa (-1), entonces se cuenta como un falso negativo puesto que debería ser etiquetado como positivo.

Si la instancia originalmente es negativa y es clasificada como negativa, entonces es contada como un verdadero negativo; si esta misma instancia es clasificada como positiva, entonces se cuenta como un falso positivo.

Dado un clasificador y un conjunto de instancias (conjunto de prueba) una matriz de confusión de dos filas por dos columnas puede ser construida, representando la disposición del conjunto de instancias. Esta matriz forma la base de muchas métricas comunes. Las columnas pertenecen a la clase pronosticada mientras que las filas son la clase actual. Dentro de la matriz de confusión la variable TN (True Negatives) corresponde al número de muestras negativas correctamente clasificadas (Verdaderos Negativos), FP (False Positives) es el número de muestras negativas incorrectamente clasificadas como positivas (Falsos Positivos), FN (False Negatives) es el número de muestras positivas incorrectamente clasificadas como negativas (Falsos Negativos) y TP (True Positives) es el número de muestras

positivas correctamente clasificadas (Verdaderos Positivos).

Cuadro 2.1: Matriz de Confusión

		Clase Original	
		Positivo	Negativo
Clase Pronosticada	Positivo	Verdadero Positivo (TP)	Falso Positivo (FP)
	Negativo	Falso Negativo (FN)	Verdadero Negativo (TN)

La métrica pronosticada *accuracy* es una medida de desempeño generalmente asociada con algoritmos para aprendizaje de máquinas y es definida como:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{2.32}$$

La proporción de verdaderos positivos (TP_{rate}) de un clasificador es estimada usando la fórmula 2.33, instancias positivas correctamente clasificadas entre el total de muestras que son realmente positivas. También es conocida como *hit rate* o *recall*.

$$TP_{rate} = \frac{TP}{TP + FN} \tag{2.33}$$

La proporción de falsos positivos (FP_{rate}), también llamado proporción de falsas alarmas, es estimada a partir de la fórmula 2.34, instancias negativas incorrectamente clasificados entre total de muestras que son realmente negativas.

$$FP_{rate} = \frac{FP}{FP + TN} \tag{2.34}$$

2.5.2. *Receiver operating characteristics* (ROC)

Las gráficas ROC son gráficas con dos dimensiones en las cuales TP_{rate} es graficado en el eje Y y FP_{rate} es graficado en el eje X. Una gráfica ROC describe

trueques relativos entre beneficios (verdaderos positivos) y costos (falsos positivos). En la Figura 2.6 se muestra una gráfica ROC con cinco clasificadores discretos etiquetados de la A hasta la E. Un clasificador discreto es aquel donde las salidas son sólo una etiqueta de clase. Cada clasificador discreto produce un par (FP_{rate}, TP_{rate}) correspondiente a un sólo punto en el espacio ROC.

Es importante hacer notar varios puntos en el espacio ROC. El punto más bajo orientado a la izquierda $(0, 0)$ representa la estrategia de nunca emitir una clasificación positiva; este clasificador no comete errores del tipo *falso positivo* sin embargo, no obtiene *verdaderos positivos*. La estrategia opuesta, de emitir incondicionalmente clasificaciones positivas, es representado por el punto superior orientado a la derecha $(1, 1)$.

El punto $(0, 1)$ representa clasificación perfecta. En la Figura 2.6 el punto D tiene un desempeño perfecto.

Informalmente, un punto en el espacio ROC es mejor que otro si está ubicado al noroeste (TP_{rate} es más alto, FP_{rate} es más bajo, o ambos) respecto al primero. Los clasificadores aparecen del lado izquierdo de una gráfica ROC, cerca del eje X, puede parecer como "conservador": este tipo de clasificadores hacen predicciones positivas sólo con evidencia fuerte, de tal manera que crean pocos errores de tipo *falso positivo*, pero frecuentemente también tienen bajos TP_{rate}. Los clasificadores en el lado superior derecho de una gráfica ROC puede ser tomada como "liberal": estos clasificadores hacen predicciones positivas con evidencia débil por lo que clasifican cercanamente todas las muestras positivas correctamente, pero frecuentemente tienen altos FP_{rate}. En la Figura 2.6, A es más conservador que B.

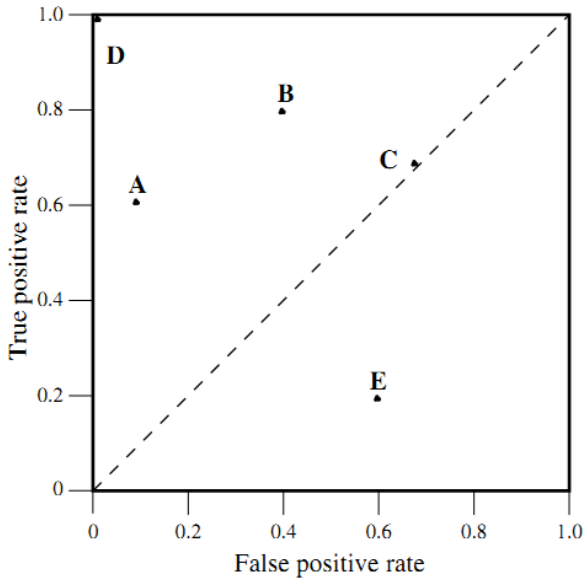


Figura 2.6: Gráfica ROC básica mostrando cinco clasificadores discretos

Área bajo la curva ROC (AUC)

La gráfica ROC muestra el desempeño de un clasificador de forma ilustrativa sin embargo, para comparar diferentes clasificadores es necesario reducir el desempeño ROC hasta un simple valor escalar que represente el desempeño esperado. Un método común es calcular el área bajo la curva ROC, abreviado como AUC. Debido a que el AUC es una proporción de un área de unidad cuadrada, este valor siempre permanecerá entre 0 y 1.0. Sin embargo, la diagonal entre los puntos (0,0) y (1,1) forma un área de 0.5, por lo cual ningún clasificador realista debería tener un AUC menor a 0.5.

El AUC tiene una propiedad estadística muy importante: el AUC de un clasificador es equivalente a la probabilidad de que este clasifique una instancia como positiva más que como una instancia negativa, elegida aleatoriamente. En la

práctica el AUC se desempeña bien y es usado a menudo cuando una medida de predicción es deseada (Fawcett, 2006).

2.5.3. Métricas de validación para conjuntos de datos no-balanceados

Para evaluar un clasificador sobre grandes conjuntos de datos altamente no-balanceados, es necesario utilizar una métrica adecuada. En conjuntos de datos con una distribución muy sesgada, la métrica de la precisión total no es suficiente. Esto es porque en un conjunto descompensado de 99 a 1, un clasificador que obtiene todos los datos negativos obtendrá una precisión del 99 %, pero será completamente inútil como clasificador para detectar los ejemplos positivos inusuales.

La comunidad médica y cada vez más la comunidad de aprendizaje de máquinas emplea dos métricas, la sensitividad y la especificidad para evaluar el desempeño de varias pruebas. La especificidad es calculada como:

$$Sn^{False} = \frac{T_N}{T_N + F_P} \tag{2.35}$$

donde Sn es la proporción de lugares candidatos en el conjunto de datos de prueba que han sido correctamente clasificados y este es evaluado como:

$$Sn = \frac{N_C}{N_T} \tag{2.36}$$

La sensitividad o Sn^{True} es la proporción de verdaderos positivos i.e.,

$$Sn^{True} = \frac{T_p}{T_P + F_N} \tag{2.37}$$

donde T_p es el número de patrones de Clase +1 reales los cuales son pronosticados como verdaderos (verdaderos positivos), T_N es el número de patrones de Clase -1 reales que son pronosticadas como falsos (verdaderos negativos), F_p es el número de patrones de clase -1 reales que fueron pronosticados como verdaderos (falsos positivos), F_N es el número de patrones de clase +1 reales que son pronosticados como falsos (falsos negativos), N_C es el número de datos positivos que han sido correctamente pronosticados en el conjunto de datos de prueba y N_T es el número total de patrones positivos en el conjunto de datos de prueba.

2.6. Algoritmos Genéticos

Cuando los problemas enfrentan un espacio de búsqueda extenso y los algoritmos existentes para resolver el problema de forma eficiente requieren tiempo exponencial, es necesario recurrir a las técnicas llamadas "heurísticas", puesto que las técnicas clásicas de búsqueda y optimización son insuficientes.

Las heurísticas son procesos que resuelven un problema en específico, pero que no ofrecen garantía de lograrlo. Una heurística es una técnica que busca soluciones buenas a un costo computacionalmente razonable, aunque sin garantizar factibilidad u optimalidad de las mismas. En algunos casos, ni siquiera es posible determinar que tan cerca está la solución encontrada respecto a la solución óptima.

Algunos ejemplos de técnicas heurísticas son *búsqueda Tabú*, *recocido simulado* y *escalando la colina*.

Dentro del espacio de búsqueda se encuentran dos zonas, la zona factible y la zona no factible, haciendo una solución factible, aquella solución válida para todas las restricciones del problema, aquella que se localiza dentro de la zona factible.

Los algoritmos genéticos (GA) usan heurísticas y técnicas estocásticas que permiten encontrar una solución factible en un tiempo razonable, sin la necesidad de información específica para guiar la búsqueda. Los GA están bioinspirados y toman como base el proceso de la evolución explicado por el Neo-Darwinismo que establece que la mayoría de la vida del planeta puede ser explicada a través de la reproducción, la mutación, la competencia y la selección.

El creador de los algoritmos genéticos fue John H. Holland, quien publicó este sistema en 1975 con el nombre de "planes reproductivos genéticos"; este sistema toma el proceso de adaptación de tal forma que los programas de una población interactúan y mejoran en base a un cierto ambiente que determina el comportamiento apropiado para estos individuos, que junto con variaciones aleatorias y un proceso de selección condujeron al desarrollo de un sistema adaptativo general.

Los GA fueron concebidos originalmente en el contexto de aprendizaje de máquinas, pero se ha usado principalmente en problemas de optimización, llegando a ser una de las técnicas más populares en la actualidad, se han usado en diferentes áreas, en optimización, para entrenar Redes Neuronales, en bases de datos para optimización de consultas, planeación de movimientos de robots, en economía y finanzas, etc.

2.6.1. Algoritmo Genético Simple

Los GA enfatizan la importancia de la cruza sexual (operador principal) sobre el de la mutación (operador secundario) usando una selección probabilística. El algoritmo básico es el siguiente:

1. Generar (aleatoriamente) una población inicial.

2. Calcular la aptitud de cada individuo.

3. Seleccionar padres (probabilísticamente) en base a la aptitud.

4. Aplicar operadores genéticos (cruza y mutación) para generar la siguiente población.

5. Ciclar hasta que cierta condición de paro se satisfaga.

Para poder aplicar el GA se requieren cinco componentes básicos:

1. Una representación para las soluciones potenciales del problema.

2. Una forma de crear una población inicial de posibles soluciones.

3. Una función de evaluación que tome el rol del ambiente, clasificando las soluciones en términos de su "aptitud".

4. Operadores genéticos que alteren la composición de los hijos que se producirán para las siguientes generaciones.

5. Valores para los diferentes parámetros que utiliza el GA (tamaño de la población, probabilidad de cruza, probabilidad de mutación, número máximo de generaciones, etc.).

2.6.2. Representación

Los GA están bioinspirados, por lo que es necesario definir algunos conceptos en el contexto de la computación evolutiva para poder entender cómo opera internamente. En los GA los individuos están representados a nivel genotipo como una cadena de bits o un arreglo de enteros, esta estructura es denominada

cromosoma. Se llama *gen* a una subsección de un cromosoma que codifica el valor de un solo parámetro. Se denomina *genotipo* a la codificación de los parámetros que representan una solución del problema a resolverse.

Se denomina individuo a un solo miembro de la población de soluciones potenciales a un problema, cada individuo contiene un cromosoma que representa una combinación de parámetros como solución posible al problema.

Se denomina generación a una iteración de la medida de aptitud y a la creación de una nueva población por medio de operadores de reproducción. Una población puede subdividirse en grupos llamados subpoblaciones, aunque normalmente, sólo pueden cruzarse entre sí los individuos que pertenecen a la misma subpoblación.

Para crear un GA se debe disponer de un mecanismo para codificar un individuo como un genotipo, eligiendo la más relevante para el problema en cuestión. A continuación se presentan dos de las codificaciones más usadas, la codificación binaria y la codificación de Gray.

Codificación binaria

La representación binaria es la tradicional para los GA, la cadena binaria es conocida como "cromosoma" cuya forma es $\langle b_m, ..., b_2, b_1 \rangle$, cada posición de la cadena se le denomina "gen" y el valor dentro de esta posición es llamado "alelo". Usando una representación binaria un alelo puede valer 0 ó 1.

De acuerdo con Holland, es preferible tener muchos genes con pocos alelos posibles que contar con pocos genes con muchos alelos posibles. Esto es sugerido no sólo por razones teóricas (teorema de los esquemas), sino que también tiene una justificación biológica, ya que es más usual encontrar cromosomas con muchas

posiciones y pocos alelos por posición que pocas posiciones y muchos alelos por posición.

La codificación binaria da pie a un grado más elevado de paralelismo implícito porque permite obtener un gran número de esquemas, denominando esquema a la plantilla que describe un subconjunto de cadenas que comparten ciertas similitudes en algunas posiciones a lo largo de su longitud.

El hecho de contar con más esquemas favorece la diversidad e incrementa la probabilidad de que se formen buenos bloques constructores en cada generación, lo que en consecuencia mejora el desempeño del GA con el paso del tiempo de acuerdo al teorema de los esquemas. Un bloque constructor es la porción de un cromosoma que le produce una aptitud elevada a la cadena en la cual está presente.

A pesar de la diversidad ofrecida por la representación binaria, se tiene una desventaja cuando se requiere optimizar una función con alta dimensionalidad junto con una buena precisión, ya que el mapeo de números reales a binarios generará cadenas extremadamente largas, resultando en un GA con bajo desempeño.

Codificación en GRAY

Un problema notable en la codificación binaria es el fenómeno conocido como risco de Hamming, presentado en los casos donde dos números adyacentes en el espacio de búsqueda, no son adyacentes en el espacio de representación (tienen una distancia). Esto es debido a que la codificación binaria no mapea adecuadamente el espacio de búsqueda con el espacio de representación. Para resolver este problema, en la literatura se ha planteado el uso de la codificación Gray que permite preservar la propiedad de adyacencia tanto en el espacio de búsqueda como en el de representación.

Para convertir cualquier número binario a un código de Gray hay que aplicar la operación lógica XOR, representada por el símbolo $\oplus$, sobre sus bits consecutivos de derecha a izquierda. Por ejemplo, dado el número 0101_2 en binario, aplicaríamos: $1 \oplus 0 = 1(bit_1)$, $0 \oplus 1 = 1(bit_2)$, $1 \oplus 0 = 1(bit_3)$ y el último bit de la izquierda permanece igual al no tener otro bit para aplicar el XOR, produciendo la cadena en código de Gray equivalente, $\langle b_4, b_3, b_2, b_1 \rangle = 0111$.

Existe una variedad de representaciones reportadas en la literatura, codificación de números reales, representaciones de longitud variable, representaciones en árbol, etc, que tratan de disminuir las limitaciones de la representación binaria.

Individuo a nivel Genotipo

El genotipo es la codificación básica de los individuos, es decir, el cromosoma. Las representaciones más utilizadas son codificaciones binarias, en GRAY y entera.

Individuo a nivel Fenotipo

Se denomina fenotipo a la decodificación del cromosoma, esto es, los valores obtenidos al pasar de la representación en genotipo (usualmente binaria) a la representación usada por la función objetivo.

2.6.3. Aptitud del Individuo

El valor que se asigna a cada individuo y que indica qué tan bueno es éste respecto a los demás como solución al problema, es lo que se denomina aptitud. Por ejemplo, si el individuo contiene un solo gen codificado binariamente como

1010_2, entonces al aplicar la función objetivo $f(x) = x^3$ con $x = 1010_2 = 10_{10}$, obtendríamos una aptitud de $f(10_{10}) = 1000_{10}$.

Dentro de un GA, la evaluación de la aptitud es el paso más costoso cuando se trata de resolver una aplicación real. Esta función deberá ser definida por el investigador y puede ser una subrutina o cualquier proceso externo que permita asignarle un valor a los individuos.

2.6.4. Técnicas de Selección

El proceso de selección de candidatos a reproducirse es una parte fundamental del funcionamiento de los GA, donde suele realizarse de forma probabilística, permitiendo que los individuos menos aptos tengan cierta oportunidad de sobrevivir.

Las técnicas de selección más usadas son: selección proporcional, selección mediante torneo y selección de estado uniforme.

Selección proporcional

Este grupo de técnicas de selección, originalmente propuestas por Holland, seleccionan los individuos de acuerdo a su contribución de aptitud respecto al total de la población, requiriendo de dos pasos: primero calcular la aptitud media y después calcular el valor esperado de cada individuo.

La Ruleta

Esta técnica fue propuesta por De Jong, es simple, pero ineficiente (su complejidad es $O(n^2)$. Asimismo, presenta el problema de que el individuo menos

apto puede ser seleccionado más de una vez.

El algoritmo es el siguiente:

1. Calcular la suma de valores esperados T.
2. Repetir N veces, donde N corresponde al tamaño de la población:
   a) Generar un número aleatorio r entre 0 y T.
   b) Ciclar a través de los individuos de la población sumando los valores esperados hasta que la suma sea mayor o igual a r.
   c) Se selecciona el individuo que haga que la suma exceda el límite.

Sobrante Estocástico

Técnica de selección propuesta por Booker y Brindle como una alternativa para aproximarse más a los valores esperados de los individuos.

La idea básica es asignar determinísticamente las partes enteras de los valores esperados para cada individuo y después usar un esquema proporcional para la parte fraccionaria. Esto reduce los problemas de la ruleta, pero puede ocasionar convergencia prematura al introducir una mayor presión de selección (producto de la asignación determinística de los valores esperados para cada individuo).

El algoritmo es el siguiente:

1. Asignar de forma determinística el conteo de valores esperados para cada individuo (valores enteros).
2. Los valores restantes (sobrantes del redondeo) se usan probabilísticamente para rellenar la población.

Hay 2 variantes principales:

1. Sin reemplazo: Cada sobrante se usa para sesgar el tiro de una moneda que determina si una cadena se selecciona de nuevo o no.

2. Con reemplazo: Los sobrantes se usan para dimensionar los segmentos de una ruleta y se usa esta técnica de manera tradicional.

Otras técnicas de selección proporcional son la Universal Estocástica, el Muestreo Determinístico, el Escalamiento Sigma, la selección por Jerarquías y la selección de Boltzmann.

Selección mediante torneo

La idea principal de este método es seleccionar con base en comparaciones directas de los individuos. Hay dos versiones, la determinística y la probabilística.

El algoritmo de la versión determinística es el siguiente:

1. Barajar los individuos de la población.

2. Escoger un número p de individuos (típicamente 2).

3. Compararlos con base en su aptitud.

4. El ganador del "torneo" es el individuo más apto.

5. Debe barajarse la población un total de p veces para seleccionar N padres (donde N es el tamaño de la población).

El algoritmo de la versión probabilística es idéntico al anterior, excepto por el paso en que se escoge al ganador. En vez de seleccionar siempre al individuo con aptitud más alta, se aplica la función $flip(p)$, que devuelve cierto (True) o falso (False) dependiendo de la probabilidad p. Si el resultado es cierto, se selecciona al más apto, de lo contrario, se selecciona al menos apto.

El valor de p permanece fijo a lo largo de todo el proceso evolutivo y se escoge dentro del siguiente rango: $0{,}5 < p \leq 1$.

Selección de estado uniforme

Esta técnica fue propuesta por Whitley y se usa en GA no generacionales, donde sólo unos cuantos individuos (los menos aptos) son reemplazados en cada generación. Esta técnica suele usarse cuando se evolucionan sistemas basados en reglas en los que el aprendizaje es incremental.

El algoritmo es el siguiente:

1. Nombrar G a la población original de un AG.

2. Seleccionar R individuos, donde $1 \leq R < M$, de entre los más aptos.

3. Efectuar la cruza y mutación a los R individuos seleccionados. Los nuevos hijos seran H.

4. Elegir al mejor individuo en H (o a los mejores).

5. Reemplazar los peores individuos de G por los mejores individuos de H.

2.6.5. Técnicas de Cruza

La cruza es un operador de reproducción que forma un nuevo cromosoma combinando partes de cada uno de sus cromosomas padres. En los GA la cruza es el operador principal.

Cruza de un punto

Ésta técnica fue propuesta por Holland y combina los cromosomas de dos padres para formar dos nuevos hijos a partir de un punto de cruza. En la Figura 2.7 se muestra un ejemplo de este tipo de cruza. El cromosoma del primer padre está con fondo negro mientras que el cromosoma del segundo padre está con fondo blanco. El punto de cruza divide los cromosomas en dos partes, la primera con 8 bits y la segunda con 12 bits, al combinarse, los descendientes de ejemplo quedan con una parte negra y una blanca, manteniendo el tamaño original de los cromosomas.

Ésta técnica fue muy popular en sus inicios pero ha caido en desuso debido a las desventajas que presenta, una de ellas es que destruye los esquemas en los que la longitud de definición (δ) es alta, siendo $\delta(H)$ la distancia entre la primera y última posición fija de un esquema H. El problema fundamental de la cruza de un punto es que presupone que los bloques constructores son esquemas cortos y de bajo orden, y cuando esto no sucede, suele no proporcionar resultados apropiados. Además de tratar preferencialmente algunas posiciones del cromosoma como los extremos de una cadena.

Cruza de dos puntos

La generalización de la cruza de un punto es la cruza de n puntos, siendo De Jong el primero en implementarlo.

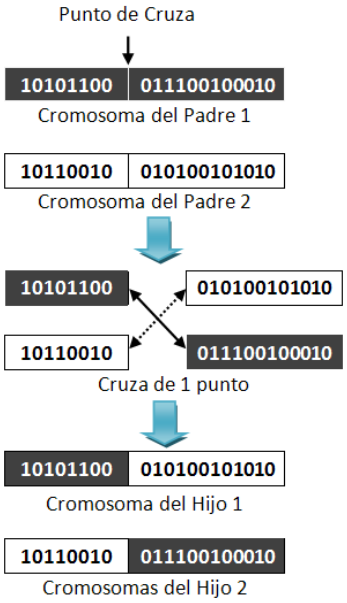


Figura 2.7: Cruza de 1 punto

La cruza con $n = 2$ es el que minimiza los efectos disruptivos de la cruza y de ahí que sea usado con gran frecuencia. En la Figura 2.8 se ejemplifica la cruza de dos puntos, donde los dos puntos de cruza dividen los cromosomas, de los padres de ejemplo, en tres partes, la primera de 6 bits, la segunda de 8 bits y la tercera de 6 bits. Después de aplicar la cruza de dos puntos los descendientes quedan con dos partes del cromosoma original y otra del segundo padre.

Cruza uniforme

Técnica propuesta por Ackey, donde la cruza tiene n puntos que no son fijos ni elegidos previamente. La cruza uniforme tiene un mayor efecto disruptivo que las dos cruzas anteriores, por lo que suele ajustarse la probabilidad de cruza con un valor cercano a 0.5 o menor.

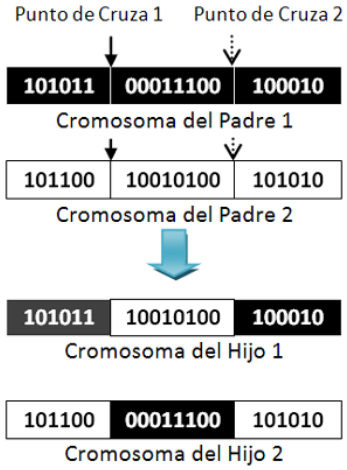


Figura 2.8: Cruza de 2 puntos

2.6.6. Mutación Uniforme

La mutación es un operador de reproducción, considerado en los GA como operador secundario, que forma un nuevo cromosoma a través de alteraciones (usualmente mínimas) de los valores de los genes de un solo cromosoma padre. En la práctica, se suelen recomendar porcentajes de mutación entre 0.001 y 0.01 para la representación binaria. Algunos investigadores, sin embargo, han sugerido que para favorecer el desempeño de los GA, el porcentaje de mutación deberá ser alto al inicio de la búsqueda y luego se deberá decrementar exponencialmente.

Para el caso de la mutación uniforme, dado un cromosoma $P = \langle b_m, ..., b_1 \rangle$, el gen mutado será el bit b'_k, donde $b'_k = 1$ si $flip(P_m) = TRUE$, de lo contrario, $b'_k = 0$, siendo P_m la probabilidad de mutación.

Entonces el individuo mutado será: $P' = \langle b_m, ..., b'_k, ..., b_1 \rangle$.

2.6.7. Reordenamiento

Otro tipo de operador de reproducción es el reordenamiento, donde se cambia el orden de los genes de un cromosoma, con el objetivo de juntar los genes que se encuentran relacionados, facilitando así la producción de bloques constructores.

La inversión es un ejemplo de un operador de reordenamiento, en el que se invierte el orden de los genes comprendidos entre dos puntos seleccionados al azar en el cromosoma.

2.6.8. GA elitista

Se denomina elitismo al mecanismo utilizado en algunos Algoritmos Evolutivos para asegurar que los cromosomas de los miembros más aptos de una población se pasen a la siguiente generación sin ser alterados por ningún operador genético. El elitismo asegura que la aptitud máxima de la población, entre cada generación, nunca se verá reducida. Sin embargo, no necesariamente mejora la posibilidad de localizar el óptimo global de una función.

En la literatura se ha reportado que el GA requiere de elitismo para poder converger al óptimo.

2.6.9. Paralelismo Implícito

El paralelismo implícito de los GA, demostrado por Holland, se refiere al hecho de que mientras el GA calcula las aptitudes de los N individuos en una población, al mismo tiempo estima de forma implícita las aptitudes promedio de un número mucho más alto de cadenas cromosómicas a través del cálculo de las aptitudes promedio observadas en los bloques constructores que se detectan en la población.

Se llama "bloque constructor" a un grupo pequeño y compacto de genes que han co-evolucionado de tal forma que su introducción en cualquier cromosoma, representa una alta probabilidad de incrementar la aptitud de dicho cromosoma.

La "decepción", por el contrario, es la condición donde la combinación de buenos bloques constructores llevan a una reducción de aptitud, en vez de un incremento.

2.6.10. Efecto de la Cruza y Mutación

En un Algoritmo Genético es importante definir una proporción entre la búsqueda de tipo exploratoria y la búsqueda de tipo explotativa. Cuando se atraviesa un espacio de búsqueda, se denomina explotación al proceso de usar la información obtenida de los puntos visitados previamente para determinar qué lugares resulta más conveniente visitar a continuación, en cambio la exploración son saltos a lo desconocido, es el proceso de visitar nuevas regiones del espacio de búsqueda para ver si se puede encontrar algo prometedor.

La exploración involucra grandes saltos, mientras que la explotación involucra movimientos finos. En el contexto de los GA, la explotación es lograda gracias al operador de cruza, permitiendo encontrar óptimos locales, mientras que la exploración, obtenida por el operador de mutación, evita que el GA quede atrapado en óptimos locales.

En el caso donde la mutación es cero, no hay alteración alguna en los cromosomas; cuando vale 1.0 creará siempre complementos del individuo original y si es 0.5, existirá una alta probabilidad de alterar drásticamente el esquema de un individuo.

Cuando se realiza la cruza, se mantienen los alelos que son comunes entre los dos padres, de manera que cuando la diversidad es baja dentro de la población,

el número de alelos se incrementará, permitiendo alcanzar una convergencia a un óptimo local.

Para lograr obtener un óptimo global, a veces la mutación puede ser más útil que la cruza, sin embargo hay una fuerte relación entre ambas.

2.6.11. *No Free Lunch Theorem*

Este famoso teorema fue formulado en un artículo escrito por David Wolpert y William MacReady, del Instituto Santa Fe, en Nuevo México (Wolpert and Macready, 1997). La principal implicación del *No Free Lunch Theorem* es que todas las técnicas de búsqueda heurística son matemáticamente equivalentes en general. Es decir, no hay una sola técnica que supere a las demás en todos los casos.

Capítulo 3

Metodología

En este Capítulo se presentan dos propuestas para mejorar el desempeño de las SVM sobre conjuntos no balanceados, presentando primero una vista general de la solución y ubicando el paso donde se puede aplicar uno de los dos métodos propuestos.

Las SVM permiten generalizar la clasificación al maximizar el espacio entre dos hiperplanos correspondientes a las clases positiva y negativa en conjuntos binarios; sin embargo, este hiperplano puede quedar sesgado cuando se trabaja con conjuntos no balanceados. Por otro lado cuando los conjuntos son linealmente no separables, como la mayoría de los problemas del mundo real, es necesario hacer uso de una función *kernel* que permita mapear los datos de entrada a un espacio n dimensional donde pueda realizarse la separación. Es en este espacio altamente dimensional llamado *espacio de características* donde se sugiere aplicar las técnicas para reducir el sesgo del hiperplano.

Para reducir el sesgo de clasificación en las SVM, se proponen los siguientes pasos metodológicos:

1. Preparar los conjuntos de entrenamiento y prueba e identificar la clase minoritaria y la clase mayoritaria.
2. Entrenar la SVM con el conjunto de entrenamiento.
3. Calcular la precisión usando el conjunto de prueba.
4. Mejorar la precisión modificando los parámetros de la SVM.
5. Obtener los SV.
6. Etiquetar los SV como SV^+ y SV^-.
7. Generar nuevos puntos artificiales en base a los SV^+.
8. Agregar los puntos artificiales a la clase minoritaria dentro del conjunto de entrenamiento.
9. Entrenar la SVM con el nuevo conjunto de entrenamiento.
10. Calcular la precisión usando el conjunto de prueba.
11. Mejorar la precisión modificando los parámetros de la SVM.
12. Si la precisión con los nuevos puntos es mejor que en la iteración anterior, se agregan los puntos artificiales al conjunto de entrenamiento.
13. En caso contrario el conjunto de entrenamiento queda igual.
14. Regresar al paso 5 hasta cumplir el criterio de paro.

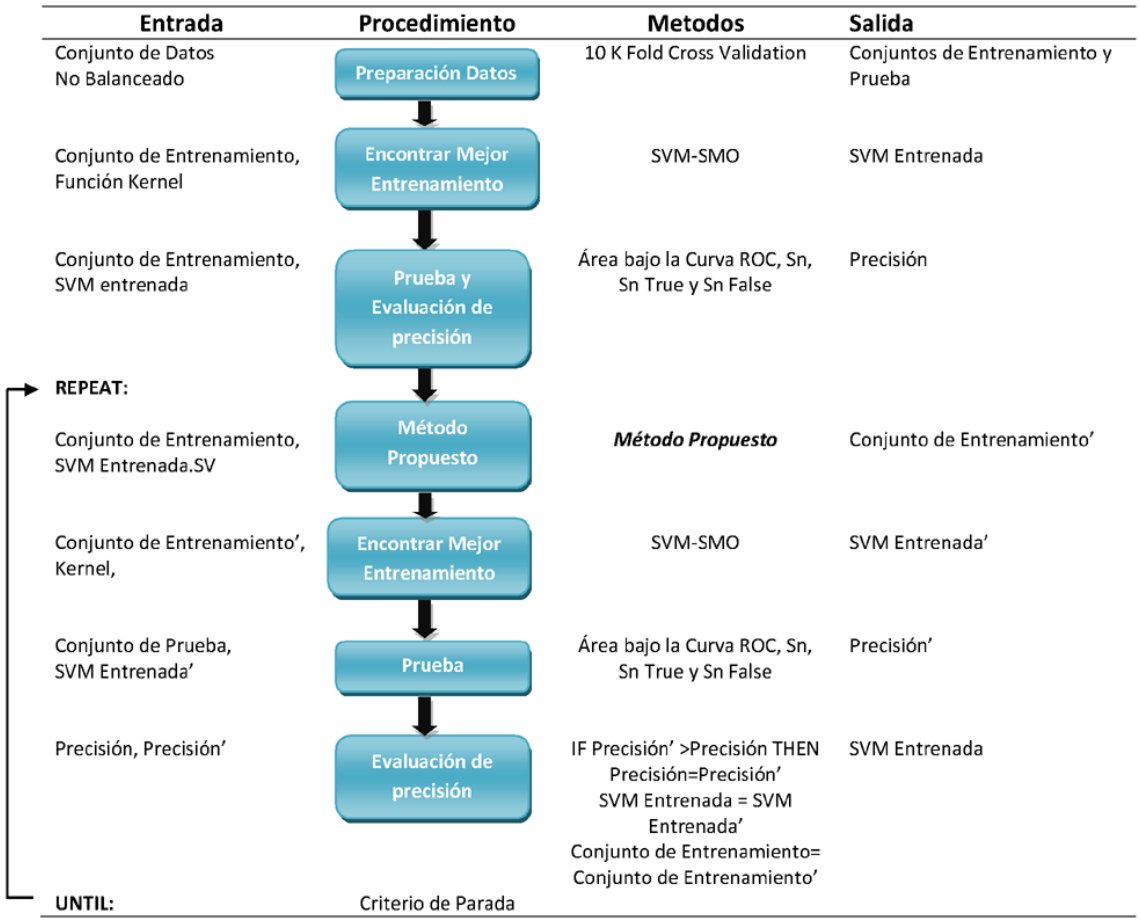


Figura 3.1: Algoritmo general para mejorar el desempeño en SVM

La Figura 3.1 muestra el flujo de información general para mejorar el desempeño de las SVM. El primer paso consiste en preparar los conjuntos de datos, uno para entrenamiento y otro para prueba usando la técnica de k *fold cross validation* con un valor de 10 para k, es decir, se divide el conjunto de entrada en 10 partes y en cada iteración se entrenará con 9 partes y la última se usará para prueba, cumpliéndose siempre que las muestras para el conjunto de prueba no pertenecen al conjunto de entrenamiento.

Los conjuntos de datos deberán tener etiquetadas las muestras, siendo X^+ las muestras positivas y X^- las muestras negativas. La clase minoritaria regularmente contiene las muestras positivas en proporción menor a la clase mayoritaria que contiene las muestras negativas.

Después de preparar los conjuntos de datos se entrena una SVM con el conjunto de entrenamiento, de este entrenamiento obtenemos un hiperplano preliminar $H_1(X^+, X^-)$ y a partir de este hiperplano obtenemos los Vectores Soporte (SV). En este paso es necesario realizar una búsqueda de los mejores parámetros para obtener una buena precisión, ya que los SV serán la base para crear los nuevos puntos artificiales.

Los SV son la mejor referencia ya que están ubicados sobre los hiperplanos de separación positivo y negativo que le permiten a la SVM distinguir las clases, además de que este reducido subconjunto de puntos representa los puntos más importantes del conjunto usado para el entrenamiento.

Una vez obtenidos los SV, en el sexto paso, es necesario etiquetarlos como SV^+ y SV^- de acuerdo a la clase a la que pertenezcan, estos SV serán la base para reducir el sesgo del hiperplano. Para lograr reducir este sesgo, en esta tesis se exponen dos algoritmos, el primero usa la excitación de SV y el segundo crea nuevos puntos artificiales, estos dos algoritmos serán discutidos con mayor profundidad en los siguientes apartados.

Finalmente después de reducir el desbalance entre clases, es necesario realizar una búsqueda de los mejores parámetros tanto para el algoritmo propuesto como para la SVM. Una buena combinación de parámetros reducirá el sesgo del hiperplano de separación de la SVM y mejorará la precisión de clasificación sin embargo, esta tarea requiere mucho tiempo de procesamiento, por lo que fue

necesario incluir primero una técnica de malla para la excitación de SV y después un algoritmo genético para la creación de puntos artificiales.

A continuación se presentan los algoritmos para mejorar el desempeño de las SVM sobre conjuntos de datos no balanceados. Primero usando la excitación de SV y después la creación de puntos artificiales.

3.1. Algoritmo 1. Mejora del desempeño de SVM mediante la excitación de vectores soporte

La excitación de vectores soporte se refiere a desplazar en alguna de sus dimensiones a los mismos SV; cuando el desplazamiento sea benéfico la SVM verá reducido el sesgo de su hiperplano de separación, en caso contrario, el sesgo será más acentuado.

Para iniciar con el algoritmo es necesario contar con una SVM ya entrenada que proveerá los SV para poder etiquetarlos como SV^+ y SV^- de acuerdo a la clase a la que pertenezcan.

El siguiente paso es excitar los SV^+, que pertenecen a la clase minoritaria, siendo necesario definir la dirección de la excitación. La dirección del movimiento se determina a través del vecino más cercano del conjunto de SV^- para cada SV^+ analizado, este vecino estará localizado sobre el hiperplano de separación negativo permitiendo que el movimiento sea dirigido hacia la frontera entre clases.

La excitación se realizará en cada dimensión del SV^+ y será aleatoria, estando definido este desplazamiento por la formula 3.1.

$$SV_{i,d}^{+'} = SV_{i,d}^{+} - \varepsilon(SV_{i,d}^{+} - SV_{N,d}^{-}) \tag{3.1}$$

donde $i = 1...tamaño(SV^+)$, $d = 1...tamaño(Dimensiones)$, $SV_N^- = SV^-$ más cercano y ϵ es un valor aleatorio entre $1x10^{-3}$ y $1x10^{-6}$.

En la Figura 3.2 se presenta una representación de la excitación de SV. La clase minoritaria está marcada con círculos de color azul mientras que la clase mayoritaria está marcada con cuadros de color verde. Los SV son los puntos marcados con el signo "+" y la dirección de la excitación para cada uno de los SV de la clase minoritaria esta marcada con flechas. Las nuevas posiciones para los SV^+ y el nuevo hiperplano de separación generado se muestran en la Figura 3.3.

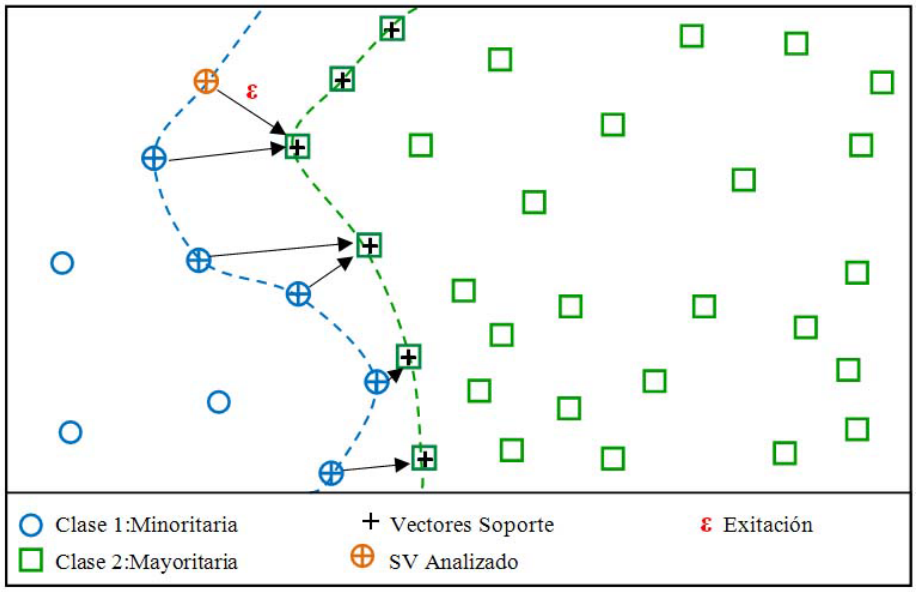


Figura 3.2: Excitación de SV para la clase minoritaria

Con el objetivo de reducir el efecto provocado por los conjuntos de datos no balanceados, la excitación sobre los SV^+ mueve el hiperplano de separación hacia la clase mayoritaria, reduciendo el sesgo.

La principal ventaja del método propuesto es que el desempeño de la SVM es

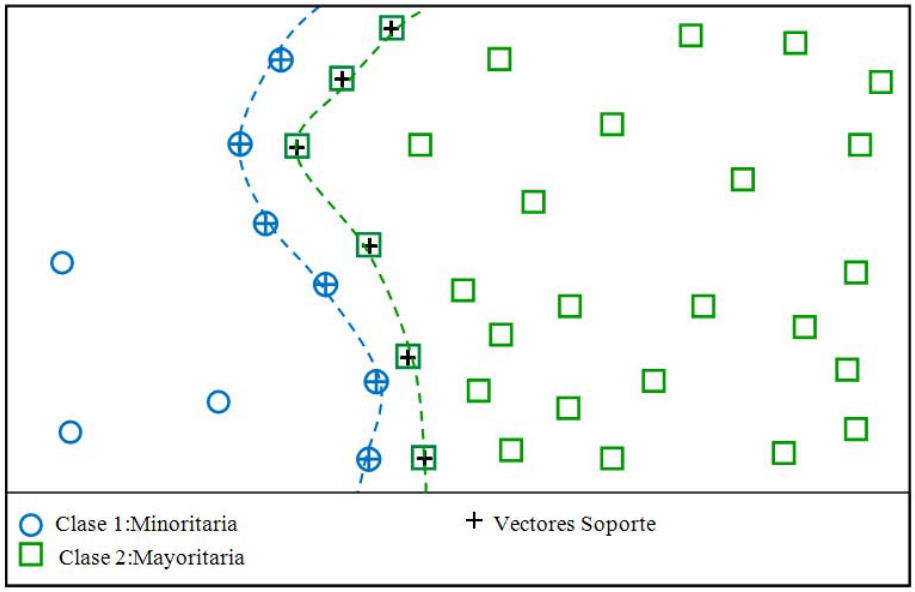


Figura 3.3: Nuevos SV e hiperplano de separación

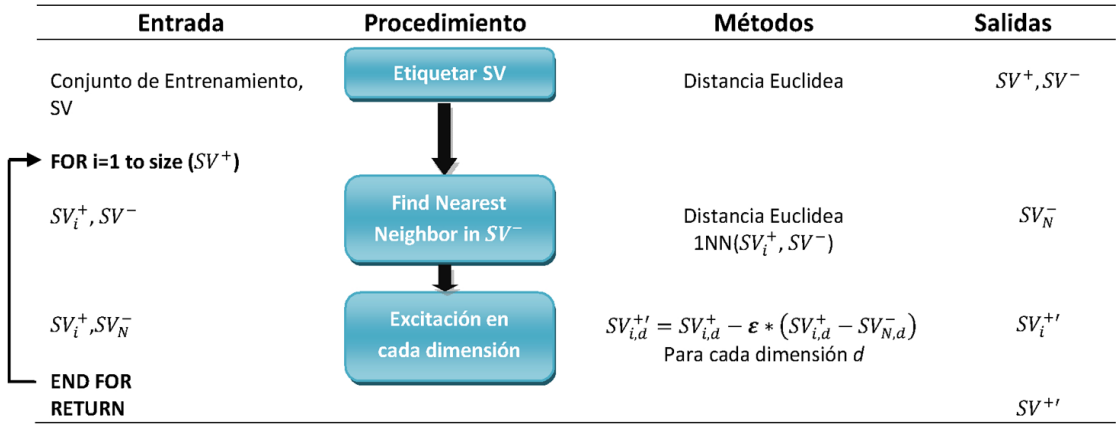


Figura 3.4: Algoritmo para la excitación de vectores soporte

mejorado gracias al desplazamiento de los SV^+ tomando como guía el hiperplano de separación máxima, que a diferencia de las técnicas clásicas, este desplazamiento ocurre dentro del espacio de características y es dirigido gracias a la distinción de SV^+ y SV^-.

La excitación en cada iteración es mínima para evitar bajar la precisión, pero al usar alguna técnica de búsqueda se pueden encontrar las excitaciones correctas para que la mejora sea significativa.

3.2. Algoritmo 2: Mejora del desempeño de SVM mediante la creación de puntos artificiales

El método propuesto tiene como objetivo reducir el desbalance entre clases mediante la creación de puntos artificiales que serán agregados a la clase minoritaria como nuevas muestras positivas dentro del conjunto de entrenamiento, esto permitirá reforzar el entrenamiento sobre la clase minoritaria y reducirá el sesgo del hiperplano de separación dentro de la SVM.

La creación de puntos artificiales no puede ser al azar, por lo que se propone usar los datos más importantes que están localizados sobre el hiperplano de separación (los SV^+), también se propone reducir el desbalance entre clases primero creando nuevas muestras dentro de la clase minoritaria y después usando los SV^- para crear puntos en la frontera entre clases.

3.2.1. Creación de puntos artificiales

El primer paso del método propuesto consiste en identificar la clase minoritaria y mayoritaria del conjunto de entrenamiento obtenido del conjunto no balanceado original. La clase minoritaria contiene t muestras positivos y son etiquetados como X_t^+, mientras que la clase mayoritaria contiene las muestras negativas etiquetadas como X_t^-. En caso de que el conjunto negativo sea grande, es recomendable aplicar una técnica de Bajo-muestreo para obtener un pequeño subconjunto y evitar un alto costo computacional.

El nuevo conjunto formado por X_t^+ y X_t^- es empleado posteriormente para entrenar una SVM con el objetivo de encontrar un hiperplano preliminar

$H_1(X_t^+, X_t^-)$ y es a partir de este hiperplano H_1 que obtenemos los vectores soporte SV's.

El segundo paso es etiquetar los SV de acuerdo a la clase a la que pertenezcan, obteniendo SV^+ y SV^-. Los SV son la mejor referencia ya que le permiten a la SVM crear el hiperplano de separación y distinguir las clases.

Una vez que se tienen etiquetados los SV, se utilizan los SV^+ como referencia para crear nuevos puntos artificiales primero dentro de la clase minoritaria y después en su frontera con la clase mayoritaria. Para poblar la clase minoritaria es necesario elegir un número k de puntos que se crearán por cada SV^+ y la cantidad de desplazamiento α que moverá cada SV^+ a una nueva posición. También hay que encontrar los k SV positivos más cercanos por cada SV^+ para finalmente obtener el nuevo punto aplicando la fórmula 3.2.

$$X_{ik}^{+'} = SV_i^+ + \alpha(SV_i^+ - SV_k^+) \tag{3.2}$$

$$\text{para cada } SV_k^+$$

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

$$X_{ik}^{+''} = SV_i^+ + \beta(SV_i^+ - SV_k^-) \tag{3.3}$$

$$\text{para cada } SV_k^-$$

En la Figura 3.5 puede apreciarse la creación de puntos artificiales dentro de la clase minoritaria. La clase minoritaria está marcada con círculos de color azul mientras

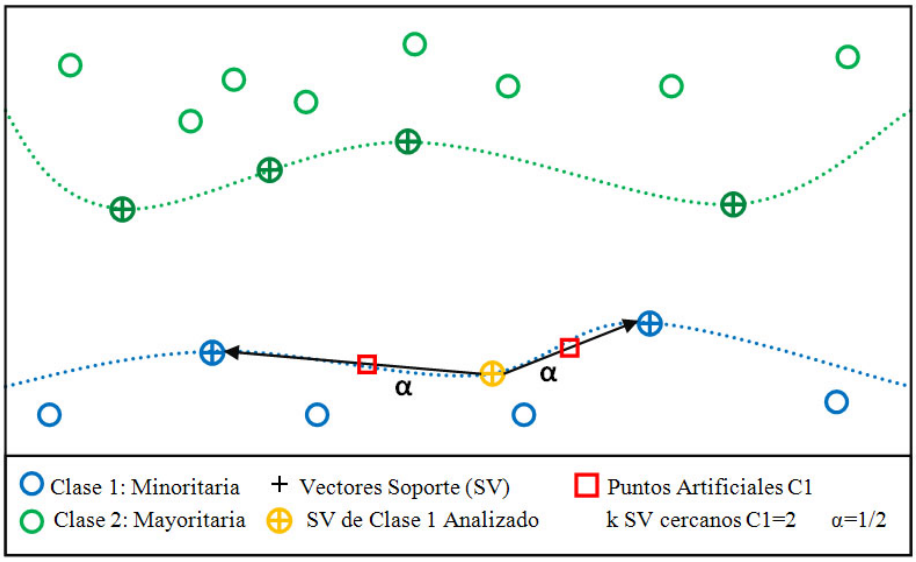


Figura 3.5: Creación de puntos artificiales dentro de la clase minoritaria

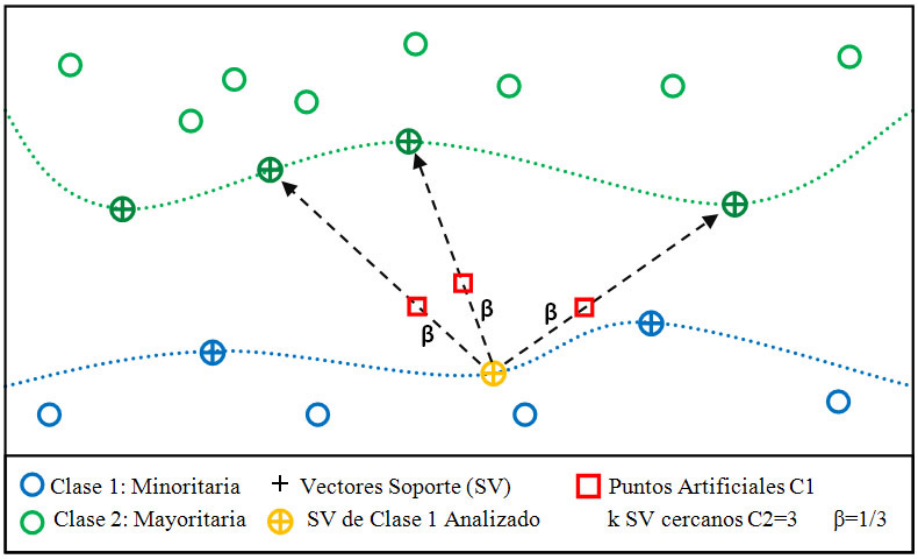


Figura 3.6: Creación de puntos artificiales en la frontera entre clases

que la clase mayoritaria se encuentra marcada con círculos de color verde. Los vectores soporte son los puntos marcados con "+" y que están sobre los hiperplanos de separación positivo y negativo. El punto analizado esta marcado con color amarillo, es un SV^+ y pertenece a la clase minoritaria; en la figura puede verse que a partir de este SV^+ se crean dos nuevos puntos artificiales marcados como cuadrados rojos, estos nuevos puntos fueron creados en base al SV analizado y desplazados una distancia $\alpha = \frac{1}{2}$ en dirección de sus dos SV más cercanos.En la

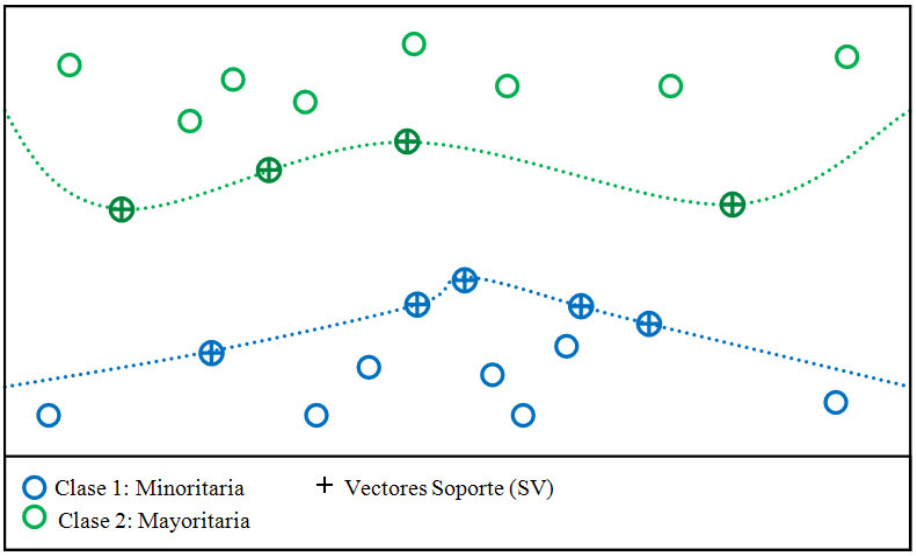


Figura 3.7: Nuevo conjunto de entrenamiento

Figura 3.6 se muestra la creación de puntos artificiales en la frontera entre clases, en ella se puede notar que se crean 3 nuevos puntos artificiales ya que se están tomando tres SV^- más cercanos. Hay que hacer notar que ahora son SV de la clase mayoritaria para asegurar que el desplazamiento β, que en este caso es de $\frac{1}{3}$, sea dirigido hacia la frontera entre clases y queden ubicados entre los SV^- vecinos y el SV^+ analizado. Finalmente en la Figura 3.7 se presenta el nuevo conjunto de entrenamiento con los nuevos hiperplanos y SV calculados, la clase minoritaria

muestra un incremento en el número de muestras positivas y la precisión mejora respecto al conjunto original.

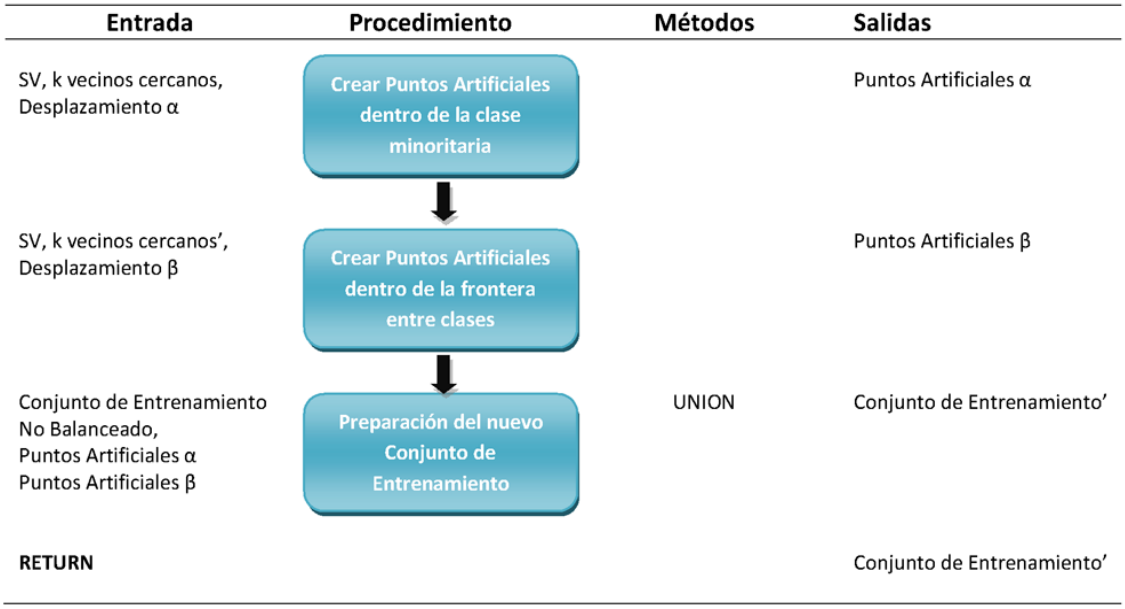


Figura 3.8: Algoritmo para la creación de puntos artificiales

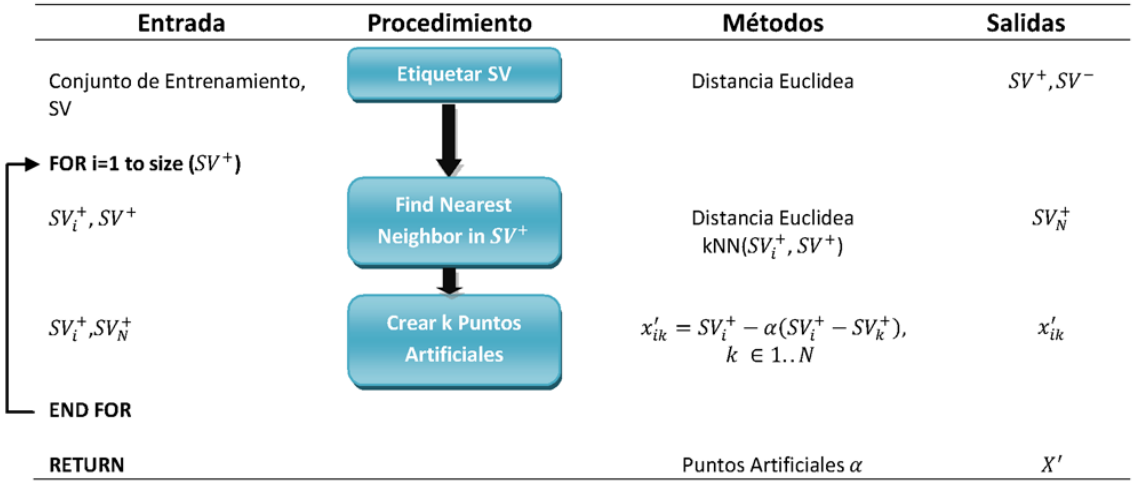


Figura 3.9: Algoritmo para la creación de puntos artificiales dentro de la clase minoritaria

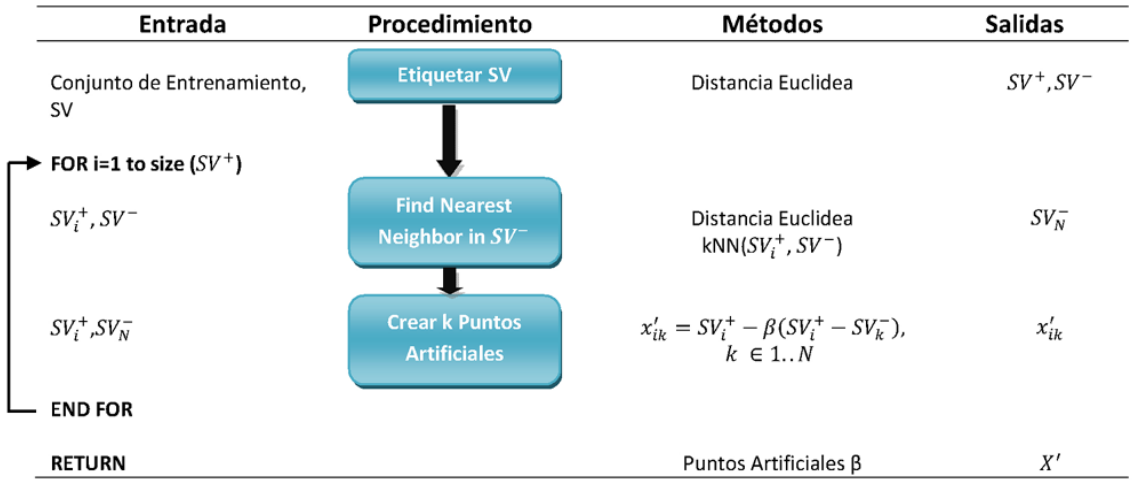


Figura 3.10: Algritmo para la creación de puntos artificiales dentro de la frontera entre clases.

3.3. Selección del Modelo

En este Capítulo se han expuesto dos nuevos métodos para reducir el sesgo de clasificación de las SVM sobre conjuntos no balanceados, sin embargo, tanto la SVM como los métodos propuestos dependen de una buena combinación de parámetros para obtener una buena precisión. Esto es llamado selección del modelo y es uno de los pasos más importantes ya que los parámetros seleccionados tienen un crucial efecto sobre el desempeño del clasificador entrenado.

La selección de un buen modelo permitirá obtener una buena habilidad de generalización, sin embargo, para cumplir este objetivo es necesario el ajuste de varios parámetros. A continuación se presentan los parámetros para las SVM.

El modelo de clasificación debe poder trabajar con conjuntos de datos no linealmente separables por lo que es necesario usar una función *kernel* que permita mapear el conjunto de entrada a un espacio altamente dimensional donde sea posible la separación entre clases.

El kernel con función de base radial (RBF), definida en la fórmula 3.4, permite realizar el mapeo y obtener una separación sobre conjuntos no linealmente separables a diferencia del kernel lineal.

$$K(X_i, X_j) = \exp(-\gamma||X_i - X_j||^2),\ \gamma > 0 \tag{3.4}$$

Una ventaja de usar kernel RBF es que sólo requiere un parámetro, gamma (γ), esta característica reduce la complejidad a diferencia de otros kernels como el polinomial, donde el número de hiperparámetros es mayor. Otra ventaja reside en el valor del kernel en la *ij-ésima* posición, donde está definida como $K_{ij} \leq 1$ a diferencia de los kernels polinomiales cuyos valores pueden tender al infinito o

pueden caer a cero cuando el grado polinomial es elevado.

Los métodos propuestos para reducir el sesgo de clasificación requieren de buenos vectores soporte por lo que se usará el algoritmo *Optimización Mínima Secuencial* (SMO), cuya ventaja reside en el corto tiempo de entrenamiento puesto que optimiza un subconjunto mínimo de únicamente dos puntos en cada iteración (Platt, 1998), esta característica será útil cuando el radio de desbalance del conjunto de datos sea alto.

En este punto, aunque el modelo ya tiene definido el tipo de kernel y la forma de calcular los SV, hay parámetros que no pueden ser estáticos, estas variables usada por los métodos propuestos deben ser obtenidas a través de una búsqueda. Para el primer algoritmo se utilizó una búsqueda en malla ya que los parámetros a optimizar son pocos, mientras que para el segundo algoritmo se requerian optimizar siete parámetros, por lo que usar una malla implicaría un alto costo computacional. La solución fue utilizar un algoritmo genético para obtener buenos parámetros en un tiempo razonablemente corto. En las siguientes subsecciones se exponen los dos tipos de búsqueda usados.

3.3.1. Mejora de parámetros usando una técnica de malla

La excitación de los SV depende de buenos parámetros para poder mejorar el desempeño de las SVM y aunque la dirección del desplazamiento es guiado, la proporción ε es un valor aleatorio entre 1×10^{-3} y 1×10^{-6} por lo que en ciertas ocasiones los resultados no serán buenos, requiriendo una búsqueda de buenos parámetros.

Para abordar este problema se utilizó una técnica de Malla, donde se realiza primero una exploración dentro del espacio de búsqueda y después una explotación

del lugar con soluciones potenciales. La técnica de malla analiza n combinaciones dentro de un rango, donde cada combinación tiene un incremento Δ a partir del punto inicial, siendo esta búsqueda una exploración. Una vez evaluadas todas las soluciones dentro del rango, se elige un subrango donde se encuentra la mejor solución y se realiza una búsqueda más fina, requiriendo un incremento Δ más pequeño, esta búsqueda es conocida como explotación. Las búsquedas de exploración y explotación se realizan por nivel, donde cada nivel analiza un rango para cada dimensión de la combinación, es decir, si la solución s contiene dos dimensiones (s_1, s_2), se creará una malla con dos niveles, pero si s tiene muchas dimensiones, el tiempo de la búsqueda incrementará significativamente.

Para la excitación de SV se requirió una malla con los parámetros gamma (γ) para el kernel RBF en un rango [10^{-2}, 10^{-1}, 10^{0}, 10^{1}] y el parámetro de regulación C con el rango [10^{0}, 10^{1}, 10^{2}, 10^{3}, 10^{4}]. El incremento es pequeño pero a través de la malla se pueden obtener los desplazamientos adecuados para mejorar la precisión del clasificador.

3.3.2. Mejora de parámetros usando un algoritmo genético

La determinación de los mejores parámetros tanto para la SVM como para la creación de puntos artificiales, es un paso importante ya que es uno de los factores de los que depende la precisión del clasificador, sin embargo, esta tarea se complica debido a la cantidad de parámetros a determinar y la precisión decimal para cada uno, creando un espacio de búsqueda realmente grande y donde una técnica de malla seria permisiva debido al alto tiempo de procesamiento que requeriría.

El algoritmo genético tiene la propiedad de encontrar 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, es esta ventaja la que se espera reflejar en el método propuesto, por lo que se usará un GA para obtener una buena combinación de parámetros y poder crear puntos artificiales de forma eficiente.

Los GA son una analogía presente en la naturaleza sobre la reproducción de individuos a nivel genotipo, donde dos padres comparten su información genética a través del proceso de cruza para crear un nuevo individuo con rasgos similares o con rasgos nuevos si el operador de mutación modifica algún gen, cada generación proveerá de un conjunto de soluciones factibles y después de varias generaciones los mejores genes estarán en todos los individuos, convergiendo a una solución. En los organismos vivos podemos encontrar cuatro diferentes tipos de pares base (Adenina, Timina, Guanina y Citocina) para representar la información genética, sin embargo, en computación los datos se representan a nivel de bits por lo que los cromosomas para cada individuo se expresaran usando una cadena de ceros y unos.

En el modelo propuesto, los cromosomas usados contendrán los parámetros de la SVM tales como tipo de kernel, costo C, *gamma* para el kernel RBF, así como

los parámetros del método propuesto, SV_k^+ que corresponde a la cantidad de SV positivos más cercanos al SV^+ analizado, SV_k^- que corresponde a la cantidad de SV negativos más cercanos al SV^+ analizado y los valores normalizados entre 0,01 y 0,90 para las variables α y β que definen en desplazamiento del SV^+ analizado dentro de la clase minoritaria y en la frontera entre clases respectivamente. En la Tabla 3.1 se especifican las variables a guardar por el cromosoma, así como el tamaño en bits ocupado por cada variable, su rango, precisión decimal y el tamaño que ocuparán medido en bits.

Cuadro 3.1: Tamaño en bits para cada variable del cromosoma

Variable	Rango	Precisión decimal	Tamaño en Bits
γ (Gamma)	[0.001, 1.000]	3	10
C	[0.001, 1.000]	3	10
k (Tipo de Kernel)	{1-lineal, 2-RBF}	0	1
SV_k^+	{0,1,2}	0	2
α	[0.01, 0.90]	2	7
SV_k^-	{0,1}	0	1
β	[0.01, 0.90]	2	7

El espacio de búsqueda para el GA incrementa mientras la variable requiera más precisión, por lo que debe cuidarse el balance entre la precisión decimal de las variables y su tamaño de representación en bits. Como puede verse en la Tabla 3.1, la precisión decimal se definió en 3 para las variables γ y C ya que requieren de buena precisión, las variables de desplazamiento α y β requieren poca precisión por lo que se les asignó dos posiciones decimales mientras que las variables k, SV_k^+ y SV_k^- son de tipo entero así que no requieren precisión decimal.

El rango es un parámetro limitado por la precisión decimal, por lo que en el caso de las variables enteras, como el tipo de kernel, sólo puede ser kernel lineal, asignándole el valor 1(no utiliza la variable gamma), o puede ser kernel con Función

de Base Radial, asignándole el valor 2. En el caso de las variables para los k SV más cercanos, se restringió como posibles valores el 0, 1 y 2 para los SV_k^+, mientras que los valores posibles para los SV_k^- sólo pueden ser 0 ó 1, por lo que la creación de puntos artificiales quedó restringida con un máximo de dos nuevos puntos dentro de la clase minoritaria por cada SV^+ y de un sólo nuevo punto dentro de la frontera entre clases por cada SV^+. Opcionalmente puede no crearse puntos artificiales, que ocurre cuando la variable k SV se vuelve cero, esto para casos donde sea mejor poblar sólo la clase minoritaria o donde con sólo mejorar los parámetros de la SVM sea suficiente para mejorar la precisión de clasificación.

La variable *gamma* es propia del kernel RBF y aunque está normalizada, al igual que el parámetro de regulación C, no puede valer 0, por lo que sus rangos se definieron entre [0,001, 1,000].

Los parámetros α y β también requieren precisión decimal sin embargo, es necesario asegurar que el desplazamiento no cree el mismo punto, que ocurre cuando el desplazamiento es 0 %, o que quede más allá de la frontera entre clases, es decir dentro de la clase negativa, que sucede cuando el desplazamiento es mayor o igual al 100 %. Esta última circunstancia crearía ruido al tener un punto artificial con etiqueta positiva y otro con etiqueta negativa ubicados en el mismo lugar; para evitar esto, los rangos para estas variables se definieron entre 0,01 y 0,90, siendo un punto muy cercano al SV^+ analizado cuando el desplazamiento es 0 % y un punto relativamente cercano al SV_k^- cuando el desplazamiento es del 90 %.

Una vez definida la precisión decimal y el rango para cada variable, es necesario determinar su tamaño usando la fórmula 3.5, donde n es la precisión deseada y $nbits$ es la cantidad de bits necesaria para almacenar la variable dentro del cromosoma.

$$nbits = \log 2[(LímiteSuperior - LímiteInferior) \times 10^n] + 0{,}5 \qquad (3.5)$$

En la Tabla 3.2 se muestra un ejemplo de la formación de un cromosoma para un individuo a partir de las especificaciones de la Tabla 3.1. El tamaño del cromosoma final es de 38 bits, sumando el tamaño individual de cada variable.

Cuadro 3.2: Individuo de ejemplo a nivel genotipo para el conjunto Four class

Individuo: Nivel genotipo

γ	C	K	SV_k^+	α	SV_k^-	β
0001010010	0000110010	1	01	1100101	0	1110111

Para obtener los valores correspondientes al fenotipo de cada variable, es necesario aplicar un mapeo de números binarios hacia números reales usando la fórmula 3.6, donde $nbits$ es el tamaño de la cadena calculada anteriormente, $x\prime$ es el valor obtenido de la conversión simple a decimal de la cadena binaria y x es el valor en número real correspondiente a la combinación obtenida de $x\prime$.

$$x = LímiteInferior + x' \frac{(LímiteSuperior - LímiteInferior)}{2^{nbits} - 1} \tag{3.6}$$

En la Tabla 3.3 se muestran los valores reales obtenidos de la conversión de binario a decimal y luego aplicando el mapeo hacia números reales. Estos valores corresponden al fenotipo del cromosoma de ejemplo y como combinación de parámetros representan una solución potencial al problema de balanceo de clases a través de la creación de puntos artificiales.

Cuadro 3.3: Individuo de ejemplo a nivel fenotipo para el conjunto Four class

Individuo: Nivel fenotipo

γ	C	K	SV_k^+	α	SV_k^-	β
0.098	0.035	2	1	0.50	0	0.64

Los valores del fenotipo son una combinación de parámetros del método propuesto, por lo que para el individuo de ejemplo, se usará una SVM con $\gamma = 0{,}098$,

$C = 0{,}035$ y un kernel con función de base radial. Después de entrenar la SVM con los parámetros anteriores se obtienen los SV y se etiquetan en positivos y negativos de acuerdo a la clase a la que pertenezcan. Para la creación de puntos artificiales se usan los SV^+ y para cada SV_i^+ se buscan primero sus k SV^+ más cercanos, en este caso k vale 1, y el nuevo punto se localizará a la mitad de la distancia entre los puntos SV_i^+ y SV_k^+ ya que α vale 0,5; esta operación creará un punto artificial dentro de la clase minoritaria por cada SV^+ analizado.

Por último para crear puntos en la frontera entre clases, se buscan los SV_k^- más cercanos a SV_i^+ y el nuevo punto se localizará a una distancia β de la distancia entre el SV_i^+ y sus SV negativos más cercanos, sin embargo, para este caso debido a que SV_k^- vale 0, no se crearán puntos en la frontera entre clases.

El Algoritmo Genético requiere de una función aptitud para evaluar que tan bueno es un individuo como solución al problema respecto al resto de la población, para el método propuesto se definió como combinación de cuatro métricas, área bajo la curva (AUC), Sn, proporción de verdaderos positivos (Sn^{True}) y proporción de falsos positivos (Sn^{False}) y no a partir de una sola métrica. En la Tabla 3.4 se muestra la precisión alcanzada por la SV después de agregarle los puntos artificiales positivos a su entrenamiento, en este caso el individuo de ejemplo tenía una buena combinación en su cromosomas permitiéndole obtener una muy buena precisión sobre el conjunto Four class, de 100 % en todas las métricas, por lo que la SVM ya no tiene sesgo en su clasificación.

Cuadro 3.4: Precisiones de clasificación alcanzadas por el Individuo de ejemplo para el conjunto Four class

Aptitud: Precisión

AUC	Sn	Sn^{True}	Sn^{False}
1.000	1.000	1.000	1.000

Para la obtención de buenos parámetros en un tiempo razonable se usa un Algoritmo Genético. En base al tamaño calculado en bits para cada variable, como se reporta en la Tabla 3.1, se fijó el tamaño del cromosoma en 38 bits, con una población de 24 individuos.

El tipo de selección de padres usado dentro del GA es el Sobrante Estocástico con reemplazo debido a sus bondades, ya que este tipo de selección permite elegir aquellos individuos que no tienen una buena aptitud, reflejándose en una búsqueda de tipo explorativa, evitando converger a un mínimo local rápidamente, mientras que la ruleta que forma con los sobrantes de aptitud, le permite seleccionar a los mejores individuos, reflejándose en una búsqueda de tipo explotativa. La rápida convergencia del sobrante estocástico con reemplazo se debe al segunda paso, la generación de la ruleta con los sobrantes, que introduce una mayor presión en el proceso de selección, con lo que se puede dar una convergencia prematura. La complejidad del sobrante estocástico con reemplazo es de $O(n^2)$ debido a sus dos selecciones, la primera determinística y la segunda estocástica en base a la ruleta.

Los operadores usados son cruza de dos puntos y mutación uniforme con una probabilidad fija de 0.25. La cruza de dos puntos fue elegida porque minimiza los efectos disruptivos de la cruza frente a técnicas como cruza de un punto y cruza uniforme.

El número de generaciones máximas se estableció en 12 ya que el GA convergía rápidamente. Para asegurar esta convergencia, se aplicó el enfoque elitista para mantener intacto el material genético del mejor individuo en la siguiente generación. Los parámetros de la SVM convergen rápidamente y se elige el mejor individuo de dos corridas. Adicionalmente se utilizó una codificación en GRAY para disminuir las debilidades de la cruza de dos puntos y las debilidades de la representación binaria.

Capítulo 4

Resultados y Análisis Experimental

La metodología y los resultados obtenidos inicialmente con los algoritmos propuestos ya han sido publicados en diferentes revistas (ver la Sección de publicaciones para mayor detalle), sin embargo, para poder comparar el desempeño de ambos fue necesario preparar pruebas con las mismas condiciones. En este Capítulo se presentan los resultados obtenidos con los algoritmos propuestos, usando 21 conjuntos de datos no balanceados. La organización de las Secciones es la siguiente: primero se presenta una descripción de los conjuntos de datos, después los resultados usando las técnicas clásicas (Bajo-muestreo, Sobre-muestreo y SMOTE), después los resultados usando el primer algoritmo propuesto (la excitación de SV) y finalmente los resultados obtenidos del segundo algoritmo propuesto (la creación de puntos artificiales y optimización de parámetros por GA).

4.1. Conjuntos de datos

Los conjuntos de datos utilizados para las pruebas son conocidos como KEEL Datasets y son conjuntos comúnmente empleados para evaluar el desempeño de clasificadores sobre conjuntos de datos no balanceados. Los conjuntos de datos empleados están disponibles en *http://sci2s.ugr.es/keel/datasets.php,* son conjuntos binarios (con dos clases) y presentan un desbalance entre sus clases. En la Tabla 4.1 se muestran las características de los 21 conjuntos, ordenados de acuerdo a su radio de desbalance que va desde 1:1.4 para el conjunto *liver disorders* hasta 1:41.4 para *yeast 6*, que puede interpretarse como 1 muestra positiva por cada 41.4 muestras negativas para este último conjunto.

A continuación se ofrece una breve descripción de cada conjunto, esta información fue extraida de las publicaciones de los autores originales, para obtener más información acerca de los conjuntos y sus atributos referirse a la página antes citada para KEEL Datasets.

Cuadro 4.1: Relación de desbalance entre clases para cada conjunto de datos

Relación de desbalance entre clases				
Conjunto de Datos	Clase Minoritaria(+1)	Clase Mayoritaria(-1)	Características	Radio de Desbalance
liver disorders	145	200	6	1:01.379
four class	307	555	2	1:01.808
glass 1	76	138	9	1:01.816
diabetes	268	500	8	1:01.866
glass 0	70	144	9	1:02.057
vehicle 2	218	628	18	1:02.881
vehicle 3	212	634	18	1:02.991
ecoli 1	77	259	7	1:03.364
ecoli 2	52	284	7	1:05.462
glass 6	29	185	9	1:06.379
yeast 3	163	1321	8	1:08.104
ecoli 3	35	301	7	1:08.600
glass 2	17	197	9	1:11.588
cleveland 0 vs 4	13	164	13	1:12.615
glass 4	13	201	9	1:15.462
ecoli 4	20	316	7	1:15.800
pageblocks-1-3vs4	28	444	10	1:15.857
glass 5	9	205	9	1:22.778
yeast 4	51	1433	8	1:28.098
yeast 5	44	1440	8	1:32.727
yeast 6	35	1449	8	1:41.400

4.1.1. Conjunto Cleveland

Este subconjunto de datos es una parte del conjunto de personas con enfermedad del corazón, fue obtenido del Centro Médico V.A, Long Beach y la Fundación Clínica de Cleveland. Cuenta con catorce atributos por cada muestra y se ha usado para detectar la presencia de enfermedad del corazón en los pacientes. El valor asignado para la clasificación es un entero que va de 0 para la ausencia de la enfermedad hasta un valor de 4.

Cuadro 4.2: Atributos del conjunto Cleveland

Atributo	Tipo	Rango
Age	real	[29.0, 77.0]
Sex	real	[0.0, 1.0]
Cp	real	[1.0, 4.0]
Trestbps	real	[94.0, 200.0]
Chol	real	[126.0, 564.0]
Fbs	real	[0.0, 1.0]
Restecg	real	[0.0, 2.0]
Thalach	real	[71.0, 202.0]
Exang	real	[0.0, 1.0]
Oldpeak	real	[0.0, 6.2]
Slope	real	[1.0, 3.0]
Ca	real	[0.0, 3.0]
Thal	real	[3.0, 7.0]

Clases: {0,1,2,3,4}

El conjunto *Cleveland 0 vs 4* usado para las pruebas en esta Tesis es una versión no balanceada del conjunto original *Cleveland*, donde las muestras positivas pertenecen a la clase 0 y las muestras negativas pertenecen a la clase 4.

4.1.2. Conjunto Diabetes

El conjunto de datos diabetes proviene de una investigación en medicina realizada en 1994 por parte de la Universidad de Washington. Las muestras de los pacientes con diabetes fueron obtenidas de dos fuentes: a través de registros electrónicos automáticos y registros en papel. El dispositivo automático tiene un reloj interno para registrar los eventos, mientras que los registros en papel sólo proporcionan el tiempo lógico dividido en desayuno (8:00), almuerzo (12:00), cena (18:00) y la hora de dormir (22:00). Debido a esto, los registros electrónicos tienen

información más realista respecto al tiempo.

El conjunto consiste de cuatro campos por muestra.

Cuadro 4.3: Atributos del conjunto Diabetes

Atributo	Descripción
Date	Fecha en formato MM-DD-YYYY
Time	Tiempo en formato XX:YY
Code	código
Value	valor

El atributo código es descifrado de acuerdo a la siguiente lista, iniciando en 33:

33 : Dosis regular de insulina.

34 : Dosis de insulina NPH.

35 : Dosis de insulina Ultra Lente.

48,57 : Medida no especificada de glucosa en la sangre.

58 : Medida de glucosa en la sangre antes del desayuno.

59 : Medida de glucosa en la sangre después del desayuno.

60 : Medida de glucosa en la sangre antes del almuerzo.

61 : Medida de glucosa en la sangre después del almuerzo.

62 : Medida de glucosa en la sangre antes de la cena.

63 : Medida de glucosa en la sangre después de la cena.

64 : Medida de glucosa en la sangre antes de un bocado.

65 : Síntomas de Hypoglycemia.

66 : Ingestión típica de comida.

67 : Ingestión más de lo usual de comida.

68 : Ingestión menos de lo usual de comida.

69 : Actividad típica de ejercicio.

70 : Actividad más de lo usual de ejercicio.

71 : Actividad menos de lo usual de ejercicio.

72 : Evento no especificado.

4.1.3. Conjunto Ecoli

El conjunto Ecoli recopila información del lugar donde se localizan las proteínas dentro de la bacteria Escherichia coli. El autor es Kenta Nakai, del instituto de Biología Molecular de la Universidad de Osaka en Japón. Este investigador propone una clasificación en función del lugar donde se localizan las proteínas a través de algunas medidas concernientes a la célula.

Las clases, que corresponden al lugar dentro de la célula, son: citoplasma (cp) con 143 muestras, membrana interna sin señal de secuencia (im) con 77 muestras, membrana interna con señal de secuencia divisible (imS) con 2 muestras, membrana interna lipoproteica (imL) con 2 muestras, membrana interna con señal de secuencia indivisible (imU) con 35 muestras, membrana externa (om) con 20 muestras, membrana externa lipoproteica (omL) con 5 muestras y periplasma (pp) con 52 muestras.

A continuación se presenta, en la Tabla 4.4, una breve descripción, el tipo de dato y el rango para cada uno de los atributos del conjunto.

Clases: Site {cp, im, imS, imL, imU, om, omL, pp}

Los siguientes conjuntos usados para las pruebas en esta Tesis, son una versión reducida del conjunto Ecoli.

Cuadro 4.4: Atributos del conjunto Ecoli

Atributo	Tipo	Rango	Descripción
Mcg	real	[0.0,89.0]	Valor obtenido en el método McGeoch para el reconocimiento de la señal de secuencia.
Gvh	real	[1.0,88.0]	Valor obtenido en el método von Heijne para el reconocimiento de la señal de secuencia.
Lip	real	[1.0,48.0]	Puntuación obtenida en el consenso de la Señal Peptidasa II de Von Heijne.
Chg	real	[1.0,5.0]	Cantidad de carga en la predicción de lipoproteínas.
Aac	real	[0.0,88.0]	Puntuación del análisis discriminante de contenido de aminoácidos en la membrana exterior y en las proteínas periplásmicas.
Alm1	real	[1.0,94.0]	Puntuación obtenida en la predicción de la magnitud para la abertura de la membrana ALOM.
Alm2	real	[0.0,99.0]	Representa la puntuación del atributo anterior excluyendo las señales susceptibles de división.

ecoli1: Es una versión no balanceada del conjunto original Ecoli, donde las muestras positivas pertenecen a la clase *im* y las muestras negativas pertenecen al resto.

ecoli2: Es una versión no balanceada del conjunto original Ecoli, donde las muestras positivas pertenecen a la clase *pp* y las muestras negativas pertenecen al resto.

ecoli 3: Es una versión no balanceada del conjunto original Ecoli, donde las muestras positivas pertenecen a la clase *imU* y las muestras negativas pertenecen al resto.

ecoli4: Es una versión no balanceada del conjunto original Ecoli, donde las muestras positivas pertenecen a la clase *om* y las muestras negativas pertenecen al resto.

4.1.4. Conjunto Four class

Este conjunto de datos proviene de la treceava conferencia internacional sobre Reconocimiento de Patrones celebrada en 1996 en Viena, Austria. Los autores son Tin Kam Ho and Eugene M. Kleinberg. Solo cuenta con dos características y 862 muestras, por lo que puede usarse para ejemplificar como operan los métodos propuestos en dos dimensiones.

4.1.5. Conjunto Glass

El conjunto de datos Glass proviene del Servicio de Ciencia Forense de Estados Unidos de América, está caracterizado por 6 tipos de cristal o vidrio que pueden encontrarse en una escena de crimen y están definidos en términos de su contenido óxido (Na, Fe, K, etc.).

Cuadro 4.5: Atributos del conjunto Glass

Atributo	Tipo	Rango
RI	real	[1.51115, 1.53393]
Na	real	[10.73, 17.38]
Mg	real	[0.0, 4.49]
Al	real	[0.29, 3.5]
Si	real	[69.81, 75.41]
K	real	[0.0, 6.21]
Ca	real	[5.43, 16.19]
Ba	real	[0.0, 3.15]
Fe	real	[0.0, 0.51]

Clases: Type Glass {1, 2, 3, 4, 5, 6, 7}

Los siguientes conjuntos usados para las pruebas en esta Tesis, son una versión reducida del conjunto Glass.

glass 0: Es una versión no balanceada del conjunto Glass original, donde las muestras positivas pertenecen a la clase 0 y las muestras negativas pertenecen al resto.

glass 1: Es una versión no balanceada del conjunto Glass original, donde las muestras positivas pertenecen a la clase 1 y las muestras negativas pertenecen al resto.

glass 2: Es una versión no balanceada del conjunto Glass original, donde las muestras positivas pertenecen a la clase 2 y las muestras negativas pertenecen al resto.

glass 4: Es una versión no balanceada del conjunto Glass original, donde las muestras positivas pertenecen a la clase 4 y las muestras negativas pertenecen al resto.

glass 5: Es una versión no balanceada del conjunto Glass original, donde las muestras positivas pertenecen a la clase 5 y las muestras negativas pertenecen al resto.

glass 6: Es una versión no balanceada del conjunto Glass original, donde las muestras positivas pertenecen a la clase 6 y las muestras negativas pertenecen al resto.

4.1.6. Conjunto Liver disorders

Conjunto de datos originario de investigación médica por parte de BUPA Ltd, donada en 1990 por Richard S. Forsyth.

Conjunto formado por siete atributos, los primeros cinco pertenecen a pruebas de sangre, las cuales se piensa pueden permitir detectar problemas con el hígado

debido al consumo excesivo de alcohol. Cada muestra constituye un registro de un individuo de género masculino.

Cuadro 4.6: Atributos del conjunto Liver disorders

Atributo	Descripción
mcv	promedio del volumen corpuscular
alkphos	alkaline phosphotase
sgpt	alamine aminotransferase
sgot	aspartate aminotransferase
gammagt	gamma-glutamyl transpeptidase
drinks	promedio de bebidas consumidas por día
selector	campo usado para dividir el conjunto en dos clases

4.1.7. Conjunto Page blocks

Este conjunto contiene bloques extraídos de páginas de un documento a través de un proceso de segmentación y la tarea es determinar el tipo de contenido, texto (1), línea horizontal (2), gráficos (3), línea vertical (4) o imagen (5).

Cuadro 4.7: Atributos del conjunto Page blocks

Atributo	Tipo	Rango
Height	integer	[1, 804]
Lenght	integer	[1, 553]
Area	integer	[7, 143993]
Eccen	real	[0.0070, 537.0]
P_black	real	[0.052, 1.0]
P_and	real	[0.062, 1.0]
Mean_tr	real	[1.0, 4955.0]
Blackpix	integer	[1, 33017]
Blackand	integer	[7, 46133]
Wb_trans	integer	[1, 3212]

Clases: {1, 2, 3, 4, 5}

El conjunto *pageblocks 1-3 vs 4* usado para las pruebas en esta Tesis, es una versión reducida y no balanceada del conjunto original *pageblocks*, donde las muestras positivas pertenecen a la clase 4 (línea vertical), y las muestras negativas pertenecen al resto.

4.1.8. Conjunto Vehicle

Este conjunto de datos fue usado en una aplicación donde se requería determinar, dada una silueta, a cual de los cuatro tipos de vehículo corresponde, usando un conjunto de características extraídas de la silueta. El vehículo puede ser visto desde uno de varios ángulos diferentes.

Cuadro 4.8: Atributos del conjunto Vehicle

Atributo	Tipo	Rango
Compactness	integer	[73, 119]
Circularity	integer	[33, 59]
Distance_circularity	integer	[40, 112]
Radius_ratio	integer	[104, 333]
Praxis_aspect_ratio	integer	[47, 138]
Max_length_aspect_ratio	integer	[2, 55]
Scatter_ratio	integer	[112, 265]
Elongatedness	integer	[26, 61]
Praxis_rectangular	integer	[17, 29]
Length_rectangular	integer	[118, 188]
Major_variance	integer	[130, 320]
Minor_variance	integer	[184, 1018]
Gyration_radius	integer	[109, 268]
Major_skewness	integer	[59, 135]
Minor_skewness	integer	[0, 22]
Minor_kurtosis	integer	[0, 41]
Major_kurtosis	integer	[176, 206]
Hollows_ratio	integer	[181, 211]

Clases: {van, saab, bus, opel}

Los siguientes conjuntos usados para las pruebas en esta Tesis, son una versión reducida del conjunto original Vehicle.

vehicle2: Es una versión no balanceada del conjunto de siluetas de vehículos, donde las muestras positivas pertenecen a la clase 2 (Bus) y las muestras negativas pertenecen al resto.

vehicle3: Es una versión no balanceada del conjunto de siluetas de vehículos, donde las muestras positivas pertenecen a la clase 3 (Opel) y las muestras negativas pertenecen al resto.

4.1.9. Conjunto Yeast

El conjunto Yeast (levadura) fue creado por Kenta Nakai de la Universidad de Osaka y publicados en 1996. Está integrado por diversos datos acerca de este hongo microscópico y la aplicación es determinar la localización de proteínas dentro de la célula de la levadura. A continuación, en la Tabla 4.9, se presenta la descripción, proporcionada por el investigador, de los atributos para cada muestra del conjunto Yeast.

Clases: {MIT: Mitocondria, NUC: Núcleo, CYT: Citoesqueleto, ME1: Membrana con señal divisible, ME2: Membrana con señal de secuencia indivisible, ME3: Membrana sin terminal, EXC: Pared celular, VAC: Vacuola, POX: Peroxisoma, ERL: Lumen del retículo endoplasmatico}

Los siguientes conjuntos usados para las pruebas en esta Tesis, son una versión reducida del conjunto original Yeast.

yeast 3: Es una versión no balanceada del conjunto Yeast, donde las muestras positivas pertenecen a la clase ME3 y las muestras negativas pertenecen al resto.

Cuadro 4.9: Atributos del conjunto Yeast

Atributo	Tipo	Rango	Descripción
Mcg	real	[0.11, 1.0]	Valor obtenido por el método McGeoch para el reconocimiento de la señal de secuencia.
Gvh	real	[0.13, 1.0]	Valor obtenido por el método Von Heijne para el reconocimiento de la señal de secuencia.
Alm	real	[0.21, 1.0]	Puntuación de la membrana de ALOM.
Mit	real	[0.0, 1.0]	Puntuación del análisis discriminante de contenido de amino ácidos de proteinas mitocondriales y no mitocondiales.
Erl	real	[0.5, 1.0]	Presencia de la cadena "HDEL".
Pox	real	[0.0, 0.83]	Señal peroxisomal de focalización en la terminal.
Vac	real	[0.0, 0.73]	Puntuación del análisis discriminante del contenido de aminoácidos de proteínas vacuolares y extracelulares.
Nuc	real	[0.0, 1.0]	Puntuación del análisis discriminante de las señales de localización nuclear de proteínas nucleares y no nucleares.

yeast 4: Es una versión no balanceada del conjunto Yeast, donde las muestras positivas pertenecen a la clase ME2 y las muestras negativas pertenecen al resto.

yeast 5: Es una versión no balanceada del conjunto Yeast, donde las muestras positivas pertenecen a la clase ME1 y las muestras negativas pertenecen al resto.

yeast 6: Es una versión no balanceada del conjunto Yeast, donde las muestras positivas pertenecen a la clase EXC y las muestras negativas pertenecen al resto.

4.2. Desempeño de la SVM usando técnicas clásicas

Los primeros resultados fueron obtenidos evaluando la precisión de la SVM sobre el conjunto de datos original, estos resultados se presentan en la Tabla 4.10, las métricas usadas fueron AUC, Sn, Sn^{True} y Sn^{False}; cada valor reportado es el promedio de 10 pruebas. Los parámetros usados para la SVM fueron Kernel RBF, C de 0,1 y un valor gamma de 0,5.

De acuerdo con los resultados obtenidos, puede notarse que si aplicamos la SVM sobre el conjunto de datos original, las precisiones se sesgan hacia Sn^{False} obteniendo valores superiores a 0,85 junto al AUC con valores arriba de 0,75 sin embargo, las métricas Sn y Sn^{True} reportan un valor por debajo de 0,75, salvo el conjunto vehicle 2 y ecoli 2 que alcanzaron un valor alto en todas las métricas. También se puede notar que hay casos donde existen altos valores para el AUC y Sn^{False}, llegando incluso al máximo de 1,0 sin embargo, al verificar el valor Sn^{True}, se encuentra que los valores para esta métrica están muy bajos e incluso de 0,0 para los conjuntos *glass 2*, *glass 5*, *yeast 4* y *yeast 6*. Fue este resultado el que motivó a redefinir la mejora de parámetros para los métodos propuestos, buscando siempre mejorar todas las métricas y no sólo el AUC ya que podría darnos conclusiones erróneas.

Por otro lado al aplicar Bajo-muestreo o Sobre-muestreo la precisión mejora pero en ambos casos 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 la Tabla 4.11 se listan los resultados para la técnica de Bajo-muestreo y se puede notar que el sesgo en la clasificación es en su mayoría para Sn y Sn^{True}, presentando mejores resultados en 16 de 21 conjuntos para Sn y Sn^{True} respecto a Sn^{False}.

Cuadro 4.10: Desempeño de SVM con el conjunto no balanceado

Conjunto	No Balanceado			
de Datos	AUC	Sn	Sn^{True}	Sn^{False}
liver disorders	0.75	0.48	0.48	0.85
four class	0.87	0.51	0.51	0.97
glass 1	0.79	0.08	0.08	0.99
diabetes	0.81	0.56	0.56	0.86
glass 0	0.85	0.31	0.31	0.92
vehicle 2	0.99	0.90	0.90	0.98
vehicle 3	0.80	0.13	0.13	0.97
ecoli 1	0.95	0.69	0.69	0.96
ecoli 2	0.96	0.81	0.81	0.98
glass 6	0.93	0.70	0.70	0.98
yeast 3	0.98	0.66	0.66	0.98
ecoli 3	0.92	0.44	0.44	0.98
glass 2	0.66	0.00	0.00	1.00
cleveland 0 vs 4	0.98	0.15	0.15	1.00
glass 4	0.97	0.05	0.05	1.00
ecoli 4	1.00	0.75	0.75	1.00
pageblocks 1-3 vs 4	1.00	0.50	0.50	1.00
glass 5	0.97	0.00	0.00	1.00
yeast 4	0.81	0.00	0.00	1.00
yeast 5	0.99	0.16	0.16	1.00
yeast 6	0.90	0.00	0.00	1.00

Por último, para SMOTE, se usaron como parámetros N igual a 400 y 10 k vecinos. Esta última técnica también presenta la debilidad de sesgar la precisión, clasificando la mayoría de los patrones de prueba como positivos cuando esta sesgada a Sn^{True} o negativos cuando esta sesgada a Sn^{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 predecir si el clasificador será bueno detectando patrones positivos o si la técnica de balanceo de clases es buena.

Cuadro 4.11: Desempeño de SVM con Bajo-muestreo

Conjunto de Datos	Bajo-muestreo			
	AUC	Sn	Sn^{True}	Sn^{False}
liver disorders	0.74	0.68	0.68	0.69
four class	0.87	0.78	0.78	0.78
glass 1	0.77	0.84	0.84	0.46
diabetes	0.81	0.74	0.74	0.71
glass 0	0.83	1.00	1.00	0.44
vehicle 2	0.99	0.97	0.96	0.91
vehicle 3	0.79	0.76	0.76	0.67
ecoli 1	0.94	0.92	0.92	0.84
ecoli 2	0.96	0.93	0.93	0.91
glass 6	0.96	0.80	0.80	0.95
yeast 3	0.98	0.91	0.91	0.93
ecoli 3	0.95	0.93	0.93	0.83
glass 2	0.64	0.97	0.97	0.31
cleveland 0 vs 4	0.94	0.95	0.95	0.70
glass 4	0.93	0.95	0.95	0.80
ecoli 4	1.00	1.00	1.00	0.93
pageblocks 1-3 vs 4	0.99	0.98	0.98	0.90
glass 5	0.92	0.90	0.90	0.83
yeast 4	0.84	0.75	0.75	0.87
yeast 5	0.99	1.00	1.00	0.91
yeast 6	0.92	0.86	0.86	0.88

Cuadro 4.12: Desempeño de SVM con Sobre-muestreo

Conjunto de Datos	Sobre-muestreo			
	AUC	Sn	Sn^{True}	Sn^{False}
liver disorders	0.75	0.64	0.64	0.75
four class	0.88	0.81	0.81	0.80
glass 1	0.79	0.80	0.80	0.57
diabetes	0.81	0.69	0.69	0.76
glass 0	0.83	0.99	0.99	0.47
vehicle 2	0.99	0.82	0.82	0.98
vehicle 3	0.81	0.57	0.57	0.82
ecoli 1	0.94	0.89	0.89	0.86
ecoli 2	0.96	0.92	0.92	0.94
glass 6	0.94	0.80	0.80	0.98
yeast 3	0.97	0.65	0.65	0.98
ecoli 3	0.94	0.89	0.89	0.88
glass 2	0.64	0.87	0.87	0.37
cleveland 0 vs 4	0.97	0.20	0.20	0.99
glass 4	0.97	0.95	0.95	0.94
ecoli 4	0.99	0.90	0.90	0.98
pageblocks 1-3 vs 4	1.00	0.90	0.90	0.98
glass 5	0.97	0.80	0.80	0.93
yeast 4	0.87	0.71	0.71	0.88
yeast 5	0.99	1.00	1.00	0.94
yeast 6	0.94	0.81	0.81	0.92

Cuadro 4.13: Desempeño de SVM con SMOTE

Conjunto de Datos	SMOTE			
	AUC	Sn	Sn^{True}	Sn^{False}
liver disorders	0.71	0.89	0.89	0.28
four class	0.83	0.91	0.91	0.71
glass 1	0.75	0.91	0.91	0.31
diabetes	0.79	0.83	0.83	0.62
glass 0	0.81	0.99	0.99	0.45
vehicle 2	0.99	0.97	0.97	0.93
vehicle 3	0.82	0.88	0.88	0.66
ecoli 1	0.95	0.91	0.91	0.84
ecoli 2	0.96	0.93	0.93	0.94
glass 6	0.96	0.80	0.80	0.98
yeast 3	0.98	0.86	0.86	0.96
ecoli 3	0.94	0.83	0.83	0.92
glass 2	0.64	0.00	0.00	1.00
cleveland 0 vs 4	0.98	0.60	0.60	0.99
glass 4	0.97	0.95	0.95	0.95
ecoli 4	1.00	0.93	0.93	0.99
pageblocks 1-3 vs 4	1.00	0.94	0.94	1.00
glass 5	0.97	0.70	0.70	0.99
yeast 4	0.86	0.54	0.54	0.97
yeast 5	0.99	0.93	0.93	0.97
yeast 6	0.94	0.70	0.70	0.98

4.3. Desempeño de la SVM usando los métodos propuestos

4.3.1. Gráfica ROC con conjuntos no balanceados

La gráfica proporcionada por el método *Receiver Operating Characteristic* (ROC) es ampliamente usado en investigación para analizar el desempeño de clasificadores binarios y al medir el área bajo la curva ROC, comúnmente conocida por sus siglas AUC, se puede tener una representación numérica de que tan separables son las clases analizadas.

La ventaja más importante 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. Una descripción más detallada sobre cómo obtener esta curva puede encontrarse en [(Fawcett, 2006)]. Sin embargo, cuando el desbalance entre clases es alto, la gráfica ROC puede devolver un AUC de 1.0 indicando que la precisión de clasificación es perfecta cuando en realidad, el clasificador es malo para clasificar muestras positivas.

En un principio se trato de distinguir qué combinación de parámetros era mejor como solución al problema, en función del área bajo la curva, sin embargo, los conjuntos no balanceados *glass 1* (Figura 4.1), *glass 2*, *glass 4*, *glass 5*, *yeast 4* y *yeast 6* (Figura 4.3) presentaban una alta precisión para AUC y Sn^{False}, muy cercana a 1,0, pero muy baja para Sn y Sn^{True}, casi de 0,0; esta peculiaridad puede llevarnos a conclusiones erróneas cuando trabajamos sobre conjuntos no balanceados.

Cuadro 4.14: Precisiones alcanzadas por SVM para el conjunto Glass 1

Método	Precisión			
	AUC	Sn	Sn^{True}	Sn^{False}
Conjunto de datos no balanceado	0.741	0.067	0.067	0.963
Bajo-muestreo	0.783	0.800	0.800	0.481
Sobre-muestreo	0.788	0.800	0.800	0.481
SMOTE	0.815	1.000	1.000	0.370
Método propuesto	**1.000**	1.000	1.000	**0.000**

La gráfica ROC, Figura 4.1, por ejemplo muestra un AUC igual a 1,0 para el conjunto *glass 1*, pero al analizar la Tabla 4.14 se nota que aunque el método propuesto obtiene Sn y Sn^{True} de 1,0, el valor Sn^{False} es muy pequeño, casi de 0,0, por lo que la precisión de clasificación ahora esta sesgada hacia la clase minoritaria. De igual forma la técnica SMOTE al mejorar el AUC a 0,815, obtiene un Sn^{True} de 1,0, pero un valor bajo en Sn^{False}, de 0,370.

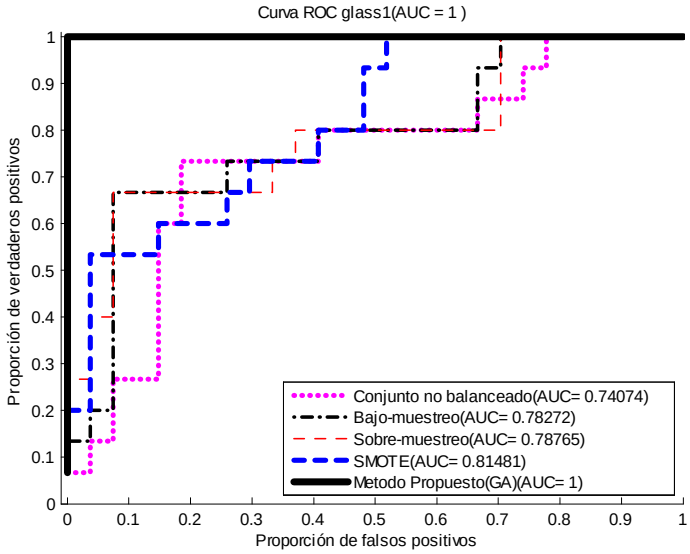


Figura 4.1: Gráfica ROC con AUC=1.0 para el conjunto Glass 1

En la Figura 4.3.1, las gráficas ROC muestran un valor AUC máximo de 1,0 sin embargo, al analizar los detalles de la precisión en la Tabla 4.15, se encuentra que para estos cinco conjuntos, el valor Sn^{True} tiene un valor muy bajo, cercano a 0.0.

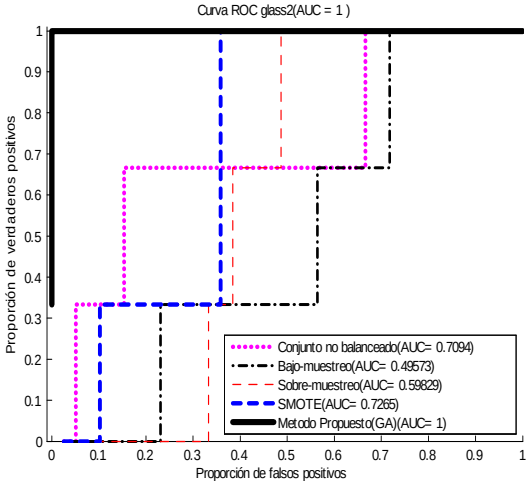
Cuadro 4.15: Precisiones alcanzadas por SVM con el método propuesto

Conjunto de Datos	Precisión			
	AUC	Sn	Sn^{True}	Sn^{False}
glass 2	**1.000**	1.000	1.000	**0.000**
glass 4	**1.000**	1.000	1.000	**0.000**
glass 5	**1.000**	1.000	1.000	**0.122**
yeast 4	**1.000**	1.000	1.000	**0.000**
yeast 6	**1.000**	1.000	1.000	**0.000**

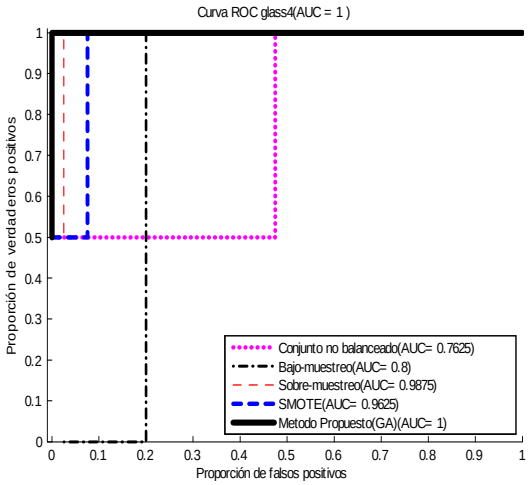
Esta peculiaridad del AUC sobre los conjuntos no balanceados provocaba que aún teniendo un AUC máximo, igual a 1,0, no se pudiera mejorar la clasificación de las muestras para la clase minoritaria independientemente de usar una búsqueda en malla o usando un algoritmo genético, por lo que para distinguir si una solución es mejor a otra hay que definir una función usando las cuatro métricas, AUC, Sn, Sn^{True}, Sn^{False}, esto evitará que la precisión de clasificación tenga tendencia a clasificar las muestras mayoritariamente como negativas cuando este sesgada hacia Sn^{False} o como positivas cuando este sesgada hacia Sn^{True}.

4.3.2. Ejemplo ilustrativo con el conjunto no balanceado Four class

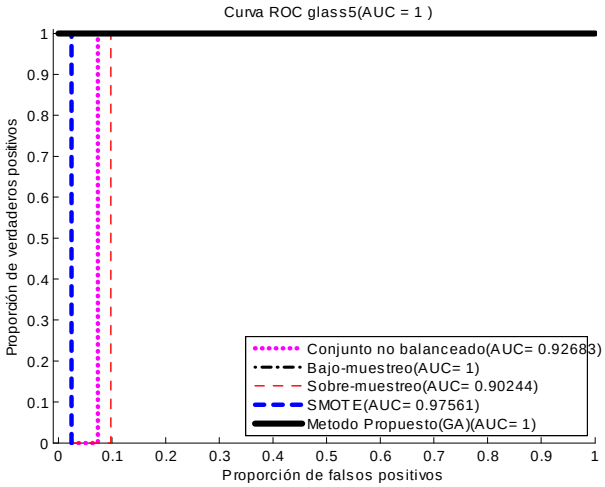
Dentro de los conjuntos no balanceados analizados se encuentra el conjunto Four class que cuenta con dos características y un bajo radio de desbalance, de 1:1.8, por lo que debería ser fácil de clasificar, sin embargo, la distribución de los datos, como se muestra en la Figura 4.4, complica el entrenamiento de la SVM, sesgando



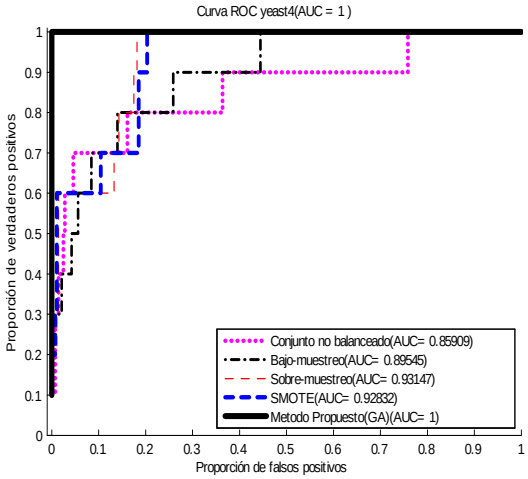
a) Glass 2



b) Glass 4



c) Glass 5



d) Yeast 4

Figura 4.2: Gráficas ROC con AUC=1.0 para los conjuntos a)Glass 2, b)Glass 4, c) Glass 5 y d)Yeast 4

la precisión de clasificación hacia la clase mayoritaria.

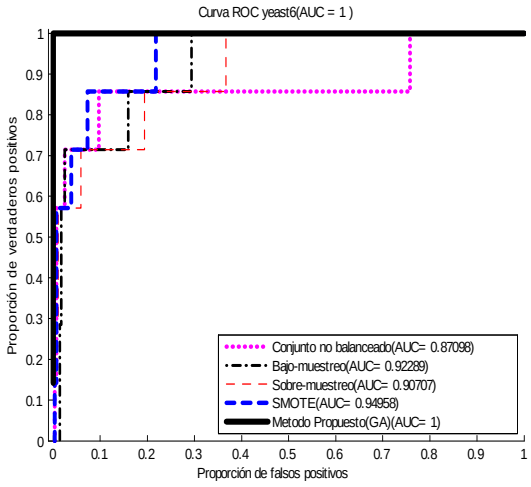


Figura 4.3: Gráfica ROC con AUC=1.0 para el conjunto Yeast 6

Al analizar el desempeño de la SVM usando el conjunto original, como se muestra en la Tabla 4.10, se nota que la precisión esta sesgada hacia la clase negativa, con Sn^{True} promedio de 51 % frente a 97 % para Sn^{False} y AUC de 87 %. Técnicas como Bajo-muestreo y Sobre-muestreo logran mejorar la precisión Sn^{True} a 78 % y 81 % respectivamente, pero el AUC se mantiene cerca de 87 %. Por último la técnica SMOTE, al crear nuevos puntos sintéticos para reforzar el entrenamiento, logra mejorar la precisión para clasificar datos positivos hasta 91 % pero ve disminuida a 71 % la precisión para detectar datos negativos, esto es porque SMOTE no distingue la cercanía entre los puntos positivos y negativos, por lo que al crear nuevos puntos podría introducir ruido. Esto sucede cuando algún nuevo punto queda ubicado en la clase negativa pero es etiquetado como positivo al momento de crearlo.

Como se puede observar cada una de las técnicas anteriores muestra un sesgo en la clasificación, por lo que el objetivo de los métodos propuestos fue mejorar el desempeño de la SVM sobre las cuatro métricas, con el fin de evitar el sesgo antes

mencionado. A continuación se presenta un ejemplo ilustrativo de la aplicación de los dos métodos propuestos sobre el conjunto Four class que cuenta con bajo radio de desbalance, primero aplicando la excitación de SV y después aplicando la creación de puntos artificiales.

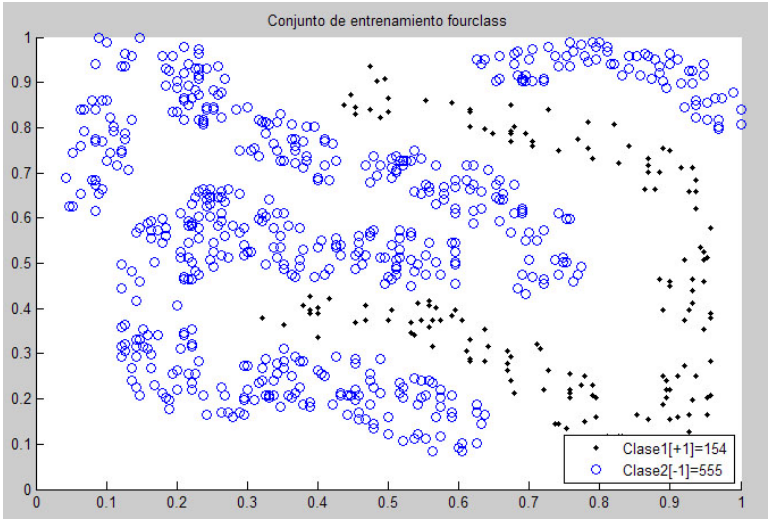


Figura 4.4: Distribución del conjunto no balanceado Four class

Aplicando la excitación de SV

El método propuesto se basa en excitar los SV^+, sin embargo, en cada iteración se necesitan calcular los parámetros del nuevo hiperplano de separación. Para esta tarea se destinó una búsqueda en malla y para este conjunto en particular, el método propuesto alcanzó en menos de 6 iteraciones la precisión máxima de 1.0 para las cuatro métricas, AUC, Sn, Sn^{True} y Sn^{False}.

En la Tabla 4.16 puede verse la precisión promedio alcanzada en diez pruebas para cada una de las técnicas analizadas, la diferencia es notable frente al conjunto original e incluso sobre la técnica SMOTE que también se basa en crear puntos

sintéticos sin embargo, las diferencias se hacen notables en la Gráfica ROC, Figura 4.5, donde el primer método propuesto alcanza una precisión de 100 % frente a SMOTE con 83 %.

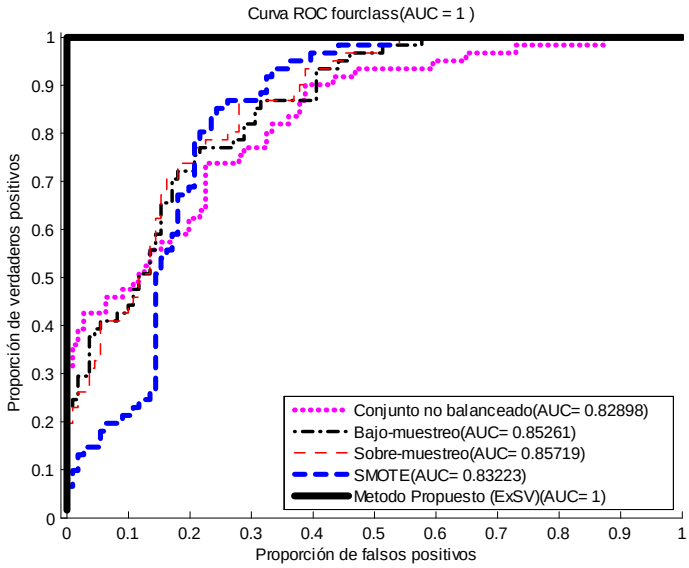


Figura 4.5: Gráfica ROC para una de las pruebas aplicadas al conjunto Four class usando excitación de SV

Cuadro 4.16: Precisiones alcanzadas por SVM para el conjunto Four class sobre 10 pruebas usando excitación de SV

Método	Precisión			
	AUC	Sn	Sn^{True}	Sn^{False}
Conjunto de datos no balanceado	0.87	0.51	0.51	0.97
Bajo-muestreo	0.87	0.78	0.78	0.78
Sobre-muestreo	0.88	0.81	0.81	0.80
SMOTE	0.83	0.91	0.91	0.71
Método propuesto:	**1.000**	**1.000**	**1.000**	**1.000**
Excitación de SV				

Aplicando la creación de puntos artificiales

A diferencia de los métodos anteriores, el Método Propuesto, obtuvo una precisión promedio de 100 % para las métricas Sn, Sn^{True}, Sn^{False} y AUC, debido a que se reforzó el aprendizaje de la SVM al crear nuevos puntos artificiales dentro de la clase minoritaria y en la frontera entre las clases, evitando que los nuevos puntos cayeran en el espacio de la clase mayoritaria.

La creación de puntos artificiales requiere varios parámetros, primero la cantidad de SV^+ vecinos seleccionados por cada SV^+ analizado, siendo este un Vector Soporte de la clase minoritaria; después la proporción de distancia α a la que será desplazado el SV^+, la cantidad de SV^- vecinos seleccionados por cada SV^+ y su proporción de desplazamiento β. Así mismo, para obtener una buena precisión se requieren de los parámetros γ para el kernel RBF, C y tipo de kernel, estos parámetros son propios de la SVM que, junto con los del método propuesto, suman siete parámetros.

La forma de mejorar los parámetros en un tiempo razonable fue la introducción de un algoritmo genético que regresara una combinación de buenos parámetros como lo muestra la Tabla 4.17, en ella se puede ver que con sólo dos generaciones el GA converge y se obtiene la mayor precisión para la SVM en las cuatro métricas. En particular en la segunda generación, el mejor individuo creo dos nuevos puntos por cada SV^+ dentro de la clase minoritaria, desplazándolos un 74 % de distancia respecto al SV^+ original, mientras que dentro de la frontera entre clases se creó un nuevo punto por cada SV^+, desplazándolo a 31 % de distancia respecto al SV^+ original.

En el caso de elegir SV^- como vecinos cercanos al SV^+ analizado, los vecinos residen en la clase mayoritaria, sin embargo, el algoritmo tiene especial cuidado

Cuadro 4.17: Precisión alcanzada por SVM usando los parámetros del cromosoma para cada generación y el método propuesto aplicado al conjunto Four class

Gene-	Aptitud				Mejor Individuo: Fenotipo						
ración	AUC	Sn	Sn^T	Sn^F	γ	C	K	SV_k^+	α	SV_k^-	β
1	0.990	0.951	0.951	0.973	0.252	0.588	2.0	1.33	0.47	0	0.63
2	1.000	1.000	1.000	1.000	0.057	0.322	2.0	2.00	0.74	1	0.31

K : Tipo de Kernel
Sn^T : Sn^{True}
Sn^F : Sn^{False}

de no agregar nuevos puntos dentro de esta clase ya que estaría introduciendo ruido. Esta restricción está regulada por la proporción de desplazamiento β, que de acuerdo a la Tabla 3.1, tiene un rango fijo entre $[0{,}01, 0{,}90]$, por lo que un nuevo punto nunca podra llegar hasta la clase negativa.

En la Tabla 4.18 puede verse la precisión promedio alcanzada en diez pruebas para cada una de las técnicas analizadas, la diferencia es notable frente al conjunto original e incluso sobre la técnica SMOTE que también se basa en crear puntos sintéticos sin embargo, las diferencias se hacen notables en la Gráfica ROC, Figura 4.7, donde SMOTE alcanza un AUC de 0,83 mientras que el método propuesto obtuvo 1,0.

También hay que hacer notar que la desviación estándar para el método propuesto fue de cero, por lo que, para este conjunto en particular se logró una precisión perfecta en todas las pruebas.

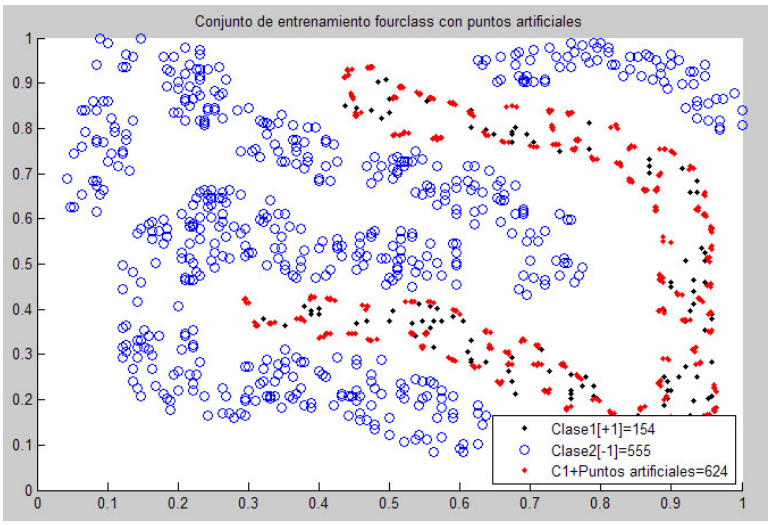


Figura 4.6: Distribución del conjunto Four class con nuevos puntos artificiales

Cuadro 4.18: Precisiones alcanzadas por SVM para el conjunto Four class sobre 10 pruebas usando puntos artificiales

Método	Precisión			
	AUC	Sn	Sn^{True}	Sn^{False}
Conjunto de datos no balanceado	0.87	0.51	0.51	0.97
Bajo-muestreo	0.87	0.78	0.78	0.78
Sobre-muestreo	0.88	0.81	0.81	0.80
SMOTE	0.83	0.91	0.91	0.71
Método propuesto: Creación de Puntos Artificiales	**1.000**	**1.000**	**1.000**	**1.000**

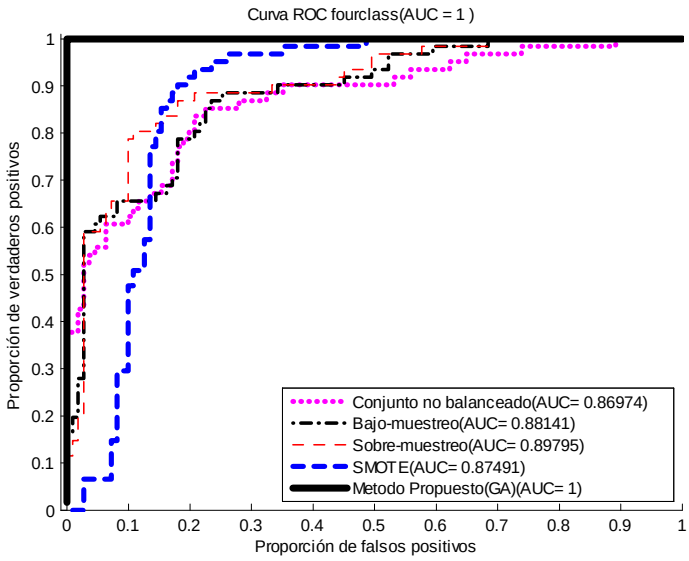


Figura 4.7: Gráfica ROC para una de las pruebas aplicadas al conjunto Four class usando puntos artificiales

4.3.3. Ejemplo ilustrativo con el conjunto no balanceado Yeast 6

Dentro de los conjuntos no balanceados analizados se encuentra el conjunto *yeast 6* que cuenta con ocho características y un alto radio de desbalance, de 1 : 41,4, por lo que presenta la mayor dificultad en la tarea de clasificación para los conjuntos analizados.

Al analizar el desempeño de la SVM usando el conjunto original, como se muestra en la Tabla 4.10, se puede notar que la precisión esta sesgada hacia la clase negativa, con Sn^{True} promedio de 0 % frente a 100 % para Sn^{False} y AUC de 90 %. Técnicas como Bajo-muestreo y Sobre-muestreo logran mejorar la precisión Sn^{True} a 86 % y 81 % respectivamente, pero el AUC se mantiene cerca de 92 %. Por último la técnica SMOTE, al crear nuevos puntos sintéticos para reforzar el entrenamiento, reporta una precisión para clasificar datos positivos de 70 % y de 98 % para detectar datos negativos.

Como se puede observar cada una de las técnicas anteriores muestra un sesgo en la clasificación.

A continuación se presenta un ejemplo ilustrativo de la aplicación de los dos métodos propuestos sobre el conjunto *yeast 6* que cuenta con alto radio de desbalance, primero aplicando la excitación de SV y después aplicando la creación de puntos artificiales.

Aplicando la excitación de SV

El método propuesto se basa en excitar los SV^+ sin embargo, en cada iteración necesita calcular los parámetros del nuevo hiperplano de separación. Para esta tarea

se destinó una búsqueda en malla y para este conjunto en particular, el método propuesto alcanzó en menos de 6 iteraciones la precisión máxima de 1.0 para la métrica Sn^{True}. Las iteraciones para este método pueden encontrarse en la Tabla 4.19, hay que hacer notar que aunque el método podría llegar muy rápido a una precisión de 100 % para alguna de las métricas, es necesario equilibrar el incremento entre ellas para no sesgar la clasificación, es decir mantener un nivel similar de precisión para todas las métricas.

En la Tabla 4.20 puede verse la precisión promedio alcanzada en diez pruebas para cada una de las técnicas analizadas, la diferencia es notable frente al conjunto original que tenia 100 % para Sn^{False} pero de 0 % para Sn^{True}, esto lo hacía incapaz de clasificar patrones positivos, mientras que la excitación de SV alcanzó una precisión promedio de 84 % y 95 % para Sn^{True} y Sn^{False} respectivamente.

Cuadro 4.19: Precisión alcanzada por SVM usando una búsqueda en malla y el método propuesto aplicado al conjunto Yeast 6

Iteración	Precisión			
	AUC	Sn	Sn^{True}	Sn^{False}
1	0.880	0.429	0.429	0.969
2	0.955	0.714	0.714	0.958
3	0.955	0.714	0.714	0.958
4	0.886	0.857	0.857	0.865
5	0.985	0.857	0.857	0.952
6	0.976	1.000	1.000	0.889

En la Figura 4.8 se muestran las curvas ROC alcanzadas por cada técnica analizada, en ella la diferencia es mínima entre el método propuesto y las demás técnicas sin embargo, esto no quiere decir que no haya mejorado la precisión, por el contrario al analizar las métricas Sn^{True} y Sn^{False}, se encuentra que estas alcanzaron 84 % y 95 % respectivamente, superando la habilidad para clasificar patrones positivos y reduciendo el sesgo de clasificación frente a las otras técnicas.

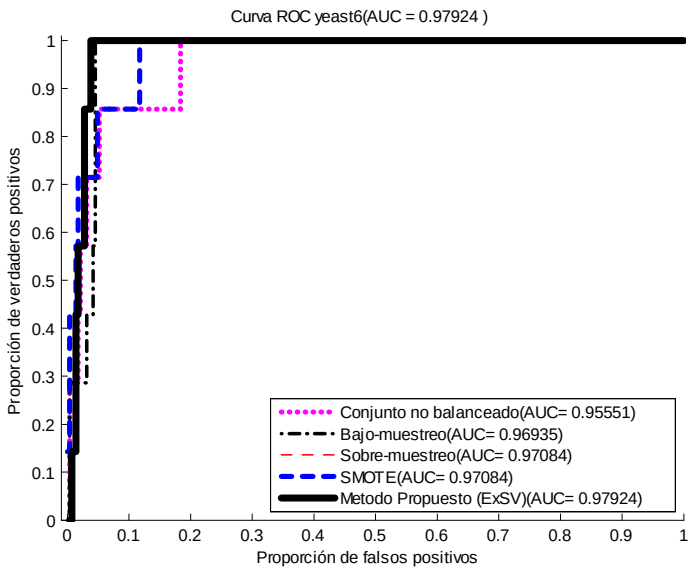


Figura 4.8: Gráfica ROC para una de las pruebas aplicadas al conjunto Yeast 6 usando excitación de SV

Cuadro 4.20: Precisiones alcanzadas por SVM para el conjunto Yeast 6 sobre 10 pruebas usando excitación de SV

Método	Precisión			
	AUC	Sn	Sn^{True}	Sn^{False}
Conjunto de datos no balanceado	0.90	0.00	0.00	1.00
Bajo-muestreo	0.92	0.86	0.86	0.88
Sobre-muestreo	0.94	0.81	0.81	0.92
SMOTE	0.94	0.70	0.70	0.98
Método propuesto:	**0.95**	**0.84**	**0.84**	**0.95**
Excitación de SV				

Aplicando la creación de puntos artificiales

El Método Propuesto, como se muestra en la Tabla 4.21, obtuvo una precisión promedio superior a 91 % para las métricas Sn, Sn^{True}, mientras que para Sn^{False} y AUC el valor subió a 98 %. Para lograr este desempeño, se tuvieron que crear

84 nuevos puntos artificiales, estos fueron etiquetados como "+1" y se introdujeron en cada prueba junto con el conjunto de entrenamiento, esto balanceo la clase minoritaria respecto a la mayoritaria.

La creación de puntos artificiales requiere varios parámetros, primero la cantidad de SV^+ vecinos seleccionados por cada SV^+ analizado, siendo este un Vector Soporte de la clase minoritaria; después la proporción de distancia α a la que será desplazado el SV^+, la cantidad de SV^- vecinos seleccionados por cada SV^+ y su proporción de desplazamiento β. Así mismo, para obtener una buena precisión se requieren de los parámetros γ para el kernel RBF, C y tipo de kernel, estos parámetros son propios de la SVM que, junto con los del método propuesto, suman siete parámetros.

Cuadro 4.21: Precisiones alcanzadas por SVM para el conjunto Yeast 6 sobre 10 pruebas usando puntos artificiales

Método	Precisión			
	AUC	Sn	Sn^{True}	Sn^{False}
Conjunto de datos no balanceado	0.90	0.00	0.00	1.00
Bajo-muestreo	0.92	0.86	0.86	0.88
Sobre-muestreo	0.94	0.81	0.81	0.92
SMOTE	0.94	0.70	0.70	0.98
Método propuesto: Creación de Puntos Artificiales	**0.98**	**0.91**	**0.91**	**0.99**

La forma de mejorar los parámetros en un tiempo razonable fue la introducción de un algoritmo genético que regresara una combinación de buenos parámetros como lo muestra la Tabla 4.22. En esta Tabla se puede ver que con 18 generaciones el GA pasa de 0,857 a 1,0 para la métrica Sn^{True} sin embargo, Sn^{False} aún es bajo, de 0,799; es el AG, quien a través de la exploración y explotación del espacio de búsqueda, mejora los parámetros. Por ejemplo, en la primera generación se usaba

un kernel lineal, pero al mutar este parámetro en la generación 18, el GA cambió el valor a kernel RBF, mejorando la precisión y después en la generación 30, la precisión con este conjunto llego a 0,956 para el AUC, de 1,0 para Sn y Sn^{True} y volvió a subir Sn^{False} a 0,872, estos valores se mantuvieron hasta la generación 100, que aunque en los resultados sólo se usaron 12 generaciones como máximo, para este ejemplo se tabularon hasta 100 generaciones para mostrar la convergencia rápida del AG.

En particular cuando el GA cambia los parámetros de la generación 18 a la 23, el valor C baja de 0,462 a 0,193, el parámetro α baja de 0,59 a 0,16 y β baja de 0,45 a 0,14, esto quiere decir que es mejor crear nuevos puntos artificiales dentro de la clase minoritaria y uno en la frontera pero dejarlos ubicados cerca del SV^+ analizado, es por eso que los desplazamientos son muy cortos.

Cuadro 4.22: Precisión alcanzada por SVM usando los parámetros del cromosoma para cada generación y el método propuesto aplicado al conjunto Yeast 6

Gene-	Aptitud				Mejor Individuo: Fenotipo						
ración	AUC	Sn	Sn^T	Sn^F	γ	C	K	SV_k^+	α	SV_k^-	β
1	0.925	0.857	0.857	0.872	0.063	0.462	1	0	0.73	1	0.49
18	0.938	1.000	1.000	0.799	0.176	0.462	2	2	0.59	1	0.45
23	0.947	1.000	1.000	0.827	0.183	0.193	2	2	0.16	1	0.14
24	0.954	1.000	1.000	0.841	0.200	0.961	2	2	0.79	1	0.18
30	0.956	1.000	1.000	0.872	0.145	0.130	2	2	0.79	1	0.09
50	0.956	1.000	1.000	0.872	0.145	0.130	2	2	0.79	1	0.09
80	0.956	1.000	1.000	0.872	0.145	0.130	2	2	0.79	1	0.09
100	0.956	1.000	1.000	0.872	0.145	0.130	2	2	0.79	1	0.09

K : Tipo de Kernel
Sn^T : Sn^{True}
Sn^F : Sn^{False}

En la Tabla 4.21 puede verse la precisión promedio alcanzada en diez pruebas para cada una de las técnicas analizadas, la diferencia es significativa frente al

conjunto original en la Figura 4.9 se muestran las curvas ROC alcanzadas por cada técnica analizada, en ella la diferencia es de 0.5 entre el método propuesto y el conjunto original, mientras que para las técnicas Bajo-muestreo, Sobre-muestreo y SMOTE, la diferencia es mínima sin embargo, esto no quiere decir que no haya mejorado la precisión, por el contrario si se analizan las métricas Sn^{True} y Sn^{False}, se encuentra que estas alcanzaron 91 % y 99 % respectivamente, superando la habilidad para clasificar patrones positivos y reduciendo el sesgo de clasificación frente a las otras técnicas.

También hay que hacer notar que la desviación estándar máxima para el método propuesto fue de 0,07 por lo que es una técnica estable.

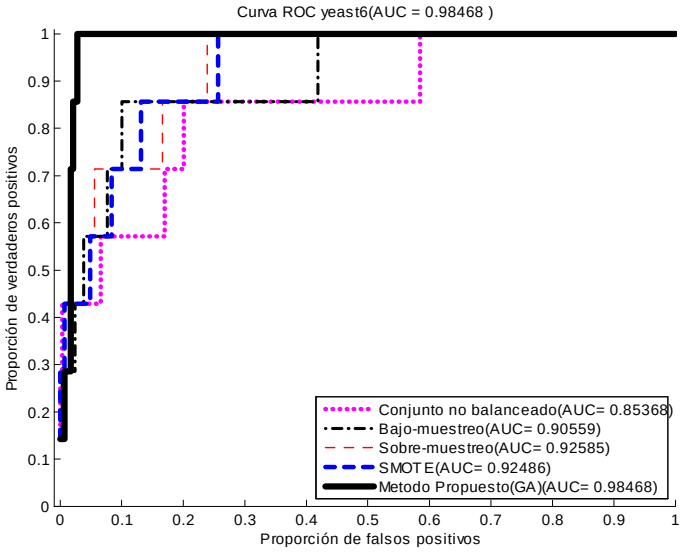


Figura 4.9: Gráfica ROC para una de las pruebas aplicadas al conjunto Yeast 6 usando puntos artificiales

4.3.4. Desempeño de la SVM usando el Método 1

El primer método propuesto es la excitación de SV. Para realizar las pruebas, se destinó 80 % del conjunto de datos original para entrenamiento y 20 % para prueba, eligiendo las muestras de forma aleatoria. Este método obtiene un conjunto de SV a partir del hiperplano de separación inicial, los etiqueta para distinguir entre SV^+ y SV^-, esto le permitirá trabajar sólo con los SV correspondientes a la clase minoritaria, desplazarlos un épsilon hacia la frontera de decisión y poder crear nuevos SV. Con el fin de mejorar la precisión de las SVM se agregó una búsqueda en malla para mejorar los parámetros en cada iteración del algoritmo.

En la Tabla 4.23 se presentan los resultados obtenidos para el método propuesto, siendo cada valor un promedio de 10 pruebas, las métricas usadas para evaluar la precisión de las SVM fueron área bajo la curva ROC (AUC), Sn, relación de verdaderos positivos (Sn^{True}) y relación de verdaderos negativos (Sn^{False}).

Cuadro 4.23: Precisiones alcanzadas por SVM aplicando la excitación de SV

Conjunto	Excitación de SV				Desviación estandar			
de Datos	AUC	Sn	Sn^{True}	Sn^{False}	AUC	Sn	Sn^{True}	Sn^{False}
liver disorders	0.73	0.75	0.75	0.67	0.05	0.04	0.04	0.06
four class	1.00	1.00	1.00	1.00	0.00	0.00	0.00	0.00
glass 1	0.84	0.92	0.92	0.70	0.06	0.06	0.06	0.10
diabetes	0.85	0.86	0.86	0.73	0.02	0.03	0.03	0.02
glass 0	0.92	0.94	0.94	0.86	0.04	0.05	0.05	0.07
vehicle 2	0.99	0.98	0.98	0.97	0.01	0.02	0.02	0.03
vehicle 3	0.85	0.87	0.87	0.77	0.03	0.04	0.04	0.03
ecoli 1	0.96	0.99	0.99	0.87	0.02	0.03	0.03	0.03
ecoli 2	0.96	0.89	0.89	0.95	0.04	0.10	0.10	0.02
glass 6	0.99	0.98	0.98	0.98	0.02	0.06	0.06	0.06
yeast 3	0.98	0.97	0.97	0.94	0.01	0.02	0.02	0.01
ecoli 3	0.97	0.99	0.99	0.93	0.02	0.05	0.05	0.03
glass 2	0.77	0.87	0.87	0.50	0.20	0.17	0.17	0.19
cleveland-0_vs_4	0.99	1.00	1.00	0.93	0.02	0.00	0.00	0.11
glass 4	0.98	1.00	1.00	0.90	0.04	0.00	0.00	0.08
ecoli 4	0.99	1.00	1.00	0.96	0.01	0.00	0.00	0.05
pageblocks-1-3vs4	0.99	1.00	1.00	0.97	0.01	0.00	0.00	0.03
glass 5	0.99	1.00	1.00	0.94	0.02	0.00	0.00	0.09
yeast 4	0.92	0.94	0.94	0.83	0.04	0.05	0.05	0.07
yeast 5	0.99	1.00	1.00	0.96	0.00	0.00	0.00	0.02
yeast 6	0.95	0.84	0.84	0.95	0.06	0.18	0.18	0.04

En los resultados fue notorio que aunque el AUC es una buena métrica para medir la precisión, sobre conjuntos no balanceados puede darse el caso donde el AUC y el Sn^{False} sean cercanos a 1,0 pero Sn^{True} y Sn sean casi 0,0, por lo que la búsqueda en malla implemento una mejora sobre los valores de todas las métricas usadas y no sólo el AUC, evitando que la precisión estuviera sesgada hacia alguna de ellas.

Al final de la Tabla se reporta la desviación estándar de cada métrica para las

10 pruebas realizadas sobre cada conjunto de datos, notándose que la STD no pasa de 0,10 para Sn^{True}, excepto para *glass 2* y *yeast 6* por lo que se considera una técnica estable y la precisión Sn^{True} fue superior a 0,94 en 14 de los 22 conjuntos de datos no balanceados.

El método propuesto mostró mejor desempeño en conjuntos con desbalance mayor a 9, manteniendo una precisión alta para Sn^{True} y Sn^{False}, a diferencia de las demás técnicas, donde se nota claramente sesgada la precisión hacia una de las dos métricas.

A continuación se presentan las gráficas ROC para cada conjunto de datos, cada gráfica corresponde a la precisión promedio alcanzada en las pruebas y se grafican las curvas para cada uno de los cinco métodos analizados.

La primera gráfica, Figura 4.10, correspondiente al conjunto *liver disorders*, presenta la comparación del método propuesto respecto a las demás técnicas analizadas y aunque la diferencia es poca para el AUC, la importancia reside en el hecho de que se mantiene una precisión similar para las métricas restantes (Sn, Sn^{True} y Sn^{False}), de 0,48, 0,48 y 0,85 respectivamente para el conjunto no balanceado frente a la excitación de SV con 0,75, 0,75 y 0,67.

Los conjuntos *liver disorders* hasta *ecoli 3* tienen un radio de desbalance menor a 10 y mientras las precisiones presentan un sesgo de clasificación hacia la clase negativa usando el conjunto original, con la excitación de SV, como se muestra en la Tabla 4.23, se logró reducir este sesgo mientras se mejoró la precisión de las cuatro métricas.

A continuación se presentan las Gráficas ROC (Figura 4.11) correspondientes a los conjuntos con radio de desbalance menor a 10, en ellas se puede notar que la curva para el método propuesto (de color negro y línea continua) se ubica por

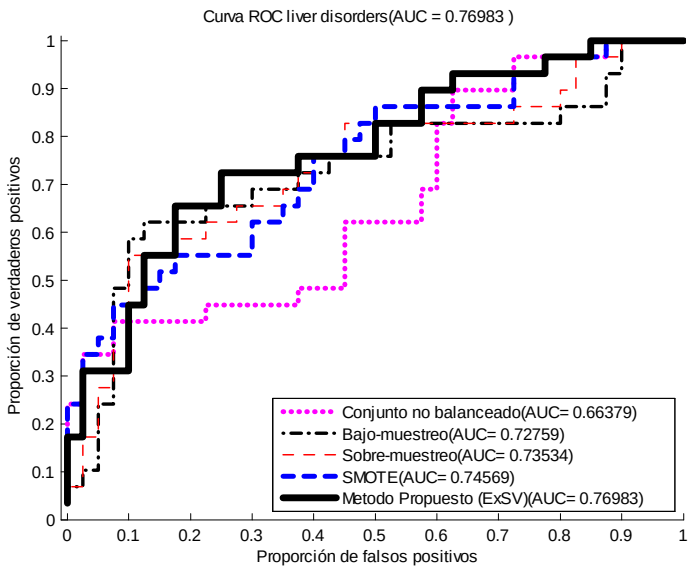
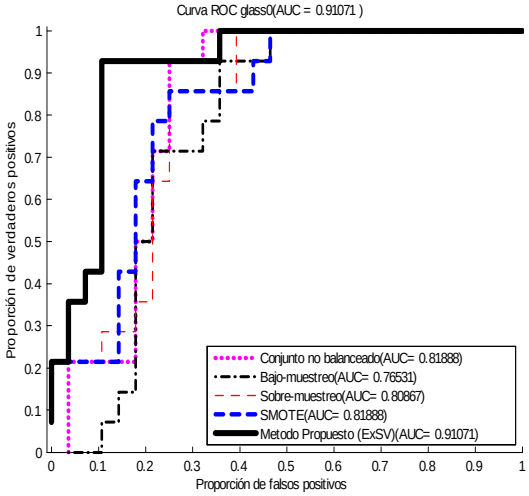


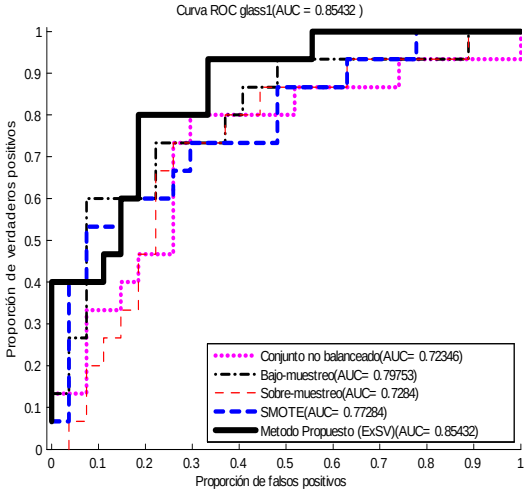
Figura 4.10: Gráfica ROC para el conjunto Liver disorders usando excitación de SV

encima de los demás métodos analizados en la mayoría de los casos.

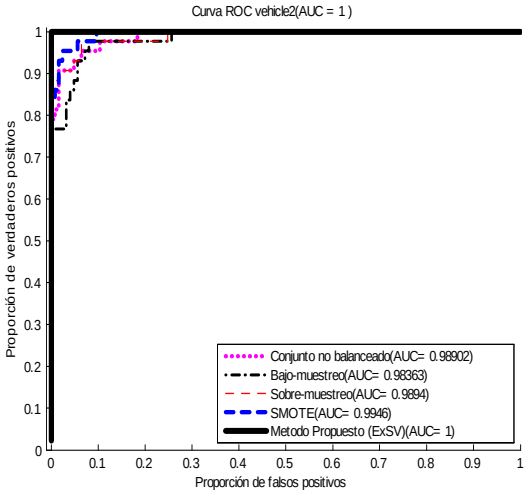
Los conjuntos *glass 2* hasta *yeast 6* tienen un radio de desbalance mayor a 10 y mientras las precisiones presentan un sesgo de clasificación hacia la clase negativa usando el conjunto original, donde alcanzan un valor de 100 % de clasificación para Sn^{False} y Sn^{True} muy bajo en la mayoría de estos conjuntos, con la excitación de SV, como se muestra en la Tabla 4.23, se logró reducir este sesgo mientras se mejoró la precisión de las cuatro métricas, presentando un mejor desempeño respecto a los conjuntos con bajo radio de desbalance. A continuación se presentan las Gráficas ROC (Figura 4.13 y Figura 4.14) correspondientes a los conjuntos con radio de desbalance superior a 10, en ellas se puede notar la curva para el método propuesto, pintada de color negro y línea continua, ubicándose por encima de los demás métodos analizados.



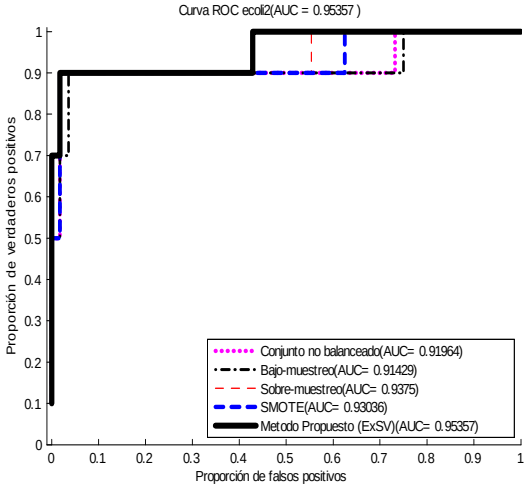
a) Glass 0



b) Glass 1



c) Vehicle 2



d) Ecoli 2

Figura 4.11: Gráficas ROC para los conjuntos con radio de desbalance menor a 10 usando excitación de SV

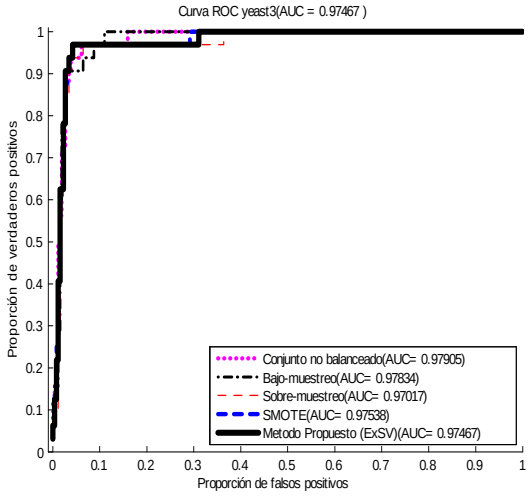
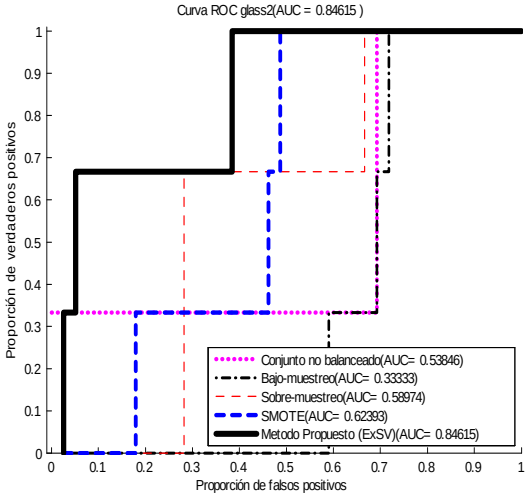


Figura 4.12: Gráficas ROC para el conjunto con radio de desbalance menor a 10, Yeast 3, usando excitación de SV

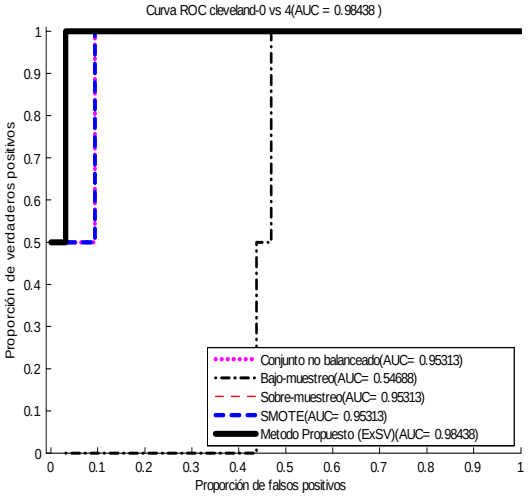
Los conjuntos con mayor cantidad de atributos, por arriba de 10 como lo muestra la Tabla 4.24, mostraron buen desempeño, con precisiones en Sn^{True} y AUC muy cercanas al 100 %, excepto el conjunto vehicle3 con 87 %.

Cuadro 4.24: Precisiones alcanzadas por SVM para los conjuntos con atributos mayores a 10 sobre 10 pruebas usando excitación de SV

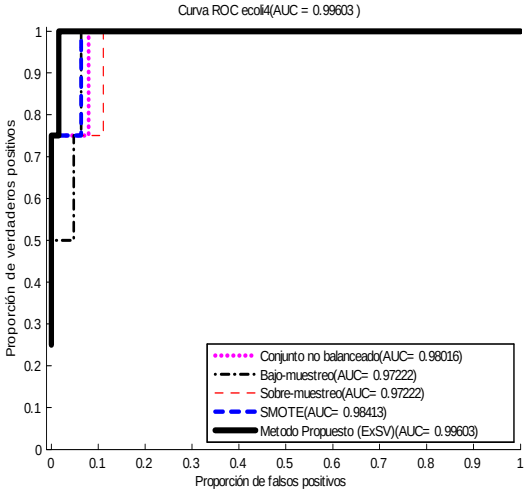
Conjunto	Excitación de SV				
No Balanceado	AUC	Sn	Sn^{True}	Sn^{False}	Atributos
vehicle2	0.99	0.98	0.98	0.97	18
vehicle3	0.85	0.87	0.87	0.77	18
cleveland0_vs_4	0.99	1.00	1.00	0.93	13
pageblocks1-3vs4	0.99	1.00	1.00	0.97	10



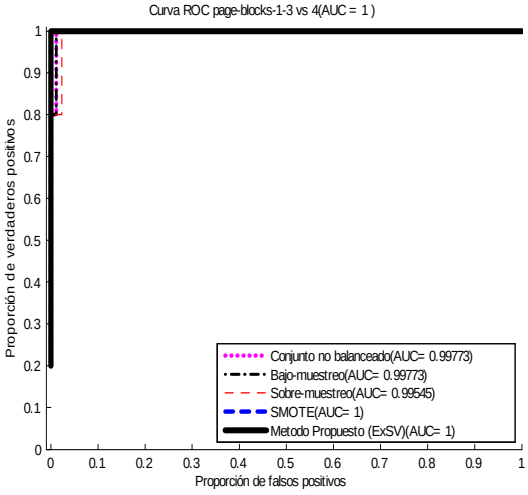
a) Glass 2



b) Cleveland 0 vs 4

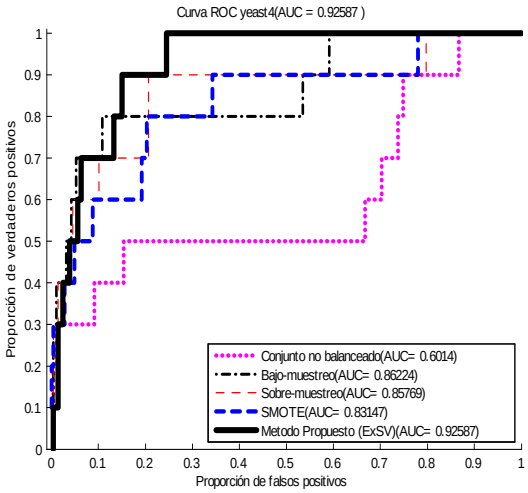


c) Ecoli 4

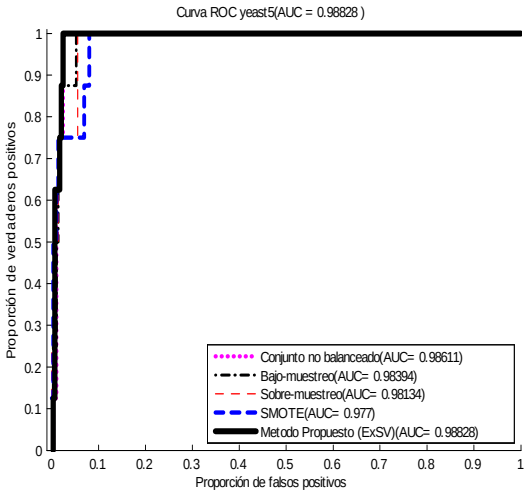


d) Page blocks 1-3 vs 4

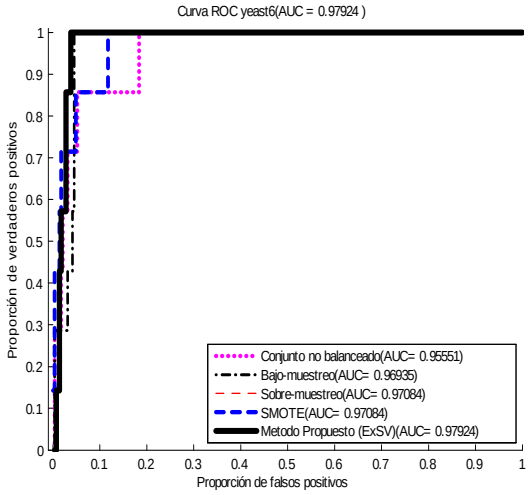
Figura 4.13: Gráficas ROC para los conjuntos con radio de desbalance mayor a 10 usando excitación de SV



e) Yeast 4



f) Yeast 5



g)Yeast 6

Figura 4.14: Gráficas ROC para los conjuntos con radio de desbalance mayor a 10 usando excitación de SV

4.3.5. Desempeño de la SVM usando el Método 2

El segundo método con el que se probaron los conjuntos de datos no balanceados, es la creación de puntos artificiales empleando un GA para mejorar los parámetros. En la Tabla 4.25 se presentan los resultados obtenidos al aplicar el método propuesto sobre los 21 conjuntos de datos no balanceados, las métricas usadas para evaluar la precisión fueron AUC, Sn, Sn^{True} y Sn^{False}, siendo cada valor mostrado en la Tabla, el promedio de 10 pruebas.

El objetivo del método propuesto es mejorar el desempeño de las SVM al balancear el conjunto de entrenamiento, incrementando la cantidad de muestras para la clase minoritaria. Para lograr esto, en cada prueba se anexaron los puntos artificiales creados por el método propuesto en base a la mejor combinación de parámetros obtenidos por el algoritmo genético con codificación GRAY. Posteriormente para mejorar aún más la precisión en cada prueba, fue entrenada la SVM con el nuevo conjunto de entrenamiento y se uso un GA para mejorar sólo los parámetros de la SVM (Kernel RBF, C y gamma).

De acuerdo con los resultados, mostrados en la Tabla 4.25, puede notarse que las precisiones mejoraron notablemente frente al conjunto original y ya no presentan gran sesgo frente a las otras técnicas analizadas, salvo para el conjunto glass0, *glass 1* y vehicle3, aunque en comparación, estas precisiones son mejores frente a las otras técnicas.

Al comparar las precisiones logradas por el método propuesto sobre los conjuntos con radio de desbalance mayor a 10, se puede notar que las precisiones son superiores al 99 %, salvo el último conjunto *yeast 6* con $AUC = 0{,}98$, $Sn = Sn^{True} = 0{,}91$ y $Sn^{False} = 0{,}99$, por lo que la creación de puntos artificiales muestra un mejor desempeño sobre conjuntos con alto radio de desbalance.

Por último, al final de la Tabla 4.25, se presenta la desviación estándar (STD) para cada una de las métricas usadas, notándose que la STD no supera el 0,04 para el AUC, el 0,07 para Sn y Sn^{True}, salvo glass6, mientras que el Sn^{False} tuvo un máximo de 0,07 por lo que sugiere que es un método estable. También se anexa una columna con la cantidad de puntos artificiales creados para lograr la precisión reportada, en algunos casos sólo se requiere de algunos puntos para obtener muy buena precisión como el caso del conjunto *glass 5* que requirió sólo 24 puntos artificiales para lograr una precisión promedio de 1,0 para las cuatro métricas.

Cuadro 4.25: Precisiones alcanzadas por SVM aplicando la creación de puntos artificiales

Conjunto	Con Puntos Artificiales				Desviación estandar				Puntos Artificiales
de Datos	AUC	Sn	Sn^T	Sn^F	AUC	Sn	Sn^T	Sn^F	Etiqueta: +1
liver disorders	0.93	0.89	0.89	0.81	0.03	0.03	0.03	0.07	1326
four class	1.00	1.00	1.00	1.00	0.00	0.00	0.00	0.00	624
glass 1	0.89	0.87	0.87	0.77	0.04	0.05	0.05	0.04	54
diabetes	0.87	0.85	0.85	0.80	0.04	0.05	0.05	0.02	338
glass 0	0.87	0.96	0.96	0.64	0.04	0.05	0.05	0.05	65
vehicle 2	1.00	1.00	1.00	0.98	0.00	0.00	0.00	0.01	500
vehicle 3	0.93	0.93	0.93	0.85	0.02	0.02	0.02	0.03	668
ecoli 1	0.96	0.95	0.95	0.89	0.04	0.05	0.05	0.05	111
ecoli 2	0.97	0.96	0.96	0.96	0.04	0.05	0.05	0.03	48
glass 6	0.98	0.90	0.90	0.99	0.04	0.11	0.11	0.03	38
yeast 3	0.98	0.98	0.98	0.96	0.01	0.02	0.02	0.01	642
ecoli 3	0.97	0.94	0.94	0.93	0.02	0.07	0.07	0.03	135
glass 2	0.99	1.00	1.00	0.97	0.03	0.00	0.00	0.06	84
cleveland-0 vs 4	1.00	1.00	1.00	1.00	0.00	0.00	0.00	0.00	90
glass 4	1.00	1.00	1.00	0.99	0.00	0.00	0.00	0.01	88
ecoli 4	1.00	1.00	1.00	0.99	0.00	0.00	0.00	0.01	108
pageblocks-1-3 vs 4	1.00	1.00	1.00	1.00	0.00	0.00	0.00	0.00	108
glass 5	1.00	1.00	1.00	1.00	0.00	0.00	0.00	0.00	24
yeast 4	0.98	0.99	0.99	0.95	0.02	0.03	0.03	0.04	62
yeast 5	1.00	1.00	1.00	0.99	0.00	0.00	0.00	0.01	238
yeast 6	0.98	0.91	0.91	0.99	0.04	0.07	0.07	0.01	84

Sn^T: Corresponde a la métrica Sn^{True}
Sn^F: Corresponde a la métrica Sn^{False}

A continuación se presentan las gráficas ROC para cada conjunto de datos, cada gráfica corresponde a la precisión promedio alcanzada en las pruebas y se grafican las curvas para cada uno de los cinco métodos analizados.

La primera gráfica corresponde al conjunto *liver disorders*, Figura 4.15 , en esta

se presenta la comparación del método propuesto respecto a las demás técnicas. La diferencia es significativa para el AUC, de 0,93 para la creación de puntos artificiales contra 0,68 para el conjunto original sin embargo, la importancia reside en el hecho de que se mantiene una precisión similar para las métricas restantes (Sn, Sn^{True} y Sn^{False}), de 0,48, 0,48 y 0,85 respectivamente para el conjunto no balanceado frente a la creación de puntos artificiales con 0,89, 0,89 y 0,81 evitando así que la clasificación quede sesgada hacia alguna de las clases.

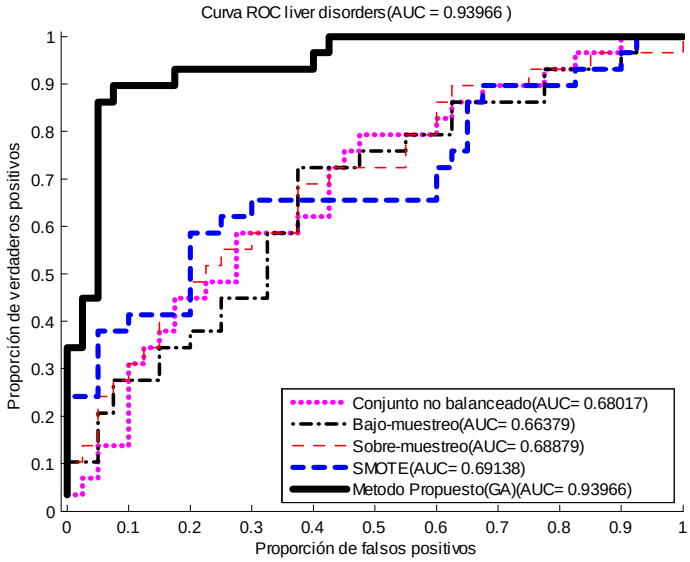
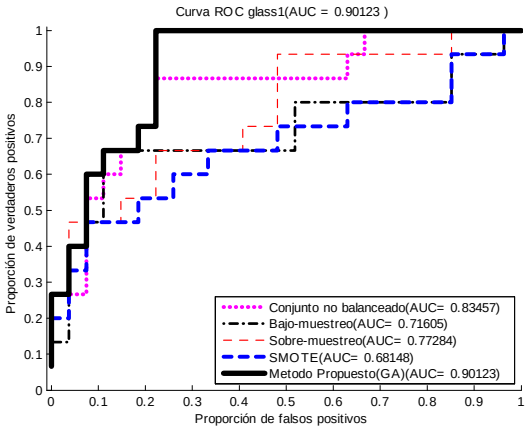


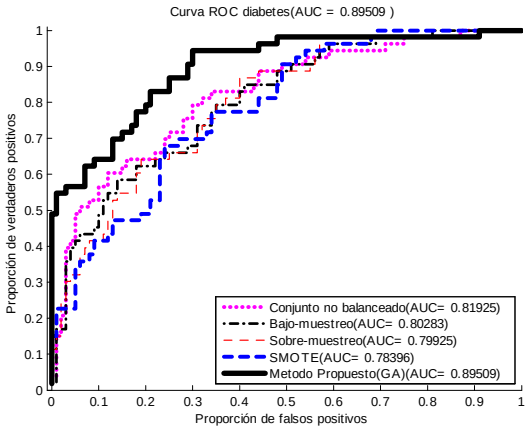
Figura 4.15: Gráfica ROC para el conjunto Liver disorders usando puntos artificiales

Los conjuntos *liver disorders* hasta *ecoli 3* tienen un radio de desbalance menor a 10 y mientras las precisiones presentan un sesgo de clasificación hacia la clase negativa usando el conjunto original, con la creación de puntos artificiales, como se muestra en la Tabla 4.25, se logró reducir este sesgo mientras se mejoró la precisión de las cuatro métricas. A continuación se presentan las Gráficas ROC (Figura 4.16) correspondientes a los conjuntos con radio de desbalance menor a 10, en ellas se

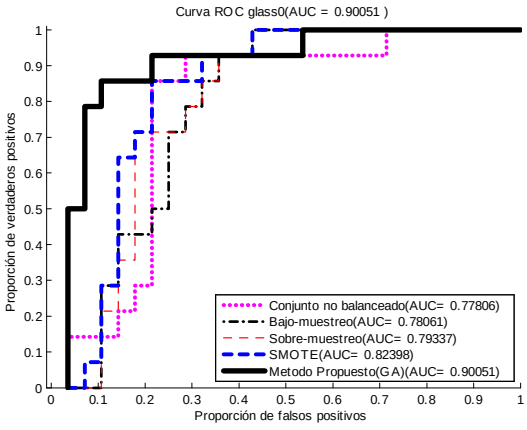
puede notar que la curva para el método propuesto (de color negro y línea continua) se ubica por encima de los demás métodos analizados en la mayoría de los casos.



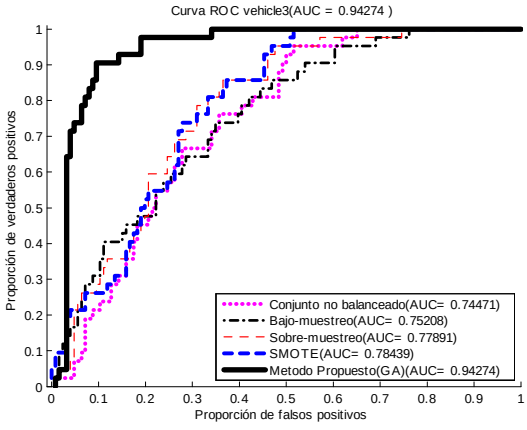
a) Glass1



b) Diabetes



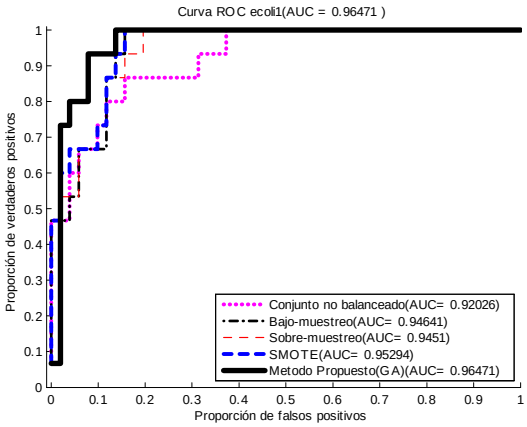
c) Glass0



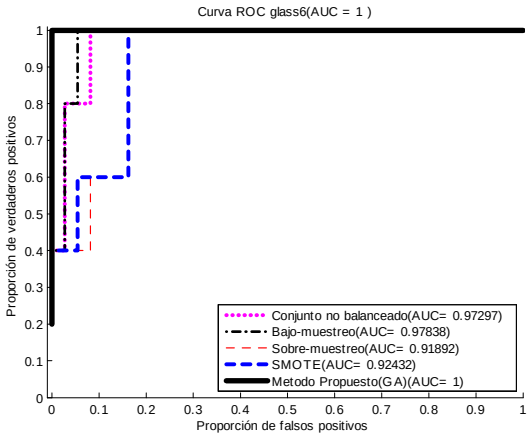
d) Vehicle 3

Figura 4.16: Gráficas ROC para los conjuntos con radio de desbalance menor a 10 usando Puntos Artificiales y un GA

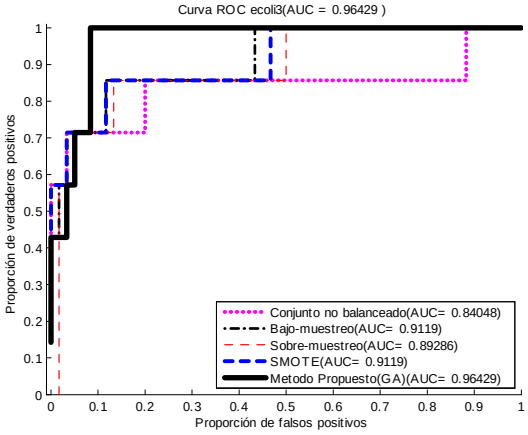
Los conjuntos *glass 2* hasta *yeast 6* tienen un radio de desbalance mayor a 10 y mientras las precisiones presentan un sesgo de clasificación hacia la clase negativa



e) Ecoli 1



f) Glass 6



g)Ecoli 3

Figura 4.17: Gráficas ROC para los conjuntos con radio de desbalance menor a 10 usando Puntos Artificiales y un GA

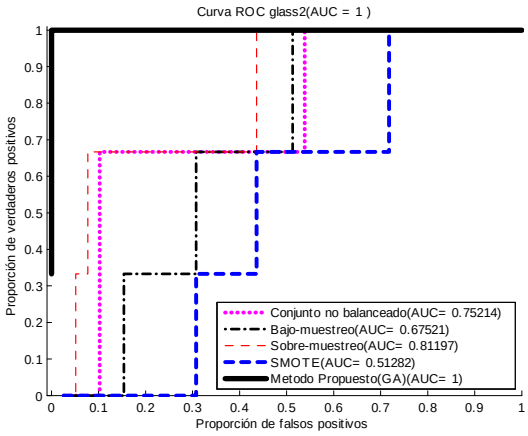
usando el conjunto original, donde alcanzan un valor de 100 % de clasificación para Sn^{False} mientras presentan Sn^{True} muy bajo en la mayoría de estos conjuntos, con la creación de puntos artificiales, como se muestra en la Tabla 4.25, se logró reducir este

sesgo mientras se mejoró la precisión de las cuatro métricas, presentando un mejor desempeño respecto a los conjuntos con bajo radio de desbalance. A continuación se presentan las Gráficas ROC (Figura 4.18) correspondientes a los conjuntos con radio de desbalance superior a 10, en ellas se puede notar que la curva para el método propuesto (de color negro y línea continua) se ubica por encima de los demás métodos analizados en la mayoría de los casos.

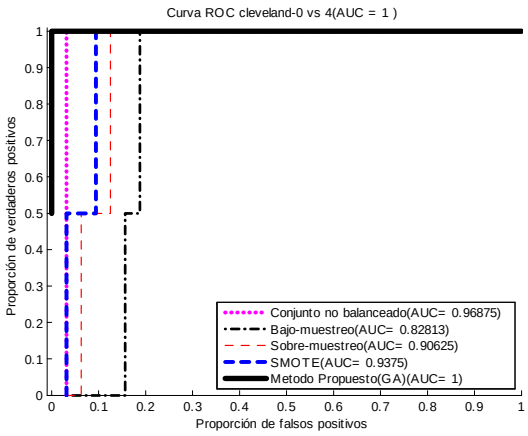
Los conjuntos con mayor cantidad de atributos, por arriba de 10 como lo muestra la Tabla 4.26, mostraron buen desempeño, con precisiones en Sn^{True} y AUC muy cercanas al 100 %, excepto el conjunto vehicle3 con 93 %.

Cuadro 4.26: Precisiones alcanzadas por SVM para los conjuntos con atributos mayores a 10 sobre 10 pruebas usando puntos artificiales

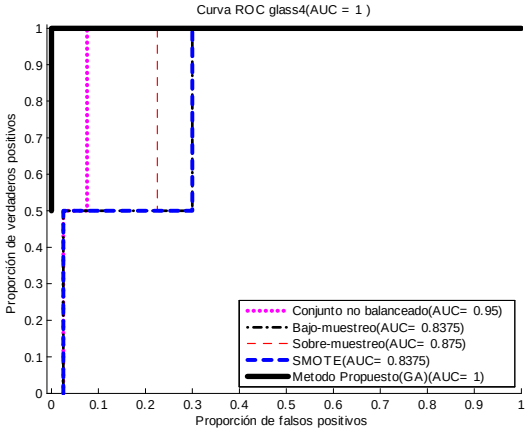
Conjunto	Excitación de SV				
No Balanceado	AUC	Sn	Sn^{True}	Sn^{False}	Atributos
vehicle 2	1.00	1.00	1.00	0.98	18
vehicle 3	0.93	0.93	0.93	0.85	18
cleveland 0 vs 4	1.00	1.00	1.00	1.00	13
page blocks 1-3 vs 4	1.00	1.00	1.00	1.00	10



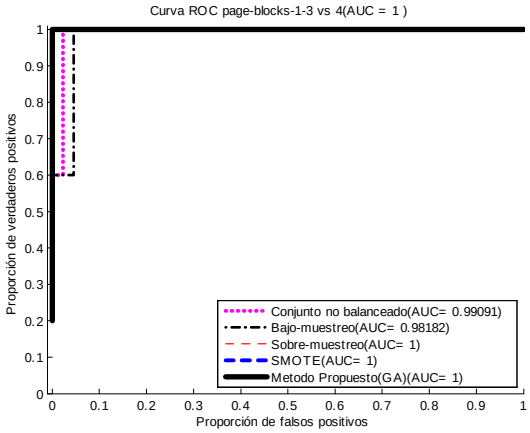
a) Glass 2



b) Cleveland 0 vs 4



c) Glass 4



d) Page blocks 1-3 vs 4

Figura 4.18: Gráficas ROC para los conjuntos con radio de desbalance mayor a 10 usando Puntos Artificiales y un GA

En este capítulo se han presentado los resultados de los algoritmos propuestos sobre 21 conjuntos de datos no balanceados, verificando la precisión de las SVM para detectar patrones positivos, tomando los datos sólo como puntos en el espacio, sin embargo, es necesario presentar una interpretación de estos resultados de acuerdo

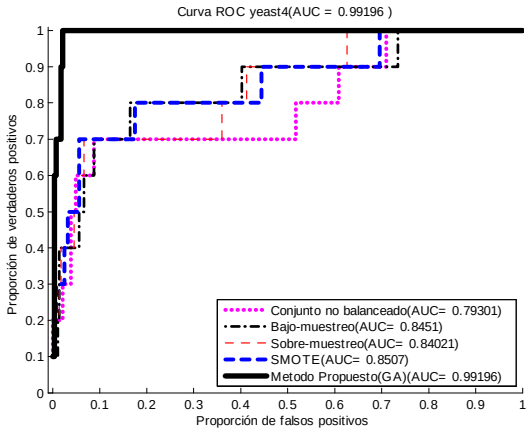


Figura 4.19: Gráficas ROC para el conjunto con radio de desbalance mayor a 10, Yeast 4, usando Puntos Artificiales y un GA

al problema del cual procede cada conjunto y su aplicación en el mundo real. A continuación se presentan los resultados generalizados para cada conjunto de datos.

El conjunto Cleveland, proporcionado por el Centro Médico V.A, Long Beach y la Fundación Clínica de Cleveland, fue elaborado para realizar investigaciones sobre personas con enfermedad del corazón. La detección de personas con este padecimiento requiere de buena precisión para una atención oportuna, sin embargo, al entrenar la SVM con el conjunto original se obtiene una baja precisión (15 %) debido al desbalance entre clases (1 : 12,6), presentando una tendencia a clasificar todas las muestras como negativas (personas sanas), pero al aplicar los algoritmos propuestos se obtuvo una precisión superior al 99 % en ambos, mejorando la detección de estos pacientes enfermos del corazón.

Otro problema médico reside en detectar problemas en el hígado. La investigación médica por parte de BUPA Ltd., publicó el conjunto Liver disorders que contiene muestras de sangre de diferentes personas. Este conjunto presenta poco radio de desbalance (1 : 1,3), sin embargo la SVM reportó una baja precisión

(Sn^{True}) de 48 % para detectar las personas con esta enfermedad. En las pruebas se logró mejorar la precisión, llegando a 89 % usando la creación de puntos artificiales junto con el algoritmo genético. Esta precisión no interfería con las muestras negativas, es decir, mantenía un balance entre la detección de pacientes con y sin problemas en el hígado, evitando el sesgo en la precisión.

El conjunto Diabetes también tiene poco radio de desbalance (1 : 1,86), es decir, por cada paciente con el padecimiento existen 1,86 que no lo presentan. Este conjunto fue creado en la investigación médica realizada por la Universidad de Washington. Inicialmente la SVM reporto una precisión (Sn^{True}) de 56 % para detectar los pacientes con diabetes, pero se logró mejorar usando el segundo algoritmo propuesto, llegando a 0,85 % al entrenar con 338 nuevos puntos artificiales y mejorar los parámetros con ayuda del GA.

Los conjuntos anteriormente mencionados requieren de buena precisión puesto que un error de clasificación representa un alto costo debido a que se trabaja con diagnósticos médicos. Estos errores podrían llevar a realizar pruebas clínicas innecesarias en el caso de presentar un falso positivo o resultar en el deterioro de la salud del paciente si no se le aplica el tratamiento adecuado, que sucede cuando el clasificador presenta un falso negativo. En las pruebas los algoritmos propuestos obtuvieron buen desempeño y lograron obtener buena precisión, reduciendo la cantidad de falsos positivos y falsos negativos, que son importantes en este tipo de aplicaciones médicas.

A continuación se analizan los resultados para otros tipos de aplicaciones en el área de ciencia forense, análisis de texto, clasificación de imágenes y bioinformática.

El Servicio de Ciencia Forense de Estados Unidos de América ha publicado varios conjuntos formados con muestras de 6 tipos de cristales que se pueden encontrar en

una escena de crimen, estos conjuntos se llaman *glass*, cuyo radio de desbalance es de 1 : 1,8 para el conjunto glass 1 y el máximo lo tiene el conjunto glass 5, con radio de 1 : 22. En ambos conjuntos la SVM presentaba una clara tendencia a clasificar todo como negativo, siendo muy pequeña la precisión en Sn^{True} (casi 0) y muy alta en Sn^{False} (casi 1,0). El primer método propuesto logró mejorar la precisión mediante la excitación de SV, reportando 92 % para glass 1 y 99 % para glass 5. Esta mejora en precisión permitirá aportar pruebas más certeras en los casos de crimen.

El conjunto Page blocks cuenta con muestras extraídas de páginas segmentadas de diversos documentos, la aplicación para este conjunto es poder determinar que segmentos corresponden a texto, líneas horizontales o verticales, gráficos e imágenes. El conjunto usado en las pruebas corresponde a una versión modificada donde la clase positiva son solo las líneas verticales y el resto de muestras son negativas, desbalanceando las clases y estableciendo un radio de 1 : 15,8. Este desbalance provocaba que la SVM entrenada con este conjunto presentara una precisión de solo 50 % para la clase positiva aunque el área bajo la curva ROC (AUC) obtuviera un valor de 1,0 (clasificación perfecta). Para solucionarlo, los algoritmos propuestos redujeron el sesgo del hiperplano de separación, llevando la precisión a valores muy cercanos a 100 % en las cuatro métricas (AUC, Sn, Sn^{True} y Sn^{False}). Este tipo de aplicaciones es útil para caracterizar documentos y una buena precisión permitiría obtener mejores resultados y reducir el tiempo de búsqueda al realizar una consulta sobre un gran número de documentos, como lo hacen los motores de búsqueda que operan en Internet.

En clasificación de imágenes, un problema de aplicación particular en el mundo real es determinar, dada una silueta, si esta pertenece a uno de cuatro tipos de vehículo a partir de una muestra formada por las características extraídas de la

silueta. En las pruebas se trato de clasificar las muestras positivas correspondientes a siluetas de un autobús, obteniendo buenos resultados, de 90 %, usando el conjunto vehicle 2 original, pero al tratar de clasificar las siluetas de autos de la marca Opel, la precisión bajo a 13 %, estas muestras correspondían al conjunto vehicle 3, donde las muestras positivas pertenecen a las siluetas de autos de la marca Opel y las muestras negativas corresponden a siluetas de los demás tipos de vehículo. Para mejorar la precisión en vehicle 3, se crearon 668 nuevos puntos artificiales, mejorando el entrenamiento y con ello la precisión de clasificación de la SVM, permitiéndole clasificar autos de la marca Opel con 93 % de precisión y clasificando el resto con 85 % de precisión.

Para las pruebas también se usaron dos conjuntos creados por el investigador Kenta Nakai de la universidad de Osaka, que publicó en 1996. El primer conjunto es Ecoli, el cual ha sido usado en aplicaciones con el objetivo de predecir la localización de sitios de proteína dentro de la célula de la bacteria Escherichia coli. Para los experimentos se utilizaron cuatro diferentes versiones del conjunto, de estos el conjunto con menor radio de desbalance lo tiene ecoli 1 (1 : 3,3), mientras que el conjunto ecoli 4 tiene el mayor radio de desbalance (1 : 15,8). La SVM entrenada con los conjuntos originales de ecoli reportaron una precisión sesgada hacia las muestras negativas, de 96 % para Sn^{False} pero de 69 % para Sn^{True}, es decir, tenía una baja precisión para predecir la ubicación de las proteínas dentro de la bacteria usando el conjunto ecoli 1. Para mejorar esta detección bastó con excitar los SV, llegando a 99 % de precisión. Por otro lado, al entrenar la SVM con el conjunto ecoli 4, se obtenía una baja precisión en Sn^{True} (75 %) con el conjunto original y para mejorarla se crearon 108 nuevos puntos artificiales, llegando a 99 % en la detección de estos sitios de proteína.

El segundo conjunto proporcionado por Kenta, es el conjunto Yeast. Está

integrado por diversos datos acerca de la célula de la levadura y el problema propuesto por el autor es determinar la localización de proteínas dentro de estas células. En las pruebas se usaron cuatro conjuntos, el de menor radio de desbalance (1 : 8,1) corresponde a yeast 3 mientras que yeast 6 presenta el mayor radio de desbalance (1 : 41,4). Con estos dos conjuntos, la SVM obtuvo inicialmente una precisión de 66 % y 0 % respectivamente. La creación de puntos artificiales fue el algoritmo que obtuvo mejor precisión, requiriendo 642 puntos artificiales para obtener una precisión de 98 % con el conjunto yeast 3, mientras que para yeast 6 solo necesitó 84 puntos artificiales para alcanzar 91 % de precisión. Esta mejora en el desempeño permitió clasificar con mayor precisión los sitios donde se ubican las proteínas, ya sea en la membrana de la célula (para el conjunto yeast 3) o en la pared celular (para el conjunto yeast 6).

Capítulo 5

Conclusiones

5.1. Discusión

La teoría indica que las SVM obtienen una buena generalización al maximizar el margen de separación entre los hiperplanos correspondientes a cada clase dentro del espacio de características, sin embargo, cuando se trabaja con conjuntos no balanceados el margen de separación se hace muy grande, por lo que aun obteniendo el margen máximo, no necesariamente se obtendrá buena precisión para clasificar datos con etiquetas positivas. Esto sucede porque el espacio que limita la clase minoritaria, es muy pequeño, frente al espacio limitado por los datos con etiqueta negativa pertenecientes a la clase mayoritaria.

Para disminuir el sesgo del hiperplano de separación de la SVM, se presentaron dos métodos, el primero reduce este sesgo al excitar los SV y el segundo lo reduce al incrementar el tamaño de la clase minoritaria con puntos artificiales. De acuerdo con los resultados, los dos métodos propuestos presentaron una mejora en el desempeño de la SVM con precisiones superiores a las de las otras técnicas analizadas, sobre

todo en los conjuntos con alto radio de desbalance (mayor a 10).

La ventaja de la creación de puntos artificiales es que estos nuevos puntos son creados dentro del espacio de características altamente dimensional, ubicándolos cerca de la frontera entre clases, esto permite disminuir el sesgo del hiperplano de separación al proveer el entrenamiento con más muestras positivos para la clase minoritaria, sin embargo, mientras más puntos artificiales son introducidos, el tiempo de entrenamiento se incrementa. Por otro lado la excitación de SV no crea nuevos puntos, sino que mueve directamente los SV hacia la frontera entre clases y aunque el incremento del desplazamiento es elegido de forma aleatoria (de ahí el nombre de excitación) es esta característica la que mantiene estable el tiempo de entrenamiento.

El desempeño de los métodos propuestos sobre los conjuntos de datos probados es bueno sin embargo, determinar la mejor combinación de parámetros, tanto para la SVM como para los dos algoritmos es una tarea costosa computacionalmente. La solución propuesta fue introducir una búsqueda en malla para los parámetros de la excitación de SV debido a que la cantidad de estos es relativamente corta, mientras que la creación de puntos artificiales requería de una combinación de siete parámetros haciendo muy tardada la implementación de una búsqueda en malla; esta restricción orillo a usar un algoritmo genético para obtener una buena combinación en un tiempo aceptable y lograr mantener una buena precisión.

El Algoritmo Genético requiere de una función de aptitud para evaluar que tan bueno es un individuo como solución potencial del problema respecto al resto de la población. En un principio se tomó la aptitud en función del área bajo la curva (AUC) sin embargo, los conjuntos no balanceados *glass 1*, *glass 2*, *glass 4*, *glass 5*, *yeast 4* y *yeast 6* presentaban una alta precisión para Sn^{False} (muy cercana a 1.0) pero muy baja para Sn y Sn^{True} (casi de 0.0). Esta peculiaridad del AUC

sobre los conjuntos no balanceados provocaba que aún teniendo un AUC máximo (1.0), no se pudiera mejorar la precisión de clasificación para las muestras de la clase minoritaria, que son los más importantes, por lo que la aptitud fue redefinida en función de las cuatro métricas (AUC, Sn, Sn^{True} y Sn^{False}) y se trataron de mejorar todas. La solución óptima, por lo tanto, es alcanzada cuando todas las métricas valen 1,0.

En general, puede decirse que ambos métodos propuestos son estables, ya que la excitación de SV alcanzó una precisión mayor o igual a 93 % en 14 conjuntos para la métrica AUC, en 14 conjuntos para la métrica Sn^{True} y en 12 conjuntos para la métrica Sn^{False} sobre los 21 conjuntos no balanceados analizados, mientras que su desviación estándar sobre la precisión no superó el 0,06 para la métrica AUC, excepto el conjunto *glass 2*; tuvo un máximo de 0,06 para la métrica Sn^{True}, exceptuando los conjuntos *ecoli 2*, *glass 2* y *yeast 6* y por último la métrica Sn^{False} no superó el 0,11, excepto para el conjunto *glass 2*, por lo que los resultados no varían demasiado.

Por otra parte, con la creación de puntos artificiales se alcanzó una precisión mayor o igual a 93 % en 18 conjuntos para la métrica AUC, en 17 conjuntos para la métrica Sn^{True} y en 15 conjuntos para la métrica Sn^{False} sobre los 21 conjuntos no balanceados analizados, mientras que su desviación estándar sobre la precisión no superó el 0,04 para la métrica AUC, tuvo como máximo 0,07 para la métrica Sn^{True} exceptuando el conjunto glass 6 y por último la métrica Sn^{False} no superó el 0,07, estos resultados indican que la precisión de clasificación no varía demasiado.

Ambas técnicas mostraron mejor desempeño sobre conjuntos con desbalance mayor a 10, manteniendo una precisión alta para Sn^{True} y Sn^{False}, a diferencia de las demás técnicas, que presentan claramente una precisión sesgada hacia una de las dos métricas. Sin embargo, la excitación de SV tiene una limitante sobre

la precisión alcanzada, ya que al mantener la cantidad de SV, esto no permite introducir más SV al hiperplano de separación, esta restricción se hace visible en la precisión Sn^{True} que es mejor para la creación de puntos artificiales que para la excitación de SV.

Adicionalmente para evitar un alto costo computacional, cuando el conjunto presenta un gran radio de desbalance, puede aplicarse alguna técnica de selección para reducir solo la clase mayoritaria del conjunto de entrenamiento y el método propuesto obtendrá resultados similares en un tiempo aceptable.

A continuación se presenta una Tabla con los puntos más importantes acerca de los dos métodos presentados en esta Tesis:

Excitación de Vectores Soporte	Creación de Puntos Arrtificiales
Trabaja sobre el espacio de características	Trabaja sobre el espacio de características
Desplaza los SV hacia la frontera con la clase mayoritaria para mejorar la precisión	Toma los SV como referencia para crear nuevos puntos artificialmente, estos nuevos puntos poblaran la clase minoritaria del conjunto de entrenamiento y mejoraran la precisión
No crea nuevos puntos, actualiza la posición de los SV	Crea nuevos puntos, primero dentro de la clase minoritaria y después en la frontera entre clases, estos nuevos puntos serán positivos y se agregan al conjunto de entrenamiento
El desplazamiento elegido para cada dimensión, es aleatorio, por lo que a veces puede no dar buenos resultados, pero puede iterarse hasta obtener una solución aceptable	El desplazamiento es el mismo para todas las dimensiones, regulado por la variable alpha para los puntos dentro de la clase minoritaria y por la variable beta para los puntos fuera de la clase minoritaria o dentro de la frontera
La dirección del desplazamiento es elegido en base al SV de clase mayoritaria (con etiqueta negativa) más cercano al SV de clase minoritaria analizado, de tal forma que el SV positivo se acerque más a la frontera	La dirección del desplazamiento es elegido primero en base a los k SV positivos más cercanos al SV de clase minoritaria analizado y después los m SV negativos más cercanos al SV de clase minoritaria analizado. Esto creara puntos dentro de la clase y fuera de la clase
El tiempo de entrenamiento es menor respecto al segundo método, puesto que trabaja con un subconjunto reducido, los SV	El tiempo de entrenamiento crece conforme se agregan más puntos artificiales a la clase minoritaria
Más de la mitad de los conjuntos no balanceados (KEEL), 14/21, obtuvieron precisiones mayores a 0.95 usando una Malla para mejorar los parámetros. Siendo la menor precisión SnTrue=0.75 y de SnFalse=0.67 para liver disorders	La mayoría de los conjuntos no balanceados (KEEL), 16/21, obtuvieron precisiones mayores a 0.95 usando un Algoritmo Genético para mejorar los parámetros. Siendo la menor precisión SnTrue=0.85 para el conjunto diabetes y de SnFalse=0.64 para glass 0

Cuadro 5.1: Comparación entre la excitación de SV y la creación de puntos artificiales

5.2. Conclusión

Las Máquinas de Vectores Soporte son una herramienta de clasificación que poseen 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 esta Tesis, se presentaron dos nuevos métodos que mejoran el desempeño de las SVM sobre conjuntos no balanceados, el primero reduce el sesgo del hiperplano de separación al excitar los SV^+ y desplazarlos una distancia *épsilon* hacia la frontera de decisión sin perjudicar la precisión; el segundo método reduce el efecto del radio de desbalance al agregar nuevos puntos artificiales positivos al conjunto de entrenamiento basándose en los SV^+.

Estos dos métodos propuestos permiten trabajar con conjuntos no balanceados y al mismo tiempo mejoran significativamente el desempeño de las SVM; son diferentes a otros métodos reportados en la literatura, ya que no trabajan con todo el conjunto original, sino con un grupo reducido y característico llamados Vectores Soporte ya que son los puntos más importantes. Ambos algoritmos incluyen una fase de asignación de etiquetas para los SV, permitiendo distinguir cuales pertenecen a la clase positiva y cuales a la clase negativa, evitando la posible inserción de ruido.

De acuerdo a los resultados, ambos métodos de clasificación propuestos obtienen un desempeño más notable cuando son aplicados sobre conjuntos de datos cuyo radio de desbalance es mayor a 10.

5.3. Trabajo futuro

En esta tesis se presentaron dos nuevos algoritmos para mejorar el desempeño de las SVM, sin embargo, algunos de los pasos metodológicos son susceptibles a mejorarse. A continuación se presentan las posibles rutas de investigación para mejorar los algoritmos propuestos:

1. Las pruebas aplicadas a los métodos propuestos utilizaban los conjuntos de datos con valores normalizados entre cero y uno, sin embargo no se le aplicó una selección de parámetros. La técnica Análisis de Componentes Principales (PCA) puede ayudar a reducir la dimensionalidad de estos conjuntos.

2. Los algoritmos propuestos dependen de una buena selección de parámetros para obtener una buena precisión de clasificación. El primer algoritmo propuesto, que corresponde a la excitación de SV, utiliza una búsqueda en malla, elevando el tiempo de entrenamiento; mientras que el segundo método usa un algoritmo genético para obtener una buena combinación de parámetros en un tiempo razonable. Es esta tarea de búsqueda de parámetros la que deberá mejorarse, sustituyendo la búsqueda en malla por un algoritmo genético, también es necesario mejorar el modelo del GA e intentar con otro tipo de representación para los genotipos, nuevas técnicas de selección de padres, otros operadores de cruza y mutación, así como mejorar la función de aptitud e incluso hacerlo un GA adaptativo.

3. En esta investigación se obtuvieron buenos resultados usando una optimización de parámetros por medio de un algoritmo genético, sin embargo, existen técnicas similares de optimización como Cúmulo de Partículas (*Particle Swarm Optimization*) o Colonia de Hormigas (*Ant Colonies Optimization*)

que podrían dar otro enfoque a los métodos propuestos.

4. Los conjuntos de datos con alto radio de desbalance provocan que el tiempo de entrenamiento crezca de acuerdo al tamaño de la clase mayoritaria, por lo que es necesario realizar una selección de datos que reduzca la cantidad de muestras negativas, sin caer en una selección aleatoria, esto conservará los datos importantes que contribuirán a crear el hiperplano de separación de la SVM. Para realizar esta tarea se requiere de una investigación sobre técnicas de agrupamiento (*clustering*).

5.3.1. Publicaciones

Durante el tiempo de redacción de esta tesis, se presentaron los resultados en los coloquios de investigación organizados por la Universidad Autónoma del Estado de México, con cedes en C.U. Texcoco (2010-B), C.U. Valle de México (2011-A), UAP Tianguistenco (2011-B) y C.U. Ecatepec (2012-A). En estos coloquios se realizaron las exposiciones de los avances de la investigación y se participó en una mesa de trabajo para Inteligencia Artificial, los comentarios y sugerencias aportadas por los demás investigadores sirvieron para mejorar el trabajo.

Los artículos científicos publicados fueron dos, el primero para el método de creación de puntos artificiales y el segundo para la excitación de SV. A continuación se presentan los datos y resumen de cada artículo, para mayor detalle ver el Anexo.

5.3.2. Mejorando la Clasificación de Datos No-Balanceados con SVM Generando Datos Sintéticos

Los resultados del primer método propuesto, fueron publicados en el Congreso Nacional de Computación e Informática (CONACI) en su versión 2011, Tomo 3 y páginas 121 a 130. Los autores fueron José Hernández Santiago, Jair Cervantes y Adrian Trueba Espinoza. A continuación se presenta el resumen del artículo:

Resumen. En los últimos años las SVM han sido una de las técnicas de clasificación extensamente estudiadas, debido a excelentes resultados mostrados en muchos campos de aplicación, la principal ventaja de las SVM, es su poder discriminativo y capacidad de generalización. Sin embargo, estudios recientes muestran que su desempeño es significativamente afectado en conjuntos de datos no-balanceados. La clasificación de datos no-balanceados es un problema crucial en aprendizaje de máquinas, siendo este problema predominante en muchas aplicaciones del mundo real. En tales problemas, la mayoría de las muestras pertenecen a una clase y una minoría a otra clase, que usualmente es la más importante, los clasificadores tradicionales tienden a clasificar todos los datos dentro de la clase mayoritaria, que es la clase menos importante, es por ello la gran necesidad de obtener técnicas de clasificación de datos que mejoren el desempeño de las SVM sobre conjuntos no-balanceados. En este artículo se implementa una nueva técnica para entrenar SVMs con conjuntos de datos no balanceados. La técnica propuesta genera datos artificialmente cercanos a los vectores soporte a partir de una primera etapa de entrenamiento, los resultados muestran que la estrategia propuesta ayuda a mejorar el desempeño de las SVM en la mayoría de los conjuntos probados.

5.3.3. Mejorando el desempeño de las SVM sobre Conjuntos de Datos No-Balanceados mediante la Excitación de Vectores Soporte

El título original de este artículo es "*Enhancing the Performance of SVM on Skewed Data Sets by Exciting Support Vectors*" y fue publicado por la Sociedad Iberoamericana de Inteligencia Artificial (IBERAMIA) en su versión 2012. Los autores fueron José Hernández Santiago, Jair Cervantes Canales, Asdrúbal López-Chau, Farid García Lamont y Lisbeth Rodríguez Masahua. A continuación se presenta el resumen del artículo:

Resumen. El imbalance en conjuntos de datos representa un problema importante en Aprendizaje de Máquinas. El imbalance provoca una baja precisión al generalizar 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 presenta un desempeño pobre debido a que el hiperplano obtenido es sesgado hacia la clase mayoritaria durante la etapa de entrenamiento. Por otro lado, en el mundo real los conjuntos de datos son regularmente no balanceados, es por ello que la clasificación de conjuntos no balanceados es un problema crucial en Aprendizaje de Máquinas. En este artículo, se presenta un nuevo algoritmo para mejorar el desempeño de las SVM en conjuntos de datos no balanceados. El algoritmo propuesto genera vectores artificiales mediante una excitación de los Vectores Soporte. Los resultados experimentales obtenidos en diversos conjuntos de datos no balanceados muestran que el algoritmo propuesto mejora el desempeño de las SVM en la mayoría de las pruebas, obteniendo mejores resultados en conjuntos con alto radio de desbalance.

Bibliografía

Akbani, R., Kwek, S., and Japkowicz, N. (2004). Applying Support Vector Machines to Imbalanced Datasets. In Boulicaut, J.-F., Esposito, F., Giannotti, F., and Pedreschi, D., editors, *XVth European Conference on Machine Learning(ECML04)*, volume 3201 of *Lecture Notes in Computer Science*, pages 39–50, Berlin, Heidelberg. Springer-Verlag.

Arbach, L., Reinhardt, J., Bennett, D., and Fallouh, G. (2003). Mammographic Masses Classification: Comparison between Backpropagation Neural Network (BNN), K Nearest Neighbors (KNN), and Human Readers. In *Electrical and Computer Engineering, 2003. IEEE CCECE 2003. Canadian Conference on*, volume 3, pages 1441–1444 vol.3.

Bazzani, A., Bevilacqua, A., Bollini, D., Brancaccio, R., Campanini, R., Lanconelli, N., Riccardi, A., Romani, D., and Zamboni, G. (2000). Automatic Detection of Clustered Microcalcifications in Digital Mammograms using an SVM Classifier. In *ESANN*, pages 195–200.

Ben-Hur, A. and Weston, J. (2010). A User's Guide to Support Vector Machines Data Mining Techniques for the Life Sciences. volume 609 of *Methods in Molecular Biology*, chapter 13, pages 223–239. Humana Press, Totowa, NJ.

Bobadilla, J. L., Mojica, T., and Niño, L. F. (2003). Identificación de Sitios en Proteínas Usando Máquinas con Vectores de Soporte. *Nova-Publicación Científica*, 1(1):65–71.

Cervantes, J., Li, X., and Yu, W. (2009). Splice Site Detection in DNA Sequences Using a Fast Classification Algorithm. In *Proceedings of the 2009 IEEE International Conference on Systems, Man and Cybernetics*, SMC'09, pages 2683–2688, Piscataway, NJ, USA. IEEE Press.

Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-Sampling Technique. *Journal of Artificial Intelligence Research*, 16:321–357.

Chih-Chia, Y. and Pao-Ta, Y. (2007). Effective Training of Support Vector Machines Using Extractive Support Vector Algorithm. In *Machine Learning and Cybernetics, 2007 International Conference on*, volume 3, pages 1808–1814, Hong Kong.

Dror, G., Sorek, R., and Shamir, R. (2005). Accurate Identification of Alternatively Spliced Exons Using Support Vector Machine. *Bioinformatics*, 21(7):897–901.

Fawcett, T. (2006). An Introduction to ROC Analysis. *Pattern Recogn. Lett.*, 27(8):861–874.

Grzymala-Busse, J. W., Stefanowski, J., and Wilk, S. (2005). A Comparison of Two Approaches to Data Mining from Imbalanced Data. *Journal of Intelligent Manufacturing*, 16:565–573.

Guo, H. (2004). Learning from Imbalanced Data Sets with Boosting and Data Generation: The DataBoost-IM approach. *SIGKDD Explorations*, 6:2004.

Han, H., Wang, W., and Mao, B. (2005). Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning. In *ICIC (1)'05*, pages 878–887.

Hart, P. (1968). The Condensed Nearest Neighbor Rule (Corresp.). *Information Theory, IEEE Transactions on*, 14(3):515 – 516.

Hu, S., Liang, Y., Ma, L., and He, Y. (2009). MSMOTE: Improving Classification Performance when Training Data is Imbalanced. In *2nd International Workshop on Computer Science and Engineering*, pages 13–17, Qingdao, China.

Köknar-Tezel, S. and Latecki, L. J. (2009). Improving SVM Classification on Imbalanced Data Sets in Distance Spaces. In *Proceedings of the 2009 Ninth IEEE International Conference on Data Mining*, ICDM '09, pages 259–267, Washington, DC, USA. IEEE Computer Society.

Kong, W., Tham, L., Wong, K. Y., and Tan, P. (2004). 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*, pages 63–66, Dunedin, New Zealand.

Kononenko, I. (2001). Machine Learning for Medical Diagnosis: History, State of the Art and Perspective. *Artificial Intelligence in Medicine*, 23:89–109.

Kuan-ming, L. and Chih-jen, L. (2003). A Study on Reduced Support Vector Machines. *IEEE TRANSACTIONS ON NEURAL NETWORKS*, 14:1449–1459.

Makal., S. and Ozyilmaz, L. (2007). Determination of splice junctions on DNA by Neural Networks. In *International Symposium on Innovations in Intelligent Systems and Applications*, pages 234–237, Istanbul.

Makal, S., Ozyilmaz, L., and Palavaroglu, S. (2008). Neural Network Based Determination of Splice Junctions by ROC Analysis. *World Academy of Science, Engineering and Technology*, 43:613–615.

Marangoni, F., Barberis, M., and Botta, M. (2003). Large Scale Prediction of Protein Interactions by a SVM-Based Method. In Apolloni, B., Marinaro, M., and Tagliaferri, R., editors, *Neural Nets, 14th Italian Workshop on Neural Nets, WIRN VIETRI 2003, Vietri sul Mare, Italy, June 4-7, 2003, Revised Papers*, volume 2859 of *Lecture Notes in Computer Science*, pages 296–301. Springer.

Nixon, M. and Aguado, A. S. (2008). *Feature Extraction and Image Processing, Second Edition*. Academic Press, 2nd edition.

Peterson, L., Ozen, M., Erdem, H., Amini, A., Gomez, L., Nelson, C., and Ittmann, M. (2005). Artificial Neural Network Analysis of DNA Microarray-based Prostate Cancer Recurrence. In *Computational Intelligence in Bioinformatics and Computational Biology, 2005. CIBCB '05. Proceedings of the 2005 IEEE Symposium on*, pages 1–8.

Platt, J. C. (1998). Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines. Technical Report MSR-TR-98-14.

Provost, F. and Fawcett, T. (2001). Robust Classification for Imprecise Environments. *Mach. Learn.*, 42(3):203–231.

Sebastiani, F. (2002). Machine Learning in Automated Text Categorization. *ACM Comput. Surv.*, 34(1):1–47.

Segovia-Juarez, J. L., Colombano, S., and Kirschner, D. (2007). Identifying DNA Splice Sites using Hypernetworks with Artificial Molecular Evolution. *Biosystems*, 87(2-3):117–24.

Sonnenburg, S., Rätsch, G., and Schölkopf, B. (2005). Large Scale Genomic Sequence SVM Classifiers. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 848–855, New York, NY, USA. ACM.

Sotolongo, G. and Guzmán, M. V. (2001). Aplicaciones de las Redes Neuronales. el Caso de la Bibliometría. *Ciencias de la Información*, 32(1):27–34.

Tan, S. (2005). Neighbor-weighted K-Nearest Neighbor for Unbalanced Text Corpus. *Expert Syst. Appl.*, 28(4):667–671.

Vapnik, V.Ñ. (1995). *The Nature of Statistical Learning Theory*. Springer-Verlag New York, Inc., New York, NY, USA.

Veropoulos, K., Campbell, C., and Cristianini, N. (1999). Controlling the Sensitivity of Support Vector Machines. In *Proceedings of the International Joint Conference on AI*, pages 55–60.

Wolpert, D. H. and Macready, W. G. (1997). No free lunch Theorems for Optimization. *Evolutionary Computation, IEEE Transactions on*, 1(1):67 –82.

Xiaoou, L., Cervantes, J., and Wen, Y. (2010). A Novel SVM Classification Method for Large Data Sets. In *Granular Computing (GrC), 2010 IEEE International Conference on*, pages 297–302.

Zeng, Z.-Q. and Gao, J. (2009). Improving SVM Classification with Imbalance Data Set. In *Proceedings of the 16th International Conference on Neural Information Processing: Part I*, ICONIP '09, pages 389–398, Berlin, Heidelberg. Springer-Verlag.

Zhang, Y., Chu, C.-H., Chen, Y., Zha, H., and Ji, X. (2006). Splice Site Prediction

Using Support Vector Machines with a Bayes Kernel. *Expert Syst. Appl.*, 30(1):73–81.

Zou, S., Huang, Y., Wang, Y., Wang, J., and Zhou, C. (2008). SVM Learning from Imbalanced Data by GA Sampling for Protein Domain Prediction. In *Young Computer Scientists, 2008. ICYCS 2008. The 9th International Conference for*, pages 982–987.

Anexo A

Artículos Publicados

CONACI

CONGRESO NACIONAL DE COMPUTACIÓN E INFORMÁTICA

CIENCIA Y TECNOLOGÍA EN COMPUTACIÓN E INFORMÁTICA

CONACI 2011 TOMO III

UNIVERSIDAD AUTÓNOMA DEL CARMEN
Dependencia de Educación Superior
Área Ciencias de la Información

CIENCIA Y TECNOLOGIA
EN COMPUTACIÓN E INFORMÁTICA

TOMO III

CONACI
CONGRESO NACIONAL DE COMPUTACIÓN E INFORMÁTICA

Colección
Documentos e Investigación

TOMO III

Colección: Documentos e Investigación

Primera edición 2011 por:

Universidad Autónoma del Carmen.
Calle 56 No4 por Av. Concordia, Colonia Benito Juárez
Cd. del Carmen, Campeche, México.

ISBN : En trámite

Esta obra fue realizada en agosto de 2011, en las instalaciones del Centro de Tecnologías de Información, Dependencia de Educación Superior Área Ciencias de la Información de la Universidad Autónoma del Carmen. Cada ejemplar consta de XXX páginas.

HECHO EN MÉXICO/MADE IN MEXICO

Inteligencia Artificial

CONACI
CONGRESO NACIONAL DE COMPUTACIÓN E INFORMÁTICA

Mejorando la Clasificación de Datos No-Balanceados con SVM Generando Datos Sintéticos

José .Hernandez Santiago, Jair Cervantes, Adrian Trueba Espinoza

Posgrado e Investigación
UAEM-Texcoco,
Av. Jardín Zumpango s/n Fraccionamiento El Tejocote, 56259
jose_hernandez_santiago@hotmail.com, chazarra17@gmail.com

Resumen. En los últimos años las SVM han sido una técnica de clasificación extensamente estudiada, debido a excelentes resultados mostrados en muchos campos de aplicación, La principal ventaja de las SVM, es su poder discriminativo y capacidad de generalización. Sin embargo, estudios recientes muestran que su desempeño es significativamente afectado en conjuntos de datos no-balanceados. La clasificación de datos no-balanceados es un problema crucial en aprendizaje de máquinas, este problema es predominante en muchas aplicaciones del mundo real. En tales problemas, la mayoría de los ejemplos pertenecen a una clase y una minoría a otra clase, que usualmente es la más importante, los clasificadores tradicionales tienden a clasificar todos los datos dentro de la clase mayoritaria, que es la clase menos importante, es por ello la gran necesidad de obtener técnicas para clasificación de datos que mejoren el desempeño de las SVM sobre conjuntos no-balanceados. En este artículo implementamos una nueva técnica para entrenar conjuntos de datos no balanceados con SVM. La técnica propuesta genera datos artificialmente cercanos a los vectores soporte a partir de una primera etapa de entrenamiento, la estrategia propuesta ayuda a mejorar el desempeño de las SVM en la mayoría de resultados experimentales.

Introducción

La clasificación en conjuntos de datos desbalanceados ha sido el centro de recientes investigaciones [9] [11] [12] [14]. El desbalanceo de datos es un problema muy importante en muchas aplicaciones de aprendizaje automático y minería de datos. Los efectos de desbalanceo de datos provocan serios efectos negativos en el desempeño de clasificadores. Este problema es predominante en muchas aplicaciones del mundo real como, detección de fraudes, detección de intrusos, diagnosis médico, clasificación de texto, entre otros. En la literatura se pueden encontrar diversos métodos para atacar este problema, sin embargo, el problema de clasificación con datos desbalanceados permanece abierto [2] [4]. A nivel de los datos, una importante dirección en la investigación de conjuntos no-balanceados es la estrategia de muestreo, entre las estrategias propuestas, sobre-muestrear y bajo-muestrear los datos de entrenamiento es una solución muy empleada para resolver esta situación [3] [5]. La primera estrategia sobre-muestrea la *i*-ésima clase hasta que el tamaño de la clase es igual al

tamaño de la clase mayoritaria. En la técnica de bajo-muestreo, la *i*-ésima clase es bajo-muestreada hasta que el tamaño de la *i*-ésima clase es igual al tamaño de la clase minoritaria. Estas técnicas se emplean con el objetivo de balancear la distribución de clases del conjunto de datos. Sin embargo, existe evidencia de que el empleo de estos métodos no tiene un gran efecto sobre el desempeño predictivo de los clasificadores, un ejemplo de ello es el clasificador Bayesiano que es insensitivo a este tipo de técnicas [8]. Algunas técnicas de generación de datos sintéticos, generan datos de la clase minoritaria tomando la diferencia entre los valores de las características de la clase minoritaria y una de sus fronteras más cercanas en la clase minoritaria, multiplicando cada diferencia por un número aleatorio entre 0 y 1, y adicionando estas cantidades al vector de características. Este tipo de técnicas mejoran la precisión de clasificación en general y pueden en ciertos casos mejorar el aprendizaje de algún valor atípico. La desventaja de estas técnicas es que trabajan en un espacio de características, es decir cada ejemplo es representado como un punto en un espacio *n* dimensional, donde *n* el número de características de cada ejemplo. Sin embargo en algunos campos como Bioinformática, visión, análisis de imágenes, los datos son a menudo representados como matrices, donde cada elemento de la matriz es una distancia de similaridad o disimilaridad entre los datos y esta distancia de espacio no satisface los requerimientos de una función métrica [4] [10]. A nivel algorítmico, varios métodos han sido propuestos para atacar este problema, una solución muy popular es asignar pesos mayores a las clases minoritarias que a las clases mayoritarias balanceando la distribución de clases en los datos. Varios clasificadores han sido empleados para clasificación de datos no-balanceados como son clasificadores de vecino cercano, discriminantes lineales de Fisher, redes neuronales y Máquinas de Soporte Vectorial (SVM) [14]. Las SVM han recibido una enorme atención en los últimos años debido a sus buenos fundamentos matemáticos y a una buena habilidad de generalización en conjuntos de datos altamente dimensionales. Además, se ha mostrado que las SVM pueden incorporar métodos de extracción de características convencionales en su arquitectura, mientras que proveen soluciones a problemas inherentes en estos métodos [13]. Sin embargo, cuando los conjuntos de datos en donde el número de ejemplos con etiquetas negativas excede significativamente al de etiquetas positivas, el desempeño de las SVM cae significativamente [2]. En la actualidad se requieren métodos que ataquen el problema que representa entrenar conjuntos de datos no-balanceados, por lo que son necesarios algoritmos de clasificación y resultados que hagan frente a este problema. El presente artículo presenta un algoritmo de clasificación de datos no-balanceados con precisiones comparables y en algunos casos mejores respecto a las precisiones de clasificación obtenidas con los algoritmos actuales.

Preliminares

Máquinas de Vectores Soporte

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

clasificará 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:

$$(x_1, y_1), (x_1, y_1), \ldots, (x_n, y_n) \quad (1)$$

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

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

donde $x = [x_1, x_2, \ldots, x_n]$ 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.

Técnicas de clasificación de datos no-balanceados

A nivel de los datos, las técnicas más importantes empleadas para mejorar el desempeño de los clasificadores son sobre-muestreo (*Over-sampling*), bajo muestreo (*Under-sampling*) y en años recientes una técnica muy popular es SMOTE (*Synthetic minority over-sampling technique*).

Sobre-muestreo (*Over-sampling*)
Esta técnica sobre-muestrea [8][15] la clase minoritaria hasta que su tamaño es igual al tamaño de la clase mayoritaria, balanceando la distribución de clases del conjunto de datos de entrenamiento. Sobre-muestreado es una técnica popular para enfrentar algunos problemas de clasificación no balanceados. Sin embargo, al emplearlo con algunos métodos de clasificación como SVM se incrementa significativamente el tiempo de entrenamiento.

Bajo-muestreo (*Under-sampling*)
Esta técnica bajo muestrea [8][9] la clase mayoritaria balanceando la distribución de clases del conjunto de datos de entrenamiento. Específicamente, la clase mayoritaria es bajo muestreada hasta que el tamaño es igual a la clase minoritaria. Algunos estudios muestran que esta técnica es mejor que sobre-muestreado en algunos casos. También debe hacerse notar que esta técnica reduce el tiempo de entrenamiento, sin embargo puede llegar a eliminar ejemplos de entrenamiento potencialmente útiles degradando el desempeño del clasificador.

SMOTE

Esta técnica [4], agrega puntos dentro de la clase menor del conjunto de entrenamiento. Usa *k* vecinos cercanos al patrón evaluado *x* y lo desplaza un α (una distancia aleatoria entre 0 y 1) de la distancia entre *x* y uno de los *k* vecinos cercanos (elegido aleatoriamente).

Metodología

Algoritmo de Clasificación

La clasificación con datos no-balanceados es uno de los retos recientes en aprendizaje de máquinas. Existen algunas técnicas en la literatura para encontrar una solución a este problema, tales como la aplicación de una etapa de pre-procesamiento enfocada en compensar los datos. Generar nuevos datos de forma inteligente en el conjunto minoritario, incrementa el sesgo del clasificador aprendido hacia este y mejora la precisión sobre las clases minoritarias. En este artículo, utilizamos una nueva técnica de generación de datos, el algoritmo propuesto genera nuevos puntos a partir de los datos más importantes, manteniendo la información más importante en el conjunto de datos de entrenamiento y mejorando la precisión de clasificación.

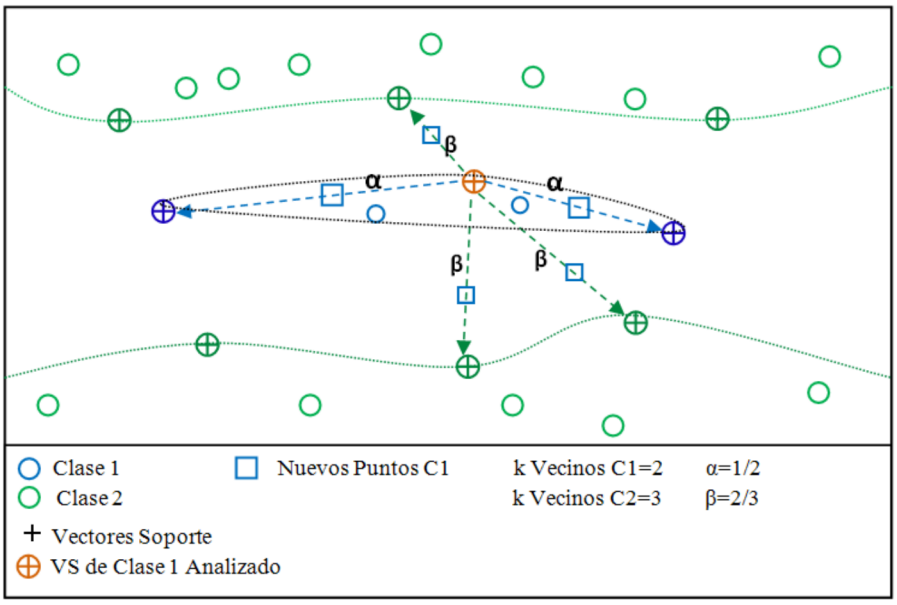


Figura 1. Generación de datos sintéticos a partir de vectores soporte.

La clasificación con SVM permite estimar una función de clasificación $H: X \rightarrow \{\pm 1\}$ al utilizar un conjunto de datos de entrenamiento etiquetados $X \times \{\pm 1\}$ tal que *H* correctamente clasificará ejemplos no vistos (datos de prueba).

Algoritmo 1. Algoritmo de entrenamiento

Input: X_{CNB}

//X_{CNB} ; Conjunto de datos no balanceado

Output: $H_f: \{x_i \in SV's\}$;

Inicio

1. $X_r^+ \leftarrow 0$ /* Conjuntos de entrenamiento con etiquetas positivas inicia vacio*/
2. $X_r^- \leftarrow 0$ /* Conjuntos de entrenamiento con etiquetas negativas inicia vacio*/
3. $X_r^+ \leftarrow \{x_i \in x_{EDS}: y_i = +1\}, i = 1,2, \dots p;$
4. $X_r^- \leftarrow \{x_i \in x_{EDS}: y_i = -1\}, i = 1,2, \dots n;$
5. $H_1 \leftarrow trainSVM(X_r^+, X_r^-);$ /* Obtener hiperplano inicial*/
6. $SV's \leftarrow GetSV(X_r^+, X_r^-);$ /* Obtener vectores soporte*/
7. Generar (X_{sr}^+, X_{sr}^-) usando algoritmo 2 /*Generar datos sintéticos*/
8. Obtener $H_2 \leftarrow trainSVM[(X_r^+, X_r^-) \cup (X_{sr}^+, X_{sr}^-)];$ /* Obtener hiperplano refinado*/
9. Obtener $SV's \leftarrow GetSV[(X_r^+, X_r^-) \cup (X_{sr}^+, X_{sr}^-)];$ /* Obtener vectores soporte*/
10. Obtener $Acc(t) \leftarrow TestingSVMH_2(X_{rt}^+, X_{rt}^-);$ /* Obtener precision de clasificación*/
11. If $Acc(t) - Acc(t-1) > 0$ then
12. Ir a paso 7
13. Else
14. **return** $H_f\ (X_{RD}^+, X_{RD}^-)$

El primer paso del método propuesto consiste en identificar la clase minoritaria la cual contiene *p* ejemplos en el conjunto de datos no-balanceado y etiquetarlas estas como X_r^+ , y en el conjunto de datos restante se usa bajo muestreo para obtener un pequeño conjunto de datos que se etiqueta como negativos X_r^- . X_r^+ y X_r^- son empleados para entrenar una SVM con el objetivo de encontrar un hiperplano preliminar $H_1(X_r^+, X_r^-)$, a partir de este hiperplano H_1 obtenemos los vectores soporte $SV's$. Los vectores soporte son una referencia excelente para generar nuevos datos, ya que los nuevos datos generados estarán dentro de la frontera de cada clase y en algunos casos los datos sintéticos generados mejoraran la precisión de clasificación al generalizar, en nuestro caso empleamos una medida de distancia con los vectores soporte encontrados en la primera fase del algoritmo como lo muestra la Figura 1. Por cada vector soporte encontrado se buscan los dos vecinos más cercanos de la misma clase y se etiquetan. Aunado a estos, se buscan los dos vecinos más cercanos con etiqueta negativa y se generan puntos a una distancia αx_i, donde $\alpha \in [0,1]$, mientras que x_i es la distancia original del vector soporte. La métrica usada para encontrar los *k* vecinos más cercanos, fue la distancia euclidiana.

Algoritmo 2. Generación de puntos

Input: SV_+, SV_-

//SV_+, SV_- ; Vectores soporte de clase positiva y negativa

Output: $Data_{new}(X_{sr}^+, X_{sr}^-)$;

Inicio

1. $Data_{new}(X_{sr}^+, X_{sr}^-) \leftarrow 0$ /* Matriz de nuevos puntos vacia*/
2. $k_{vecinos}\ (SV_+, SV_-) \leftarrow \mathbf{0}$ /* matriz de vecinos cercanos vacia*/
3. Obtener $k_{vecinos}\ (SV_+, SV_-)$; /* Obtener k vecinos cercanos de vectores soporte*/
4. Generar $Data_{new}(X_{sr}^+, X_{sr}^-)$; /*generar nuevos puntos a partir de vecinos cercanos y vectores soporte negativos*/
5. **return** $Data_{new}(X_{sr}^+, X_{sr}^-)$

La principal ventaja del método propuesto es que mejora el desempeño de las SVM en conjuntos de datos no-balanceados reduciendo la influencia de características irrelevantes y redundantes, reteniendo y generando información valiosa en la frontera de decisión.

Resultados experimentales

En esta sección, describimos la metodología empleada y mostramos los resultados obtenidos con el algoritmo propuesto.

Métricas para clasificación de datos no-balanceados

Para evaluar un clasificador sobre grandes conjuntos de datos altamente no-balanceados, es necesario utilizar una métrica adecuada. En conjuntos de datos con una distribución muy sesgada, la métrica de la precisión total no es suficiente. Esto es porque en un conjunto descompensado de 99 a 1, un clasificador que obtiene todos los datos negativos obtendrá una precisión del 99%, pero será completamente inútil como clasificador para detectar los inusuales ejemplos positivos.

La comunidad médica y cada vez más la comunidad de aprendizaje de máquinas emplea dos métricas, la sensitividad y la especificidad para evaluar el desempeño de varias pruebas. La sensitividad es calculada como:

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

S_n es la proporción de lugares candidatos en el conjunto de datos de prueba que han sido correctamente clasificados y este es evaluado como:

$$S_n = \frac{N_C}{N_t} \tag{4}$$

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

$$S_n^{verdadero} = \frac{T_P}{T_P + F_N} \tag{5}$$

donde T_P es el número de patrones de Clase +1 reales los cuales son pronosticados como verdaderos (verdaderos positivos), T_N es el número de patrones de Clase -1 reales que son pronosticadas como falsos (verdaderos negativos), F_P es el número de patrones de clase -1 reales que fueron pronosticados como verdaderos (falsos positivos), F_N es el número de patrones de clase +1 reales que son pronosticados como falsos (falsos negativos), N_C es el número de datos positivos que han sido correctamente pronosticados en el conjunto de datos de prueba y N_t es el número total de datos positivos en el conjunto de datos de prueba

Selección del Modelo

El entrenamiento con SVM involucra el ajuste de varios parámetros, los parámetros seleccionados tienen un crucial efecto sobre el desempeño del clasificador entrenado. En los resultados obtenidos, empleamos una función de base radial (*RBF*) para entrenar la SVM. La función de base radial es definida como

$$K(x_i - x_j) = exp\left(-\gamma \|x_i - x_j\|^2\right), \gamma > 0 \quad (7)$$

El parámetro C regula el punto medio entre error de entrenamiento y complejidad, mientras que γ es un parámetro del kernel. Para obtener los parámetros óptimos empleamos una búsqueda de malla sobre C y γ empleando validación cruzada.

Resultados

Para probar el clasificador propuesto se realizaron diversos experimentos, primero sobre conjuntos de datos no-balanceados artificialmente y después sobre conjuntos de datos empleados por diversos autores, para probar la implementación propuesta para clasificación de datos no-balanceados se emplearon diversos conjuntos de datos. El conjunto de datos empleado para evaluar el desempeño del clasificador es el keel dataset obtenido de http://sci2s.ugr.es/keel/datasets.php, el radio de desbalance varía en cada conjunto de datos empleado. Los resultados son mostrados en la Tabla 1, en ella se muestran las comparaciones con las técnicas de bajo-muestreo, sobre-muestreo y la técnica SMOTE.

Tabla 1. Resultados del método propuesto en conjuntos de datos desbalanceados

Datos	Bajo-muestreo			Sobre-muestreo			SMOTE			Método propuesto		
	s_n	s_n^v	s_n^f	s_n	s_n^v	s_n^f	s_n	s_n^v	s_n^f	s_n	s_n^v	s_n^f
iris	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
new-thyroid	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
ecoli	0.81	0.81	0.92	0.75	0.75	0.94	0.75	0.75	0.94	**0.81**	**0.81**	**0.94**
shuttle	0.91	0.91	0.99	0.91	0.91	0.99	0.87	0.87	0.99	**1.00**	**1.00**	**0.99**
yeast	0.44	0.44	0.90	0.46	0.46	0.90	0.55	0.55	0.88	**0.61**	**0.61**	**0.88**
vowel	1.00	1.00	0.93	1.00	1.00	0.98	1.00	1.00	0.98	1.00	1.00	0.98
page-blocks	0.91	0.91	0.84	0.73	0.73	0.93	0.51	0.51	0.97	**0.58**	**0.58**	**0.98**
segment	0.98	0.98	1.00	0.98	0.98	1.00	0.98	0.98	1.00	0.98	0.98	1.00

En algunos resultados el radio de desbalance es muy pequeño y la precisión obtenida con el método propuesto es igual a la obtenida con otros métodos. Sin embargo, cuando el desbalance es grande, la técnica propuesta mejora los resultados obtenidos con las otras implementaciones. Los parámetro empleados para evaluar el desempeño del método propuesto fueron $\alpha = 0{,}33$, mientras que los parámetros C y γ para entrenar la SVM se utilizaron para todas las implementaciones los mismos, los parámetros para entrenar la SVM se encontraron empleando validación cruzada.

Conclusiones

En este artículo, presentamos un nuevo enfoque de clasificación con SVM para conjuntos de datos no balanceados. Con el objetivo de mejorar la precisión de clasificación de las SVM al entrenar conjuntos de datos desbalanceados, empleamos un algoritmo modificado que genera datos nuevos cercanos a los vectores soporte, superando las desventajas al trabajar con conjuntos de datos no-balanceados. La ventaja más importante de este método es una mejora en el desempeño de las SVM reduciendo de forma significativa la influencia de características irrelevantes y redundantes, el método propuesto además retiene y genera información valiosa en la frontera de decisión de las SVM. Los experimentos realizados en conjuntos de datos reales, muestran que el método propuesto es superior a otras técnicas empleadas en conjuntos de datos no balanceados. Además, las técnicas empleadas para medir el desempeño del clasificador, proporcionan una medida adecuada de la calidad del clasificador.

Referencias

1. Agarwal, S. and Roth, D. “Learning a sparse representation for object detection,” *In European Conference on Computer Vision*, Vol. 4, pp. 113-130,2002.
2. R. Akbani, S. Kwek, and N. Japkowicz, “Applying Support Vector Machines to Imbalanced Datasets,” *In Proc. ECML*, 2004, pp.39-50.
3. N. V. Chawla, K. W. Bowyer, and W. P. Kegelmeyer, “Smote: Synthetic minority oversampling technique,” *Journal of Artificial Intelligence Research*, vol. 16, pp. 321-357, 2002.
4. N. V. Chawla, A. Lazarevic, L. O. Hall, and K. W. Bowyer, “Smoteboost: improving prediction of the minority class in boosting,” *In Proceedings of the Principles of Knowledge Discovery in Databases, PKDD-2003*, 2003, pp. 107-119.
5. H. Han, W. Wang, and B. Mao, “Borderline-smote: A new over-sampling method in imbalanced data sets learning,” *Lecture Notes in Computer Science*, vol. 3644. Springer, 2005, pp. 878-887.
6. P-H.Chen, R-E.Fan and C-J.Lin, “A Study on SMO-Type Decomposition Methods for Support Vector Machines,” *IEEE Trans. Neural Networks*, Vol.17, No.4, 893-908, 2006.
7. N.Cristianini, J.Shawe-Taylor, “An Introduction to Support Vector Machines and Other Kernel-based Learning Methods,” *Cambridge University Press*, 2000.

8. Japkowicz N., “Learning from imbalanced data sets a comparison of various strategies”, *Working Notes of the AAAI00 Workshop Learning from Imbalanced Data Sets*, pp. 10-15,2000
9. Japkowicz N., Stephen S., “The class imbalance problem: a systematic study,” *Intelligent Data Analysis* 6 (5) (2002) 429-450.
10. Suzan Koknar-tezel and Longin Jan Latecki, “Improving SVM Classification on Imbalanced Data Sets in Distance Spaces,” *IEEE International Conference on Data Mining,* 2009, pp 259 -267.
11. Maloof M.A., “Learning when data sets are imbalanced and when costs are unequal and unknown,” in: *Proceedings of Working Notes ICML_03 Workshop Learning from Imbalanced Data Sets*, 2003.
12. Opelt, A., Fussenegger, M., Pinz, A., and Auer, P.; “Weak hypotheses and boosting for generic object detection and recognition,” *In European Conference on Computer Vision,* Vol. 2, pp. 71-84, 2004.
13. Platt J., “Fast Training of support vector machine using sequential minimal optimization.” In A.S.B. Scholkopf, C. Burges, editor, *Advances in Kernel Methods: support vector machine* . MIT Press, Cambridge, MA 1998.
14. Vapnik V., “Statistical Learning Theory,” *Springer, N.Y.*, 1998.
15. F., D. Jensen and T. Oates, “Efficient Progressive Sampling,” *In Proceedings of the Fifth International Conference on Knowledge Discovery and Data Mining.1999*

Enhancing the Performance of SVM on Skewed Data Sets by Exciting Support Vectors

José Hernández Santiago[1], Jair Cervantes[1], Asdrúbal López-Chau[2], and Farid García Lamont[1]

[1] Posgrado e Investigación, UAEM-Texcoco, Av. Jardín Zumpango s/n Fraccionamiento El Tejocote, Edo. Mex., C.P. 56259, chazarra17@gmail.com
[2] UAEM, Centro Universitario UAEM Zumpango, Camino viejo a Jilotzingo continuación calle Rayón, Valle Hermoso Zumpango, México, C.P. 55600, alchau@uaemex.mx

Abstract. In pattern recognition and data mining a data set is named skewed or imbalanced if it contains a large number of objects of certain type and a very small number of objects of the opposite type. The imbalance in data sets represents a challenging problem for most classification methods, this is because the generalization power achieved for classic classifiers is not good for skewed data sets. Many real data sets are imbalanced, so the development of new methods to face this problem is necessary. The SVM classifier has an exceptional performance for data sets that are not skewed, however for imbalanced sets the optimal separating hyper plane is not enough to achieve acceptable results. In this paper a novel method that improves the performance of SVM for skewed data sets is presented. The proposed method works by exciting the support vectors and displacing the separating hyper plane towards majority class. According to the results obtained in experiments with different skewed data sets, the method enhances not only the accuracy but also the sensitivity of SVM classifier on this kind of data sets.

Keywords: SVM, skewed data sets, imbalanced data sets, SMOTE

1 Introduction

A data set is said skewed or imbalanced whether it contains a large number of objects of certain type (majority class) and a very small number of objects of the opposite type (minority class). There are many real world applications that present a remarkable imbalance in their training data sets, for example in fraud detection problems, the imbalance ratio can be from 100 to 1 up to 100,000 to 1 [12], another examples are the classification of protein sequences [3][5][17], medical diagnosis [10], intrusion detection and text classification [13][14]. Recent experiments [1][18][9] show that the performance of most classification methods is affected when they are applied on skewed data sets, this is more evident when the imbalance ratio is large. The imbalance of data sets considerably affects the performance of most classifiers, because in general they are designed to reduce

the global mean error regardless classes distribution and therefore the decision boundaries are biased to the majority class in the training phase. Support Vector Machine (SVM) classifier is currently one of the most important classification techniques[5], it achieves better classification accuracy over other methods such as artificial neural networks [2], decision trees and Bayesian classifiers[3][19] in some applications. The generalization power of SVM is one of its most remarkable characteristics, the reason of the excellent generalization can be explained by the statistic learning theory [16] related to maximum margin separating hyper planes. In spite of maximum margin, in the case of skewed data sets the slanting to majority class of decision boundary (separating hyper plane) negatively impacts in achieved accuracy because minority class can be considered as noise and therefore ignored by classifier. Sensitivity and specificity are measures commonly used to detect this. The development of new techniques to enhance the performance of classifiers such as SVM to be applied on skewed data sets is an important challenge in the areas of pattern recognition, data mining and learning machines. Some authors have proposed methods to reduce the negative effect of imbalance of data sets. Under sampling and over sampling methods intent to balance the data sets by randomly selecting a small number of objects from majority class and taking all or doubling the objects from minority class [1]. A drawback with this is approach is that if objects that are support vectors (SV-objects that define the separating hyper plane) are removed then the separating hyper plane is different from optimal one and accuracy, sensibility and sensitivity are damaged. In addition doubling the number of objects in minority class increases training time of SVM, whose complexity is about O(n2) [3]. Chawla et al [4] proposed Synthetic Minority Over sampling Technique (SMOTE) that generates artificial objects to be included as members of the minority class. SMOTE takes an object from minority class and produces a new version of it by multiplying each feature of original object times a random number between 0 an 1 and adding up this result to the original features. SMOTE does not include a data selection step to recover more important objects from data set. According to the results presented in [4] the technique is better than under sampling and over sampling. Applying SMOTE along with over sampling was proposed in [1]. That method also introduces a scheme to penalize errors depending on the majority (decrease cost) or minority class (increase cost), such combination makes denser the distribution of minority class and puts closer the separating hyper plane from the majority class. In [17] different penalization criteria are used to produce similar effect in separating hyper plane. In [18] a kernelized version of SMOTE is proposed. Some other proposals inspired in SMOTE can be seen in [11][8][7]. In [9] an algorithm to populate the minority class is proposed. The new objects are generated by computing the similarity and dissimilarity among pairs of objects. Genetic algorithms have been used to face the problem of classification on skewed data sets. In [20] a genetic algorithm is used to balance skewed data sets, the method produces better results than simple random sampling.

In this paper a new sampling technique to enhance the performance of SVM on skewed data sets is presented. This novel method differs from previous mainly

in that there is no necessity of changing original SVM formulation, add new penalization schemes or just randomly add up new artificial objects to data set, which can lead to unrepeatable experiments. In the proposed method a SVM is first trained using whole training set in order to find the SV. These SV are then "excited" to be forced to moved forward majority class, after that a few examples are added up to data set close to decision boundary to improve classification accuracy. This approach produces consistent results because examples are not put on arbitrary locations, but always close to decision boundaries.

The results presented in this paper show that accuracy obtained is improved, also the sensitivity and specificity of classifier is enhanced.

The rest of the paper is organized as follows. In section 2 a brief overview on SVM and on metrics for testing classifiers on skewed data sets is presented. Section 3 presents the proposed method. The results of experiments are shown in Section 4. Discussion and Conclusions are in section 5 and 6 respectively. The references are in the last part of this paper.

2 Preliminaries

2.1 Support Vector Machines

SVM are inspired on statistical learning theory developed by Vapnik on 70's [15]. This classifier is one of the most effective methods for complex binary classification problems, so it has been applied in many different fields. The training of SVM begins with a training set X_{tr} given as (1):

$$X_{tr} = \{(x_i, y_i)\}_{i=1}^{n} \tag{1}$$

with $x_i \in R^d$ and $y_i \in R\{+1, -1\}$. The classification function is determined as (2)

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

where α_i are the Lagrange multipliers, $K\left\langle x_i \cdot x_j \right\rangle$ is the kernel matrix, and b is the bias. Deeper details can be found in [15].

2.2 Metrics for testing classifiers on Skewed Data Sets

Accuracy is most time the measurement used to evaluate and to compare a classifier method against other ones. For the special case of skewed data sets, using only accuracy as a metric for evaluating a classifier can lead to wrong conclusions, because minority class has a pretty small impact on accuracy compared with majority class. Consider for example a data set presenting an imbalance ratio of 99 to 1. A classifier that achieves 99% of accuracy is considered good for the general case, however for the skewed example such classifier is not useful.

In order to evaluate a classifier on large and skewed data sets, it is necessary to use a different metric. Medical and machine learning communities use more and more the sensitivity and specificity to evaluate the performance. Sensitivity is computed with (3)

$$S_n^{true} = \frac{T_P}{T_P + F_N} \tag{3}$$

and specificity is computed with (4).

$$S_n^{false} = \frac{T_N}{T_N + F_P} \tag{4}$$

With

T_P is the number of objects (true class +1) that have been predicted as class +1.

T_N is the number of objects (true class) that have been predicted as -1.

F_P is the number of objects (true class -1) that have been predicted as class +1.

F_N is the number of objects (true class +1) that have been predicted as class -1.

Sensitivity is the proportion of positive examples that are correctly identified, whereas the specificity is the proportion of negative examples that are correctly identified.

In addition to these numeric performance metrics mentioned above, the area under the ROC curve (AUC),is also used in this paper. Receiver Operating Characteristic (ROC) analysis is a widely used method for analyzing the performance of binary classifiers. The area under the ROC curve represents how separable two objects are.

A ROC curve can be generated using the labels of the input data set and the classifier output. A detail description on how to plot a ROC curve also can be found in [6].

The most important advantage of ROC analysis is that it is not necessary to specify the misclassification costs. The visual and numeric metrics associated with this method allow for great flexibility in performance analysis.

3 Proposed Method

In order to reduce the effect of imbalance in data sets, we propose to excite the SV that belong to minority class, and force the decision boundary to move forward majority class. The main advantage of the proposed method is that the performance of SVM is enhanced. New points are generated taking as a guide the optimal separating hyperplane and added close to it. This is different to other methods such as SMOTE, where some points are just randomly generated and added up to training set without considering the decision boundary.

The steps of the method can be stated as follows.

1. Separate majority and minority class examples. The examples in minority class form a partition called X_r^+, and the rest of the examples form the majority class partition X_r^-.
2. Under sample. Some objects from the majority class (X_r^-) are randomly selected, the subset is renamed as X_r^-.
3. Train SVM. A SVM is trained using only X_r^+ and X_r^-, the idea in this step is to identify the SV.
4. Excite SV. Once the support vectors have been identified, those that belong to minority class are excited by moving them forward majority class.

The direction of movement is chosen according to (5) and (6)

$$\nu_i = \frac{x_{svi}^+ - x_{ij}^-}{\left\| x_{ij}^- - x_{svi}^+ \right\|_2},\ i = 1, ..., \left| X_r^- \right| \tag{5}$$

with

$$x_{ij}^- = \min_j \left\| x_{svi} - x_{svj} \right\|_2,\ j = 1, ..., \left| X_r^+ \right| \tag{6}$$

where
$x_{svi}^- \in SV$ of X_r^-
$x_{svj}^+ \in SV$ of X_r^+
The SV of minority class are then moved forward majority class using (7). The step size of the movement is ϵ, the values of ϵ are between 1×10^{-3} and 1×10^{-6}.

$$x_{svi} = x_{svi} + \epsilon \cdot \nu_i \tag{7}$$

This displacement of SV belonging to minority class moves the decision boundary towards majority class improving the classification accuracy, sensitivity and sensibility.

The algorithm 1, represents the entire process followed by our proposal:

In the Algorithm $Data_{new}(X_{sr}^+, X_{sr}^-)$ represent the data points created from the first hyperplane using the ecs (6)(7) and (8). H_2 represents the tunned hyperplane obtained with the data points created and original data points.

4 Experiments

In this section the results the proposed method on skewed data sets are presented.

Training a SVM involves the choosing of some parameters. Such parameters have an important effect on the performance of classifier. In all the experiments we use the radial basis function (RBF) as kernel, this function is defined in (8).

$$K(x_i - x_j) = e^{(\gamma \| x_i - x_j \|)}, \gamma > 0 \tag{8}$$

Cross validation and grid search was used to find parameters in (8) and also for computing the regularization parameter of SVM. We use model selection

Algorithm 1 Training Algorithm

Input

X : A skewed data set

Output

$H_f : \{x_i \in SV\}$

Begin

$X_r^+ \leftarrow 0$ // Training set with positive labels starts empty

$X_r^- \leftarrow 0$// Training set with negative labels starts empty

$X_r^+ \leftarrow \{x_i \in X : y_i = +1\}$, i=1,...,p

$X_r^+ \leftarrow \{x_i \in X : y_i = +1\}$, i=1,...,n

$SV \leftarrow getSV(X_r^+, X_r^-)$ //Obtain the SV

repeat

Get support vectors SV_+, SV_- from H_1

Move SV according to (7)

Create $Data_{new}(X_{sr}^+, X_{sr}^-)$ from $SV_+, SV_- \in H_1$

$H_2 \leftarrow$ trainSVM with $X_r^+, X_r^- \cup X_{sr}^+, X_{sr}^-$ //Compute tunning hyper plane

$SV \leftarrow getSV(X_r^+, X_r^- \cup X_{sr}^+, X_{sr}^-)$ //Obtain SV

Acc(t)$\leftarrow TestSVMH_2(X_{rt}^+, X_{tr}^-)$ //Test accuracy

while (Acc(t)-Acc(t-1) > 0)

return $H_f(X_{RD}^+, X_{RD}^-)$

End

Table 1. Data sets

Data set	Cm(+1)	CM(-1)	Dim	Imbalance ratio
australian	307	383	14	1:1.248
diabetes	268	500	8	1:1.866
german_numer	300	700	24	1:2.333
yeast1	429	1,055	8	1:2.459
vehicle0	199	647	18	1:3.251
ecoli1	77	259	7	1:3.364
new-thyroid1	35	180	5	1:5.143
ecoli2	52	284	7	1:5.462
segment0	329	1,979	19	1:6.015
glass6	29	185	9	1:6.379
yeast3	163	1,321	8	1:8.104
page-blocks0	559	4,913	10	1:8.789
cleveland-0_vs_4	13	164	13	1:12.615
shuttle-c0-vs-c4	123	1,706	9	1:13.870
p-blocks-1-3_vs_4	28	444	10	1:15.857
shuttle-c2-vs-c4	6	123	9	1:20.500
glass5	9	205	9	1:22.778
yeast4	51	1,433	8	1:28.098
yeast5	44	1440	8	1:32.727
yeast6	35	1,449	8	1:41.400

to get the optimal parameters. The hyper-parameter space is explored on a two dimensional grid with $\gamma = \left[10^{-2}, 10^{-1}, 10^{0}, 10^{1}\right]$ and the regularization parameter $C = \left[10^{0}, 10^{1}, 10^{2}, 10^{3}, 10^{4}\right]$. In the experiments all data sets were normalized and the 10 fold cross validation method was applied for the measurements. A number of 30 runs were executed in each experiment. For creating the training set and testing sets, the 80% and 20% of elements of each data set were randomly selected respectively.

4.1 Datasets

The KEEL data sets are imbalanced ones (Public available at http:// sci2s.ugr.es/ keel/datasets.php). Table 1 shows the datasets used in the experiments. In order to measure the performance of the proposed method in different scenarios, the data sets chosen have an imbalance ratio from 1 to 1.248 up to 1 to 41.4.

Table 2. Performance of the proposed method

Data set	(**average**)				(**std dev**)			
	AUC	S_n	S_n^v	S_n^f — AUC	S_n	S_n^v	S_n^f
australian	0.897	0.941	0.941	0.783	0.024	0.021	0.021	0.049
diabetes	0.847	0.857	0.857	0.730	0.024	0.034	0.034	0.018
german_numer	0.686	0.880	0.880	0.602	0.028	0.040	0.040	0.048
yeast1	0.797	0.773	0.773	0.722	0.018	0.019	0.019	0.029
vehicle0	0.981	0.982	0.982	0.938	0.009	0.012	0.012	0.021
ecoli1	0.959	0.987	0.987	0.869	0.023	0.028	0.028	0.029
new-thyroid1	0.998	1.000	1.000	0.989	0.004	0.000	0.000	0.014
ecoli2	0.959	0.890	0.890	0.954	0.035	0.099	0.099	0.019
segment0	1.000	0.995	0.995	1.000	0.000	0.007	0.007	0.000
glass6	0.982	0.940	0.940	1.000	0.030	0.097	0.097	0.000
yeast3	0.978	0.969	0.969	0.939	0.009	0.021	0.021	0.013
page-blocks0	0.973	0.852	0.852	0.976	0.010	0.031	0.031	0.007
cleveland-0_vs_4	1.000	0.950	0.950	0.994	0.000	0.158	0.158	0.020
shuttle-c0-vs-c4	1.000	1.000	1.000	1.000	0.000	0.000	0.000	0.000
page-blocks-1-3_vs_4	1.000	1.000	1.000	1.000	0.000	0.000	0.000	0.000
shuttle-c2-vs-c4	1.000	1.000	1.000	1.000	0.000	0.000	0.000	0.000
yeast4	0.953	0.710	0.710	0.975	0.034	0.057	0.057	0.007
yeast5	0.994	0.900	0.900	0.990	0.007	0.115	0.115	0.006
yeast6	0.981	0.857	0.857	0.975	0.012	0.095	0.095	0.012

In Table 2 the reader can observe the full test results, the standard deviations for S_n^{false}, S_n^{true} and AUC were included to compare against other methods shown in table 3.

In order to compare the results of the proposed method, the Table 3 shows the results achieved by the following methods.

– SVM (first column of Table 3).

– Under sampling method (second column).
– over sampling method (third column)
– SMOTE algorithm (fourth column).

It is notable that when imbalance ratio is large, the performance achieved by the proposed method is much better that the obtained by using traditional SVM or the other methods.

5 Discussion

Many works on imbalanced classification use ROC-Curves in order to show the performance of the proposed algorithms. However, in some cases ROC curves are not sufficient to evaluate the performance of a classifier on skewed datasets, because it is possible to achieve a good AUR — S_n^t with a regular classifier.

In our experiments we use four evaluation metrics to show the performance of proposed method. Due to the nature of the SVM, the decision surface relies on the positive/negative support vectors, hence SVM is less sensitive to the statistical prosperities of the features. In this way, to create new data points can be unfavorable, because in some extreme cases, a single misclassified example of the minority class can create a significant drop in the classifier performance. in order face this disadvantage, the proposed method only use the data points created when the performance is improved.

It can be seen in the tables of results that the improvement on the classifier performance is better when the imbalance radio is large. In some datasets with small imbalance radio, there is not improvement in the performance. In datasets with large imbalance radio the threshold can be flexibly set, the goodness of the decision surface learned from the training data determines the classification accuracy. In our experiments, we showed that the SVM could learn a good decision surface generating data points along of the decision surface.

Table 3. Comparative against other state of the art methods

Data Set	SVM				Under sampling				Oversampling				SMOTE			
	AUC	S_n	S_n^T	S_n^F	AUC	S_n	S_n^T	S_n^F	AUC	S_n	S_n^T	S_n^F	AUC	S_n	S_n^T	S_n^F
australian	0.89	0.73	0.73	0.88	0.89	0.81	0.81	0.85	0.88	0.75	0.75	0.87	0.87	0.93	0.93	0.67
glass6	0.98	0.86	0.86	0.99	0.97	0.92	0.92	0.90	0.97	0.90	0.90	0.99	0.98	0.92	0.92	0.99
yeast3	0.98	0.69	0.69	0.98	0.97	0.90	0.90	0.93	0.98	0.92	0.92	0.94	0.98	0.85	0.85	0.96
page-blocks0	0.96	0.53	0.53	1.00	0.95	0.82	0.82	0.94	0.97	0.83	0.83	0.96	0.97	0.66	0.66	0.98
cleveland-0_vs_4	0.98	0.20	0.20	1.00	0.90	0.80	0.80	0.78	0.97	0.40	0.40	0.99	0.97	0.50	0.50	0.99
page-blocks-1-3_vs_4	1.00	0.50	0.50	1.00	0.98	0.96	0.96	0.89	1.00	0.90	0.90	0.98	1.00	0.94	0.94	1.00
shuttle-c2-vs-c4	1.00	0.90	0.90	1.00	1.00	1.00	1.00	0.95	1.00	0.90	0.90	1.00	1.00	1.00	1.00	1.00
glass5	0.99	0.00	0.00	1.00	0.91	1.00	1.00	0.79	1.00	1.00	1.00	0.93	1.00	1.00	1.00	0.98
yeast4	0.75	0.00	0.00	1.00	0.89	0.81	0.81	0.88	0.84	0.34	0.34	0.97	0.89	0.52	0.52	0.98
yeast5	0.99	0.10	0.10	1.00	0.99	1.00	1.00	0.91	0.99	0.64	0.64	0.99	0.99	0.85	0.85	0.98
yeast6	0.92	0.00	0.00	1.00	0.94	0.89	0.89	0.89	0.94	0.74	0.74	0.97	0.95	0.73	0.73	0.97

6 Conclusions

Current classification methods produce good results when they are applied on data sets that are balanced, however for the specific case of skewed data sets most classifiers cannot obtain acceptable results because decision boundaries are computed regardless minority and majority class.

In this paper a novel method that enhances the performance of SVM for skewed data sets was presented. The method reduces the effect of imbalance ratio by exciting SV and moving separating hyper plane toward majority class. The method is different from other state of the art methods in that it does not simply adds artificial objects to training sets, instead the new objects are added close to optimal separating hyperplane, which has the effect of dramatically increasing the performance of SVM on skewed data sets.

According to the experiments, the proposed method produces the most noticeable results when the imbalance ration if greater than 10.

References

1. R. Akbani, S. Kwek, and N. Japkowicz, "Applying support vector machines to imbalanced datasets," in *In Proceedings of the 15th European Conference on Machine Learning (ECML*, 2004, pp. 39–50.
2. R. J. B. D. F. G. Arbach, L., "Mammographic masses classification: Comparison between backpropagation neural network (bnn), k nearest neighbors (knn), and human readers." in *In: Canadian Conference on Electrical and Computer Engineering, IEEE Press, New York*, oct. 2003, pp. 1441 –1444.
3. J. Cervantes, X. Li, and W. Yu, "Splice site detection in dna sequences using a fast classification algorithm," in *Proceedings of the 2009 IEEE international conference on Systems, Man and Cybernetics*, ser. SMC'09. Piscataway, NJ, USA: IEEE Press, 2009, pp. 2683–2688. [Online]. Available: http://dl.acm.org/citation.cfm?id=1732003.1732163
4. N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *J. Artif. Int. Res.*, vol. 16, no. 1, pp. 321–357, Jun. 2002. [Online]. Available: http://dl.acm.org/citation.cfm?id=1622407.1622416
5. G. Dror, R. Sorek, and R. Shamir, "Accurate identification of alternatively spliced exons using support vector machine," *Bioinformatics*, vol. 21, no. 7, pp. 897–901, Apr. 2005.
6. T. Fawcett, "An introduction to roc analysis," *Pattern Recogn. Lett.*, vol. 27, no. 8, pp. 861–874, Jun. 2006.
7. H. Guo, "Learning from imbalanced data sets with boosting and data generation: The databoost-im approach," *SIGKDD Explorations*, vol. 6, p. 2004, 2004.
8. S. Hu, Y. Liang, L. Ma, and Y. He, "Msmote: Improving classification performance when training data is imbalanced," in *Computer Science and Engineering, 2009. WCSE '09. Second International Workshop on*, vol. 2, oct. 2009, pp. 13 –17.
9. S. Koknar-Tezel and L. Latecki, "Improving svm classification on imbalanced data sets in distance spaces," in *Data Mining, 2009. ICDM '09. Ninth IEEE International Conference on*, dec. 2009, pp. 259 –267.

10. I. Kononenko, "Machine learning for medical diagnosis: history, state of the art and perspective," *Artificial Intelligence in Medicine*, vol. 23, pp. 89–109, 2001.
11. H. M. Nguyen, E. W. Cooper, and K. Kamei, "Borderline over-sampling for imbalanced data classification," *Int. J. Knowl. Eng. Soft Data Paradigm.*, vol. 3, no. 1, pp. 4–21, Apr. 2011.
12. F. Provost and T. Fawcett, "Robust classification for imprecise environments," *Mach. Learn.*, vol. 42, no. 3, pp. 203–231, Mar. 2001.
13. F. Sebastiani, "Machine learning in automated text categorization," *ACM Comput. Surv.*, vol. 34, no. 1, pp. 1–47, Mar. 2002.
14. S. Tan, "Neighbor-weighted k-nearest neighbor for unbalanced text corpus," *Expert Syst. Appl.*, vol. 28, no. 4, pp. 667–671, May 2005.
15. V. N. Vapnik, *The nature of statistical learning theory*. New York, NY, USA: Springer-Verlag New York, Inc., 1995.
16. ——, *Statistical Learning Theory*. Wiley-Interscience, 1998.
17. K. Veropoulos, C. Campbell, and N. Cristianini, "Controlling the sensitivity of support vector machines," in *Proceedings of the International Joint Conference on AI*, 1999, pp. 55–60.
18. Z.-Q. Zeng and J. Gao, "Improving svm classification with imbalance data set," in *Proceedings of the 16th International Conference on Neural Information Processing: Part I*, ser. ICONIP '09. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 389–398.
19. Y. Zhang, C.-H. Chu, Y. Chen, H. Zha, and X. Ji, "Splice site prediction using support vector machines with a bayes kernel," *Expert Syst. Appl.*, vol. 30, no. 1, pp. 73–81, Jan. 2006.
20. S. Zou, Y. Huang, Y. Wang, J. Wang, and C. Zhou, "Svm learning from imbalanced data by ga sampling for protein domain prediction," in *Proceedings of the 2008 The 9th International Conference for Young Computer Scientists*, ser. ICYCS 08. Washington, DC, USA: IEEE Computer Society, 2008, pp. 982–987.