Introducción al reconocimiento de patrones
Proyecto final

Eduardo Santos
13 de diciembre de 2010

1. Resumen

Se implementaron diferentes clasificadores con el objetivo de etiquetar lesiones en la piel como benignas o malignas -melanomas-. Los datos de entrenamiento con que contamos corresponden a una serie de lesiones etiquetadas, obtenidas de imágenes dermascópicas de donde se extrajeron diferentes características relativas a la forma, color y textura.

Los algoritmos implementados se centran en hacer frente al problema del desbalance de datos, ya que tenemos muchos más patrones correspondientes a lesiones benignas que malignas.

2. Introducción

En trabajos previos realizados por el Grupo de Tratamiento de Imágenes atacaron el problema de la clasificación de lesiones en la piel utilizando imágenes dermascópicas [1], esta técnica utiliza microscopía de epiluminiscencia para expandir la imagen y permitir una mejor exploración visual. En estos trabajos se encargaron de procesar las imágenes adquiridas y de extraer diferentes características relativas a la forma, el color y la textura de las lesiones.

En casi todos los problemas de diagnóstico médico, se cuenta con una mayor cantidad de muestras de la clase sana, que de la enferma, este caso no es la excepción, siendo mayoría las muestras correspondientes a lesiones benignas, esto llevó a que en [1] utilizaran la técnica de SMOTE para sintetizar datos de la clase minoritaria, de forma de lograr un equilibrio en el número de muestras entre clases, luego utilizaron métodos de *boosting* junto a árboles de decisión para elaborar un clasificador.

En este trabajo tomamos los datos originales, que constan de 433 lesiones benignas y 80 melanomas, con 57 características normalizadas y sus respectivas etiquetas, para implementar técnicas de clasificación en donde no se sinteticen datos, de forma de comparar resultados con los obtenidos en [1].

Comenzaremos con SVM para una clase, y luego exploraremos algunas alternativas que incluyen la selección de características y técnicas sensibles al costo.

Como herramientas se utiliza Matlab con el *toolbox* SVM-KM para la implementación de las diferentes técnicas de SVM. Se utiliza WEKA para la selección de parámetros mediante *information gain*. Para todos los entrenamientos se utiliza validación cruzada con 5 grupos, y para la determinación de los parámetros se realizan búsquedas exhaustivas en rangos convenientes. Se utiliza la etiqueta +1 para los melanomas y -1 para las lesiones benignas.

Durante este trabajo vamos a considerar que el diagnóstico es positivo cuando se clasifica la lesión como maligna, y el diagnóstico es negativo, cuando se lo clasifica como

benigna, por lo tanto llamaremos falsos negativos a los melanomas clasificados como benignos, siendo este el caso más grave, por lo tanto el que tendrá mayor costo.

Como forma de presentar los resultados se utilizarán matrices de confusión, curvas ROC y el área bajo las curvas ROC.

A continuación se comienza con una explicación del algoritmo de SVM, útil ya que es la base para el resto de los algoritmos implementados. Luego seguimos con la implementación de SVM para una clase. Posteriormente analizamos un poco el problema del desbalance de datos y de la "maldición de la dimensionalidad", para luego aplicar selección de características y métodos sensibles al costo. La última técnica implementada es z-SVM la cual es específica para casos de desbalance de datos. Por último comparamos los resultados obtenidos y presentamos nuestras conclusiones.

3. SVM

La técnica SVM busca encontrar el hiperplano que separa las clases de la mejor manera.

Consideramos el problema de clasificación binaria, los datos de entrenamiento son dados de la forma:

$$(x_1, y_1), (x_2, y_2), \dots, (x_l, y_l), \mathbb{x} \epsilon \mathcal{R}^n, y \epsilon \{+1, -1\}. \quad \text{(Ec. 1)}$$

Además no conocemos la distribución subyacente de los patrones. Si estas clases son linealmente separables, va a existir un hiperplano que divide el espacio en dos y deja las clases en diferentes subespacios, lo que se busca es que de todos los hiperplanos que hacen esto, encontrar el que deja mayor margen entre el hiperplano y las muestras de entrenamiento. Esto, a primera vista parece la mejor opción en la búsqueda de la mejor performance cuando se intente clasificar nuevos patrones.

Usando los patrones de entrenamiento, este método retorna los parámetros $\mathbb{w} = [w_1\ w_2\ \dots\ w_n]^T$ y b, con esto formamos la función discriminante $d(\mathbb{x}, \mathbb{w}, b)$ de la forma:

$$d(\mathbb{x}, \mathbb{w}, b) = \mathbb{w}^T\mathbb{x} + b = \sum_{i=1}^{n} w_i x_i + b \quad \text{(Ec. 2)}$$

donde $\mathbb{x}, \mathbb{w} \epsilon \mathcal{R}^n$ y b es escalar. En los puntos pertenecientes al hiperplano, $d(\mathbb{x}, \mathbb{w}, b) = 0$, este es el límite de decisión.

Luego de entrenado, dando un patrón $\mathbb{x}$, produce una salida o dado por:

$$i_F = o = sign\big(d(\mathbb{x}, \mathbb{w}, b)\big), \quad \text{(Ec. 3)}$$

otra forma de ver esto, sí $d\big(\mathbb{x}_p, \mathbb{w}, b\big) > 0$, el patrón $\mathbb{x}_p$ se clasifica como clase 1 ($o = y_1 = +1$) y si $d\big(\mathbb{x}_p, \mathbb{w}, b\big) < 0$, se clasifica como clase 2 ($o = y_2 = -1$).

De todas las formas de representar el hiperplano, utilizamos aquella que nos da que $|d(\mathbb{x}, \mathbb{w}, b)| = 1$ para los puntos más cercanos al hiperplano, estos puntos los llamaremos *support vectors*, esto se hace escalando $\mathbb{w}$ y b, además garantiza que

$$y_i(\mathbb{w}^T\mathbb{x}_i + b) \geq 1 \quad \text{(Ec. 4)}$$

para todos los patrones de entrenamiento.

El margen M que queremos maximizar es la proyección, sobre la normal del hiperplano, de la distancia entre dos vectores de entrenamiento de diferentes clases, entonces:

$$M = \frac{\mathbb{x}_1^T \mathbb{w} - \mathbb{x}_2^T \mathbb{w}}{\|\mathbb{w}\|} \quad \text{(Ec. 5)}$$

ahora, como $\mathbb{x}_1$ y $\mathbb{x}_2$ son *support vectors*, $\mathbb{w}^T \mathbb{x}_1 + b = 1$ y $\mathbb{w}^T \mathbb{x}_2 + b = -1$,

$$M = \frac{2}{\|\mathbb{w}\|} \quad \text{(Ec. 6)}$$

Para hallar el hiperplano óptimo, tenemos que minimizar $\|\mathbb{w}\|^2$ sujeto a (Ec. 4).

Esto se resuelve con la función de Lagrange:

$$L(\mathbb{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\, \mathbb{w}^T \mathbb{w} - \sum_{i=1}^{l} \alpha_i \{y_i [\mathbb{w}^T \mathbb{x}_i + b] - 1\} \quad \text{(Ec. 7)}$$

donde α_i son los multiplicadores de Lagrange. L tiene que ser minimizado respecto de $\mathbb{w}$ y b, y maximizado respecto de los α_i no negativos.

Para resolver este problema en el espacio dual (el espacio de los multiplicadores de Lagrange) derivamos L respecto de $\mathbb{w}$ y b:

$$\frac{dL}{d\mathbb{w}_0} = 0, \rightarrow \mathbb{w}_0 = \sum_{i=1}^{l} \alpha_i y_i \mathbb{x}_i \quad \text{(Ec. 8)}$$

$$\frac{dL}{db_0} = 0, \rightarrow \sum_{i=1}^{l} \alpha_i y_i = 0 \quad \text{(Ec. 9)}$$

Sustituyendo $\mathbb{w}_0$ y b_0 en $L(\mathbb{w}, b, \boldsymbol{\alpha})$ obtenemos:

$$L_d(\boldsymbol{\alpha}) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} y_i y_j \alpha_i \alpha_j \mathbb{x}_i^T \mathbb{x}_j \quad \text{(Ec. 10)}$$

Aquí se ve que el lagrangeano depende, respecto de los patrones de entrenamiento, solamente de el producto $\mathbb{x}_i^T \mathbb{x}_j$. Nuevamente, debemos maximizar $L_d(\boldsymbol{\alpha})$ respecto de los α_i no negativos, sujeto a (Ec. 8) y (Ec. 9).

Este problema se puede expresar de la siguiente forma:

$$Maximizar\ L_d(\boldsymbol{\alpha}) = -\frac{1}{2} \boldsymbol{\alpha}^T H \boldsymbol{\alpha} + \boldsymbol{f}^T \boldsymbol{\alpha} \quad \text{(Ec. 11)}$$

$$sujeto\ a: \mathbb{y}^T \boldsymbol{\alpha} = 0\ y\ \boldsymbol{\alpha} \geq 0 \quad \text{(Ec. 12)}$$

donde $(\boldsymbol{\alpha})_i = \alpha_i$, H es la matriz hessiana ($H_{ij} = y_i y_j (\mathbb{x}_i, \mathbb{x}_j)$) y $\boldsymbol{f}$ es el vector unidad.

Las soluciones α_{0i} determinan los parámetros $\mathbb{w}_0$ y b_0 del hiperplano óptimo de la siguiente manera:

$$\mathbb{w}_0 = \sum_{i=1}^{l} \alpha_{0i} y_i \mathbb{x}_i \quad \text{(Ec. 13)}$$

$$b_0 = \frac{1}{N_{SV}} \left(\sum_{s=1}^{N_{SV}} \left(\frac{1}{y_S} - \mathbb{x}_S^T \mathbb{w}_0 \right) \right) \quad \text{(Ec. 14)}$$

N_{SV} es el número de *support vectors*, y son los únicos que se usan para el cálculos de $\mathbb{w}_0$ y b_0, esto es porque los multiplicadores de Lagrange son cero para todos los patrones que no son *support vectors*.

Finalmente, con $\mathbb{w}_0$ y b_0 tenemos nuestro hiperplano de decisión $d(\mathbb{x})$ y nuestra función discriminante que asigna clases a los patrones i_F.

$$d(\mathbb{x}) = \sum_{i=1}^{l} w_{oi} x_i + b_o = \sum_{i=1}^{l} \alpha_i y_i \mathbb{x}^T \mathbb{x}_i + b_o \quad \text{(Ec. 15)}$$

$$i_F = o = sign\big(d(\mathbb{x})\big) \quad \text{(Ec. 16)}$$

Lo anterior funciona en el caso en que las muestras no se solapen, lo cual no es lo común. Cuando existe solapamiento lo anterior no vale debido a la restricción dada por la (Ec. 4). Para solucionar esto, se debe permitir que algunos patrones queden del lado equivocado del hiperplano de decisión mediante un *soft margin*, y todos los patrones dentro de este margen son obviados. El ancho del *soft margin* es controlado por el parámetro de penalización C. Para resolver este nuevo problema introducimos las variables ξ_i $(i = 1, .., l)$ y ahora el hiperplano debe cumplir que:

$$y_i(\mathbb{w}^T \mathbb{x}_i + b) \geq 1 - \xi_i \quad \text{(Ec. 17)}$$
$$\xi_i \geq 0$$

y la función a ser minimizada cambia para tener en cuenta el parámetro de penalización:

$$\boldsymbol{J}(\mathbb{w}, \xi_i) = \frac{1}{2}\, \mathbb{w}^T \mathbb{w} + C \left(\sum_{i=1}^{l} \xi_i \right)^k \quad \text{(Ec. 18)}$$

Al igual que antes, resolvemos (Ec. 18) restringido a (Ec. 17) mediante el uso de una función de Lagrange:

$$L(\mathbb{w}, b, \xi_i, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{1}{2}\, \mathbb{w}^T \mathbb{w} + C \left(\sum_{i=1}^{l} \xi_i \right)^k - \sum_{i=1}^{l} \alpha_i \{ y_i [\mathbb{w}^T \mathbb{x}_i + b] - 1 + \xi_i \} - \sum_{i=1}^{l} \beta_i \xi_i \quad \text{(Ec. 19)}$$

donde α_i y β_i son multiplicadores de Lagrange.

La solución a este problema es casi idéntica a la anterior, con la única diferencia que los α_i tienen un límite superior dado por C:

$$C \geq \alpha_i \geq 0 \quad \text{(Ec. 20)}$$

C es un parámetro externo al sistema y es definido antes. Para el caso anterior $C = \infty$.

Este método es fácilmente modificable para obtener como superficie de decisión hipersuperficies, en lugar de hiperplanos, y así darle más flexibilidad al método y lograr mejores resultados en datos que no son linealmente separables. Para esto tenemos que buscar un hiperplano en un espacio diferente de mayor dimensión, mapeando los vectores de entrada $\mathbb{x} \epsilon \mathcal{R}^n$ en vectores $\mathbb{z}$ pertenecientes al espacio de mayor dimensión F ($\mathbb{z} =$

$\Phi(\mathbb{x})$ donde Φ es el mapeo de $\mathcal{R}^n \rightarrow \mathcal{R}^f$) y resolver el problema de clasificación lineal en este espacio. La solución a este problema es la función de decisión lineal en el espacio F:

$$i_F(\mathbb{x}) = sign\left(\sum_{i=1}^{l} \alpha_i y_i \mathbb{z}^T(\mathbb{x})\mathbb{z}(\mathbb{x}_i) + b\right) \quad \text{(Ec. 21)}$$

Lo bueno de esto es que no es necesario realizar el mapeo estrictamente, sino que notando que en el problema cuadrático y en la expresión final del clasificador, los patrones solo aparecen en la forma de producto escalar $\mathbb{x}_i^T \mathbb{x}_j$, estos productos son remplazados por productos $\mathbb{z}^T \mathbb{z}_i$ en el espacio F y esto último puede ser expresado usando una *Kernel function*

$$K(\mathbb{x}_i, \mathbb{x}_j) = \mathbb{z}_i^T \mathbb{z}_j = \Phi^{\mathrm{T}}(\mathbb{x}_i)\Phi(\mathbb{x}_j) \quad \text{(Ec. 22)}$$

La *Kernel function* es una función en el espacio de entrada, de esta manera, al usar una *Kernel function* nos evitamos tener que hacer el mapeo $\Phi(\mathbb{x})$, en lugar de eso, los productos requeridos en el espacio F son calculados directamente mediante $K(\mathbb{x}_i, \mathbb{x}_j)$.

Algunas de las *Kernel function* más utilizadas son:

Linear: $K(x_i, x_j) = x_i^T x_j$

Polinómico: $K(x_i, x_j) = (\gamma x_i^T x_j + r)^d, \gamma > 0$

Gaussiano: $K(x_i, x_j) = e^{-\gamma \|x_i - x_j\|^2}, \gamma > 0$

Sigmoide: $K(x_i, x_j) = tanh(\gamma x_i^T x_j + r)$

En resumen, la hipersuperficie de decisión es:

$$d(\mathbb{x}) = \sum_{i=1}^{l} \alpha_i y_i K(\mathbb{x}, \mathbb{x}_i) + b \quad \text{(Ec. 23)}$$

y la función discriminante:

$$i_F(\mathbb{x}) = sign\left(\sum_{i=1}^{l} \alpha_i y_i K(\mathbb{x}, \mathbb{x}_i) + b\right) \quad \text{(Ec. 24)}$$

además, se cumple las restricciones para los multiplicadores de Lagrange:

$$C \geq \alpha_i \geq 0 \quad \text{(Ec. 25)}$$

y cabe recordar que solo los multiplicadores correspondientes a los *support vectors* serán diferentes de cero, por lo que son estos vectores los que determinan la clasificación de nuevos patrones.

4. OC-SVM

$\upsilon - SVM$

Antes de ver el funcionamiento de OC-SVM hay que explicar una variación del algoritmo anterior, llamada $\upsilon - SVM$. $\upsilon - SVM$ usa un nuevo parámetro υ para controlar el número de *support vectors* y el error de entrenamiento. El parámetro $\upsilon \in (0,1]$ es un límite

superior para el error en los patrones de entrenamiento y un límite inferior para la cantidad de *support vectors*. Ahora el problema a resolver es el siguiente:

$$minimizar_{\mathbb{w},b,\boldsymbol{\xi},\rho}\ \frac{1}{2}\ \mathbb{w}^T\mathbb{w} - \upsilon\rho + \frac{1}{l}\sum_{i=1}^{l}\xi_i \quad \text{(Ec. 26)}$$

$$restringido\ a{:}\ y_i(\mathbb{w}^T\Phi(\mathbb{x}_i) + \mathrm{b}) \geq \rho - \xi_i \quad \text{(Ec. 27)}$$
$$\xi_i \geq 0, i = 1, \dots, l, \rho \geq 0$$

Aquí se excluye la constante C y aparecen el parámetro υ y la variable ρ que deberá ser optimizada.

Nuevamente para resolver esto usamos una función de Lagrange:

$$L(\mathbb{w}, b, \boldsymbol{\xi}, \rho, \boldsymbol{\alpha}, \boldsymbol{\beta}, \delta) \quad \text{(Ec. 28)}$$
$$= \frac{1}{2}\ \mathbb{w}^T\mathbb{w} - \upsilon\rho + \frac{1}{l}\sum_{i=1}^{l}\xi_i$$
$$-\sum_{i=1}^{l}\alpha_i\{y_i[\mathbb{w}^T\Phi(\mathbb{x}_i) + b] - \rho + \xi_i + \beta_i\xi_i\} - \rho\delta$$

Esta función tiene que ser minimizada respecto de $\mathbb{w}, b, \boldsymbol{\xi}, \rho$ y maximizada respecto de los multiplicadores de Lagrange $\boldsymbol{\alpha}, \boldsymbol{\beta}, \delta$. Trabajando de manera similar a lo anterior, llegamos a una función en el espacio dual (el correspondiente a los multiplicadores de Lagrange) que debemos maximizar:

$$maximizar_{\boldsymbol{\alpha}}\ L_d(\boldsymbol{\alpha}) = -\frac{1}{2}\sum_{i,j=1}^{l} y_i y_j \alpha_i \alpha_j K(\mathbb{x}_i, \mathbb{x}_j) \quad \text{(Ec. 29)}$$

y a las siguientes restricciones que se deben cumplir:

$$restingido\ a{:}\ \frac{1}{l} \geq \alpha_i \geq 0, \quad \text{(Ec. 30)}$$

$$\sum_{i=1}^{l} y_i\alpha_i = 0 \quad \text{(Ec. 31)}$$

$$\sum_{i=1}^{l} \alpha_i \geq \upsilon \quad \text{(Ec. 32)}$$

Luego tenemos la misma función discriminante:

$$i_F(\mathbb{x}) = sign\left(\sum_{i=1}^{l}\alpha_i y_i K(\mathbb{x}, \mathbb{x}_i) + b\right) \quad \text{(Ec. 33)}$$

Respecto del algoritmo anterior ($C - SVM$), las diferencias son: la restricción dada por la ecuación (Ec. 32) y que no aparece el primer término de la (Ec. 10) en (Ec. 29) .

OC-SVM

Los problemas de una clase son problemas no supervisados, tenemos un grupo de datos de entrenamiento del que desconocemos la distribución de probabilidad y tratamos de determinar un subespacio donde sabemos que si un patrón está dentro de este subespacio

tiene probabilidad P de pertenecer a la clase de los patrones de entrenamiento. Para esto tenemos que encontrar una función que sea positiva dentro del subespacio, y negativa en el resto. Esto permite, dado un nuevo patrón, clasificarlo como perteneciente a la clase de los patrones de entrenamiento, o como un *outlier*, es decir, como cualquier otra cosa.

La estrategia a seguir es la siguiente: mapear los datos a un espacio de mayor dimensión, separar estos datos del origen y encontrar un hiperplano que los separe del origen con el mayor margen posible. Luego, para un nuevo patrón determinamos de qué lado del hiperplano queda, y si es del lado del origen, lo clasificaremos como un *outlier*. Obviamente trataremos de no realizar el mapeo de los datos, sino utilizar las *kernel functions*.

Para separar los datos del origen debemos resolver el siguiente problema:

$$minimizar_{\mathbb{w},\boldsymbol{\xi},\rho}\ \frac{1}{2}\,\mathbb{w}^T\mathbb{w} + \frac{1}{vl}\sum_{i=1}^{l}\xi_i - \rho \tag{Ec. 34}$$

$$restringido\ a\colon (\mathbb{w}^T\Phi(\mathbb{x}_i) + \mathrm{b}) \geq \rho - \xi_i \tag{Ec. 35}$$
$$\xi_i \geq 0, i = 1, \dots, l, \rho \geq 0$$

Aquí, $\upsilon \in (0,1]$ es análogo al parámetro introducido en $\upsilon - SVM$, y tiene las mismas propiedades, es decir, es un límite superior para los *outliers* en los patrones de entrenamiento y un límite inferior para la cantidad de *support vectors*.

La función de Lagrange en este caso es:

$$L(\mathbb{w},\boldsymbol{\xi},\rho,\boldsymbol{\alpha},\boldsymbol{\beta}) = \frac{1}{2}\,\mathbb{w}^T\mathbb{w} + \frac{1}{vl}\sum_{i=1}^{l}\xi_i - \rho - \sum_{i=1}^{l}\alpha_i(\mathbb{w}^T\Phi(\mathbb{x}_i) - \rho + \xi_i + \beta_i\xi_i) \tag{Ec. 36}$$

Luego llegamos al problema en el espacio dual:

$$maximizar_{\boldsymbol{\alpha}}\ L_d(\boldsymbol{\alpha}) = -\frac{1}{2}\sum_{i,j=1}^{l}\alpha_i\alpha_j K(\mathbb{x}_i,\mathbb{x}_j) \tag{Ec. 37}$$

Con las restricciones:

$$restingido\ a\colon \frac{1}{vl} \geq \alpha_i \geq 0, \tag{Ec. 38}$$

$$\sum_{i=1}^{l}\alpha_i = 1 \tag{Ec. 39}$$

Y la función discriminante en este caso es:

$$i_F(\mathbb{x}) = sign\left(\sum_{i=1}^{l}\alpha_i K(\mathbb{x},\mathbb{x}_i) - \rho\right) \tag{Ec. 40}$$

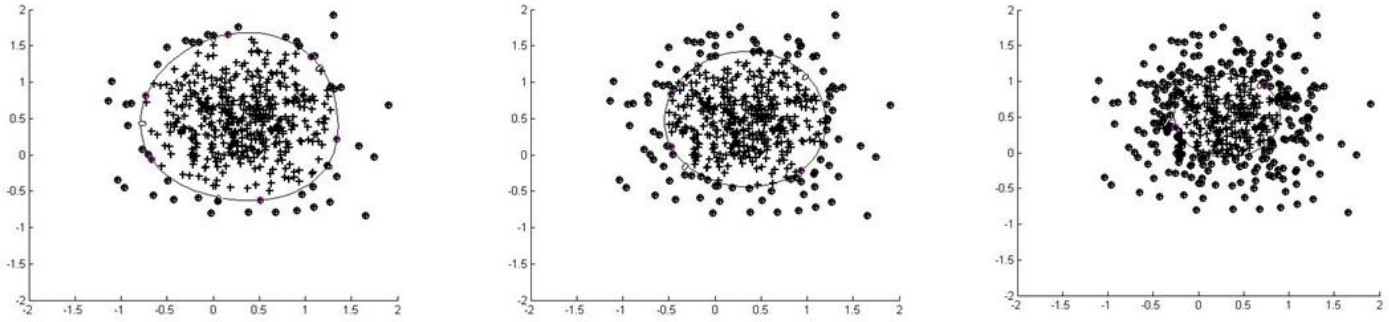

Ilustración 1 - Efecto de la variación del parámetro υ en la superficie de decisión para $OC-SVM$, a la izquierda, $\upsilon = 0.1$, en el medio $\upsilon = 0.2$ y a la derecha $\upsilon = 0.5$

OC-SVM para clasificación de lesiones en la piel

Vamos a implementar OC-SVM en nuestro caso, tomando como patrones de entrada los correspondientes a las lesiones benignas, y tomaremos como melanomas todos los que sean clasificado como *outliers*. Esto lo hacemos así debido a la mayor cantidad de patrones del tipo benigno que tenemos, lo que hace suponer que tendremos una mejor adaptación al problema de esta forma. Para la implementación se utiliza el *toolbox* SVM-KM para utilizar en Matlab. Esta herramienta permite fácilmente entrenar un clasificador, teniendo como entrada, los datos, el tipo de *kernel*, el parámetro de ajuste relativo a ese *kernel* y el υ, a su vez la función devuelve, el vector $\mathbb{w}, \rho$, los $\alpha_i \neq 0$ correspondientes a los *support vectors* y los respectivos vectores. Con esto es suficiente para tener nuestra función discriminante, en donde, si la función da un valor positivo para un nuevo patrón, significa que es una lesión benigna, y si la función devuelve un valor negativo, tenemos un melanoma.

Debido a que los mejores resultados para estos casos se ven con el *kernel* gaussiano, este es el que usamos para el clasificador. Entonces debemos seleccionar dos parámetros, υ y γ. Para esto realizamos una búsqueda exhaustiva en un rango razonable de valores. Como el objetivo es encontrar curvas *ROC*, vamos a tener diferentes parámetros, que corresponderán a valores de verdaderos negativos -lesiones benignas clasificadas como benignas- que nosotros fijamos, mientras buscamos la menor cantidad de falsos negativos -melanomas clasificados como lesiones benignas-. A su vez, utilizamos validación cruzada con 5 grupos, de forma que para entrenar usamos 4/5 de los patrones benignos, y corroboramos con 1/5 de los patrones benignos y la totalidad de los malignos. Los resultados se muestran en las siguientes tablas y figuras.

Tabla 1 - Resultados del clasificador OC-SVM

Resultados Validación cruzada			
Verdaderos negativos	Falsos negativos	Lambda (kernel)	NU
0.90760387	0.1175	0.9	0.0616
0.91915338	0.15	1.2	0.0598
0.9307029	0.1525	1.1	0.0548
0.94225242	0.175	1.2	0.0462
0.95380193	0.205	1.4	0.0406
0.96310426	0.22	1.6	0.0396
0.96535145	0.22	1.6	0.0368
0.97690097	0.275	1.7	0.017
0.98845048	0.355	2.2	0.0174
0.99310165	0.4625	3	0.0066

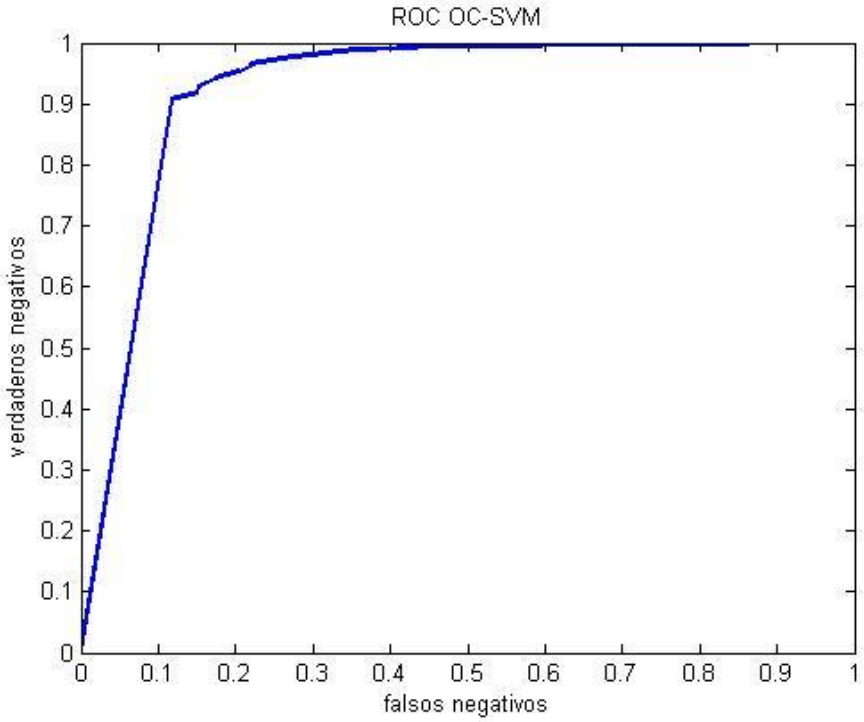

Ilustración 2 - Curva ROC para el clasificador OC-SVM

Para graficar la curva ROC y calcular el área, tomamos los promedios de los valores obtenidos en los diferentes grupos de la validación cruzada. El área bajo la curva ROC obtenida es $AUC = 0{,}9234$

Para medir el rendimiento de este clasificador, volvemos a entrenar sobre 4/5 de las muestras negativas con los parámetros promediados correspondientes a la cercanía de un 95% de verdaderos negativos, esto es $\upsilon = 0{,}0406$ y $\gamma = 1{,}4$.

el resultado es para estos valores:

Matriz de confusión:

Tabla 2 - Matriz de confusión para el clasificador OC-SVM

	Clasificados benignos (Negativos)	Clasificados malignos (Positivos)
Lesiones benignas	0,9663	0,0337
Melanomas	0,2375	0,7625

Dado que los resultados no son buenos, trataremos de buscar la razón de esto y tratar de mejorar los resultados.

5. Desbalance de clases

Un problema notorio que tiene este conjunto de datos, y que ayuda a la baja performance de los clasificadores, es la diferencia entre las muestras de lesiones malignas y benignas, esto es conocido como desbalance de clases. Este tipo de problemas están siendo estudiados actualmente dado que muchos problemas reales tienen esta característica [2], en este tipo de problemas casi todos los patrones son de una misma clase, mientras que unas pocas muestras pertenecen a la otra clase, que generalmente es la más importante. El problema es que los clasificadores tratan de optimizar la precisión total en los datos de entrenamiento, lo que lleva a que "prefieran" clasificar bien los datos mayoritarios, mientras no se preocupan de los minoritarios, con esto se obtienen buenos valores de precisión en general, pero malos valores sobre los datos minoritarios [3].

Los problemas de desbalance de clase se ven en numerosas aplicaciones: Detección de usuarios no reales en telecomunicaciones, detección de pérdidas de petróleo mediante imágenes satelitales, clasificación de textos, manejo de riegos, recuperación de datos y tareas de filtrado, diagnósticos médicos, monitoreo de redes y detección de intrusos, detección de fraudes y detección de terremotos entre otros. [2]

Las formas de tratar con estos datos se pueden reunir en tres tipos: Cambio en la distribución de las clases, selección de características y cambio en los clasificadores [2].

Cambio en de la distribución de las clases

Una forma directa de combatir el desbalance de las clases, es lograr que las clases se balanceen. Esto se puede lograr mediante muestreo de la clase mayoritaria, sobre muestreo de la clase minoritaria o una combinación de ambas.

Dentro del muestreo de la clase mayoritaria, lo que buscamos es reducir el número de patrones de esta clase hasta lograr cierta igualdad en cantidad con las muestras de la clase minoritaria. Una forma es el muestreo aleatorio, lo cual puede llevar a la pérdida de patrones útiles que podrían ser importantes para los clasificadores.

Los otros tipos de muestreo se basan en dos hipótesis: uno considera que los patrones cerca del borde de clasificación son ruido, y el otro considera que los patrones con más vecinos de otra clase son ruido. En vista de esto, la mayoría de estos algoritmos de muestreo se basan en los clasificadores de vecino más cercano, como *Condensed Nearest Neighbor Rule*, *OSS, Wilson's Edited Nearest Neighbor Rule, Neighborhood Cleaning Rule*, más detalles en [2].

El sobre muestreo tiene el mismo objetivo que el muestreo, pero trabaja por la contraria, busca incrementar los patrones de la clase minoritaria, inventando nuevos datos, aquí también podemos hacer un sobre muestreo aleatorio, en este caso, replicando

aleatoriamente muestras de la clase minoritaria. Esto aumenta la probabilidad de *over-fitting*, ya que hace copias exactas de las muestras.

Hay muchos métodos de sobre muestreo basados en SMOTE. SMOTE genera muestras sintéticas de la clase minoritaria. Básicamente genera las muestras interpolando las muestras cercanas que tenemos de esa clase. Con esto evita el *over-fitting,* y fuerza al borde de decisión a irse hacia el espacio de la clase mayoritaria.

No se aplican técnicas de este tipo en este trabajo, ya que fueron tratadas para estos mismos datos en [1].

Selección de características

Trabajar con selección de características nos ayuda no solo con el desbalance de clases, también con el problema de la poca cantidad de datos disponibles, contamos con 513 datos para entrenar el clasificador, teniendo 57 características, por lo tanto, una relación entre patrones y características menor a 10, esto, que dificulta el entrenamiento, es conocido como "la maldición de la dimensionalidad", que dice que no por tener muchas características, vamos a tener mejores clasificadores.

El objetivo de la selección es obtener las características que optimizan la performance del clasificador. El número de características es definido por el usuario. Cuando contamos con muchas características, las ordenamos según algún tipo de medida [3].

Como los problemas de desbalance de clase son generalmente acompañados por problemas de alta dimensión, la selección de características es una forma natural de proceder en estos casos. Y, en algunas aplicaciones, la selección de características ha demostrado ser más útil que el sobre muestreo y el uso de algoritmos específicos para tratar con el desbalance de datos [3].

Selección de características como parte de clasificadores ha sido ampliamente estudiado, pero su uso para tratar con datos desbalanceados es reciente. En [3] se discute algunos de estos trabajos.

Algunos de los métodos para selección de características son:

Chi-Square: Es un test estadístico que mide la independencia de una característica respecto de las clases.

Info gain: Evalúa la relación entre la entropía de las clases y la entropía condicional de las clases dada un característica.

Pearson Correlation Coefficient: Evalúa el grado de relacionamiento entre una característica y las clases.

Feature Assessment by Sliding Thresholds: utiliza el área debajo del *ROC* para la selección de características.

Cambio en los clasificadores y clasificadores específicos

De todos los cambios posibles en clasificadores y todos los clasificadores específicos para desbalance de datos, solo vamos a ver los relacionados a sensibles a costos.

Estos métodos consideran el costo asociado a una mala clasificación de un patrón. Para esto se utiliza una matriz que indica el costo de clasificar incorrectamente una muestra. Estudios muestran que en algunas aplicaciones con desbalance de clases, los métodos sensibles al costo tienen rendimientos superiores a las técnicas de muestreo [4].

Aprendizaje sensible al costo

Un concepto fundamental en estos métodos es el de matriz de costos. La matriz de costos es una representación numérica de la penalización por clasificar una muestra erróneamente. Por ejemplo, en un caso de dos clases, se define $C(Min, May)$ como el costo de clasificar incorrectamente una muestra de la clase mayoritaria como de la clase minoritaria, y $C(May, Min)$ el costo en el caso contrario. Típicamente no hay un costo asociado cuando se clasifica correctamente, y el costo de clasificar incorrectamente muestras de la clase minoritaria es mayor al caso contrario, esto es: $C(May, Min) > C(Min, May)$. El objetivo de un método sensible al costo es el de minimizar el costo total sobre las muestras de entrenamiento, lo que se conoce como *Bayes conditional risk*.

Existen muchas formas de implementar métodos que sean sensibles al costo, pero la mayoría cae dentro de tres clases. El primer método aplica los costos de clasificación incorrecta a los patrones de entrenamiento de forma de alterar el espacio de características. Los segundos métodos aplican técnicas para la minimización de costos dentro de esquemas de combinación de métodos, esto consiste en varias *Meta técnicas* donde métodos estándar de clasificación son integrados mediante métodos de ensamble desarrollando un método sensible al costo. La otra forma de implementar métodos sensibles al costo es la de incorporar estos conceptos directamente dentro de los algoritmos, de forma de alterarlos para que tomen en cuenta esto.

Aplicación de selección de características para el caso de detección de lesiones en la piel.

Utilizando la selección de características trataremos de solucionar dos problemas que aparecen en nuestros datos, la maldición de la dimensionalidad y el desbalance entre clases. Para esto implementamos dos métodos de selección: El discriminante de Fisher, e *Info gain*. Ambos métodos nos proporcionan una lista con las características ordenadas según el resultado de aplicar los respectivos test.

Discriminante de Fisher

Esté método estadístico se usa para mostrar que tan discriminante es una característica. Se utiliza la siguiente definición del discriminante de Fisher [5]:

$$f = \frac{(\mu_1 - \mu_2)^2}{\sigma_1^2 + \sigma_2^2} \qquad \text{(Ec. 41)}$$

donde μ_1, μ_2, σ_1^2 y σ_2^2 son las medias y las varianzas de las dos clases respectivamente. Altos valores de f indican que esa característica es capaz de discriminar las clases.

Para aplicar esto a nuestros datos, se miden los valores de f sobre todas las muestras y se obtiene la lista de las características con su respectivo valor.

Tabla 3 - Ordenamiento de las características tomando en cuenta su *Fisher Ratio*

Fisher Ratio					
Característica	Fisher ratio	Característica	Fisher ratio	Característica	Fisher ratio
37	16.9356589	24	5.43980915	48	0.87077828
43	11.878208	38	4.88324741	7	0.83678602
45	11.878208	3	4.66669967	47	0.81613696
8	11.8498096	15	3.98622846	10	0.67692051
9	11.4853698	12	3.3788763	35	0.66953491
31	11.2546145	34	3.22892876	18	0.63252153
33	11.2546145	42	3.01813632	26	0.59440522
28	10.3396322	14	3.00732214	51	0.51873025
44	9.98195088	2	2.9226658	30	0.41874148
4	9.93841521	41	2.79504609	22	0.39587708
27	9.54546203	39	2.35732747	25	0.2439496
19	9.46718452	36	2.00595044	11	0.1991808
21	9.46718452	6	1.76267811	49	0.17075721
16	7.28137563	1	1.58754444	46	0.17075721
40	6.9619901	55	1.5653462	57	0.16761153
5	6.88602372	23	1.51571854	54	0.14301769
13	6.61517923	17	1.25199968	53	0.03436566
20	6.1044233	52	1.0585258	50	0.03436566
32	5.91988029	29	0.95124504	56	0.02213214

Visto los resultados, seleccionamos las 13 características con mayor valor de f, debido a que en ese punto se da un quiebre, con un salto sensible en el poder de discriminación.

Luego se toman estas características y se entrena el clasificador OC-SVM. Por razones de costo computacional, no podremos construir una curva ROC, sino que buscamos, para el caso en que los verdaderos negativos llegan a un 95%, el menor valor de falsos negativos.

A continuación se muestran los valores obtenidos:

Tabla 4 - Resultados del clasificador OC-SVM con selección de características mediante *Fisher Ratio*

	Grupo 1	Grupo 2	Grupo 3	Grupo 4	Grupo 5	Promedio
Verdaderos negativos	0.9551	0.9535	0.9535	0.9535	0.9535	0.95382
Falsos negativos	0.2875	0.1625	0.1125	0.175	0.2375	0.195
Lambda	0.5	0.5	0.5	1	0.5	0.6
Nu	0.013	0.067	0.085	0.08	0.025	0.054

Matriz de confusión:

Tabla 5 - Matriz de confusión para el clasificador OC-SVM con selección de características mediante *Fisher Ratio*

	Clasificados benignos (Negativos)	Clasificados malignos (Positivos)
Lesiones benignas	0,95382	0,04618
Melanomas	0,195	0,805

Info Gain

Este método de clasificación de características ya se describió anteriormente, y en [3] se indica como uno de los que da mejores resultados. La aplicación de este método se realiza en Weka, donde ya está implementado, y se obtiene la lista de características:

Tabla 6 - Ordenamiento de las características tomando en cuenta su *Info Gain*

Característica	*infogain*	Característica	*infogain*	Característica	*infogain*
31	0.2865	32	0.1411	12	0.0722
33	0.2865	44	0.1395	22	0.0717
40	0.2816	9	0.1391	15	0.0711
19	0.2757	20	0.1356	13	0.071
21	0.2757	27	0.1347	30	0.0702
28	0.2622	8	0.1338	35	0.0688
16	0.2534	3	0.1331	29	0.0687
45	0.2499	7	0.1284	17	0.0637
43	0.2499	37	0.119	52	0.0606
47	0.2059	24	0.1122	34	0.0603
2	0.1974	6	0.0937	42	0.0603
51	0.1935	36	0.0859	5	0.0587
49	0.1725	41	0.0858	14	0.0573
46	0.1725	56	0.0839	39	0.054
57	0.1668	4	0.0834	10	0.0326
54	0.1668	38	0.0807	18	0.0326
55	0.1632	26	0.0787	25	0.0324
50	0.1551	23	0.0756	11	0.0309
53	0.1551	48	0.0747	1	0

Nuevamente explorando visualmente los datos, tomamos las primeras doce características debido que es donde se encuentra un salto en los valores.

Se procede de forma similar a lo anterior y se obtienen los siguientes resultados:

Tabla 7 - Resultados del clasificador OC-SVM con selección de características mediante *Info Gain*

	Grupo 1	Grupo 2	Grupo 3	Grupo 4	Grupo 5	Promedio
Verdaderos negativos	0.9551	0.9535	0.9535	0.9535	0.9535	0.95382
Falsos negativos	0.2125	0.2375	0.2625	0.2625	0.2125	0.2375
Lambda	1.5	1	0.5	2.5	1.5	1.4
Nu	0.067	0.035	0.017	0.031	0.075	0.045

Matriz de confusión:

Tabla 8 - Matriz de confusión para el clasificador OC-SVM con selección de características mediante *Info Gain*

	Clasificados benignos (Negativos)	Clasificados malignos (Positivos)
Lesiones benignas	0,95382	0,0337
Melanomas	0,2375	0,7625

Cost sensitive SVM

Otras formas de tratar con datos desbalanceados eran la referidas a métodos sensibles al costo. Dentro de estos métodos se encontraban los que variaban algoritmos conocidos para que tuvieran en cuenta esto. *Cost sensitive SVM* (*CS-SVM*) se encuentra dentro de estos métodos. En el problema de SVM, si nos paramos en la (Ec. 18) vemos que se puede expresar de la siguiente forma [6]:

$$minimizar\ \boldsymbol{J}(\mathbb{w}, \xi_i) = \frac{1}{2}\ \mathbb{w}^T\mathbb{w} + \sum_{i=1}^{l} C_i\xi_i \qquad \text{(Ec. 42)}$$

restringido a (Ec. 17).

Usualmente C_i es único para todos los patrones, pero en *CS-SVM* este factor es utilizado para representar un peso dado a cada clase, por lo que (Ec. 42) queda:

$$minimizar\ \boldsymbol{J}(\mathbb{w}, \xi_i) = \frac{1}{2}\ \mathbb{w}^T\mathbb{w} + \sum_{i/y_i=1} C^+\xi_i + \sum_{i/y_i=-1} C^-\xi_i \qquad \text{(Ec. 43)}$$

restringido a (Ec. 17).

La resolución de este problema es idéntica a la ya vista para *C-SVM*, siendo la única diferencia que ahora los multiplicadores de Lagrange tendrán cotas superiores diferentes, esto es:

$$C^+ \geq \alpha_i \geq 0, y_i = +1 \qquad \text{(Ec. 44)}$$
$$C^- \geq \alpha_i \geq 0, y_i = -1 \qquad \text{(Ec. 45)}$$

y la función discriminante es la misma que en (Ec. 21).

Si durante el entrenamiento utilizamos $C^+ > C^-$ el plano de clasificación se va a volcar hacia la clase negativa (en nuestro caso, las lesiones benignas) haciendo que más patrones caigan del lado de la clase positiva, haciendo que una clasificación incorrecta de la clase positiva sea más difícil. Esto se puede ver como una forma de tratar el desbalance, aunque también es cierto que clasificar como benigno un melanoma tiene un costo mayor al caso contrario.

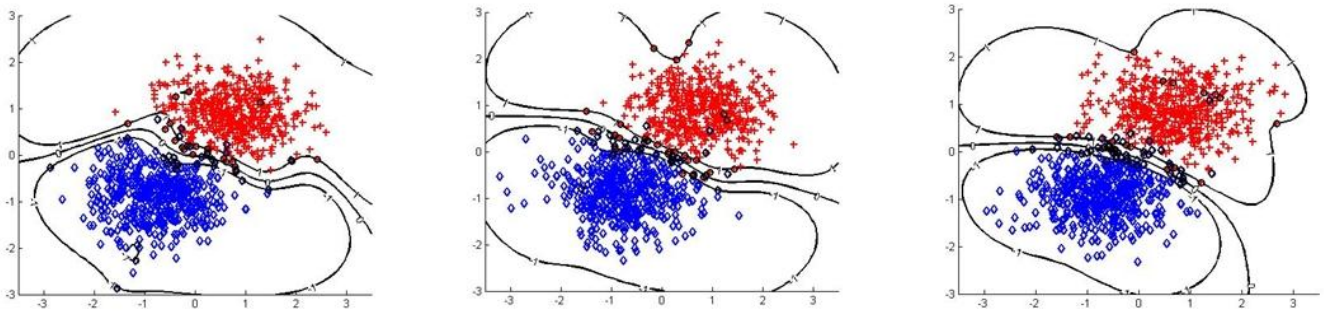

Ilustración 3 - Efecto de la variación del costo de las clases en el clasificador *CS-SVM,* a la izquierda, $C_{ROJO} = C_{AZUL} = 100$, en el medio, $C_{ROJO} = 100, C_{AZUL} = 80$ y a la derecha $C_{ROJO} = 100, C_{AZUL} = 20$

Para aplicar esto se procedió de la siguiente manera:

Las lesiones benignas, que son nuestros patrones negativos, son etiquetados con -1, y los melanomas, al ser nuestros patrones positivos, son etiquetados con +1.

Nuevamente por temas de costo computacional, no podemos, en una primera instancia, elaborar una curva ROC, por lo que nuevamente tomamos como objetivo una tasa de verdaderos negativos de un 95% y tratamos de disminuir lo más posible los falsos negativos. Además, dividimos en dos la búsqueda de parámetros, en una primera instancia buscamos el objetivo anterior utilizando *C-SVM* y realizando una búsqueda exhaustiva del parámetro γ del *kernel* y el parámetro de regularización del algoritmo. Luego tomamos los parámetros encontrados y los usamos para entrenar *CS-SVM* variando los costos, esta forma de implementación nos permite ver el efecto de variar los costos respecto del algoritmo de dos clases normal.

Siempre consideramos que el costo de clasificar incorrectamente un melanoma C^+ mayor a C^-. C^+ va entre 200 y 1000 mientras que C^- entre 10 y 250, estos son rangos empíricos. A continuación se muestran los resultados obtenidos.

Tabla 9 - Resultados del clasificador C-SVM

C-SVM						
	Grupo 1	Grupo 2	Grupo 3	Grupo 4	Grupo 5	Promedio
Verdaderos negativos	0.95505618	0.95348837	0.95348837	0.95348837	0.95348837	0.95380193
Falsos negativos	0.1875	0.3125	0.125	0.3125	0.0625	0.2
Lambda	2.5	4.5	5.5	3.5	7	4.6
Par. Regularización	3.90E-05	5.00E-06	1.00E-06	1.00E-06	4.00E-06	0.00001

Tabla 10 - Resultados del clasificador CS-SVM

CS-SVM						
	Grupo 1	Grupo 2	Grupo 3	Grupo 4	Grupo 5	Promedio
Verdaderos negativos	0.95505618	0.95348837	0.95348837	0.95348837	0.95348837	0.95380193
Falsos negativos	0.1875	0.125	0.0625	0.3125	0.0625	0.15
C+	310	250	220	270	910	392
C-	30	30	20	20	230	66

Cost sensitive SVM + feature selection

Tratando de mejorar los resultados, se vuelve a implementar *CS-SVM*, esta vez aplicando también la selección de características, utilizando el ordenamiento elaborado mediante *fisher ratio*, utilizando las trece características con mayor poder discriminante.

Se utiliza *kernel* gaussiano, con $\gamma = 4$.

Tabla 11 - Resultados del clasificador CC-SVM con selección de parámetros mediante *fisher ratio*

Resultados Validación cruzada			
Verdaderos negativos	Falsos negativos	C+	C-
0.73255814	0.0625	200	10
0.813953488	0.09375	200	12.5
0.839169062	0.04166667	200	13.3
0.885876666	0.03125	200	17.5
0.906976744	0.08333333	200	20
0.918604651	0.0625	200	27.5
0.930820486	0.078125	205	26.25
0.933727463	0.078125	205	27.5
0.942513718	0.14583333	200	25
0.953880324	0.125	200	38.75
0.965508231	0.20833333	200	48.3
0.977005487	0.27083333	206.7	71.7
0.988470081	0.25	200	77.5

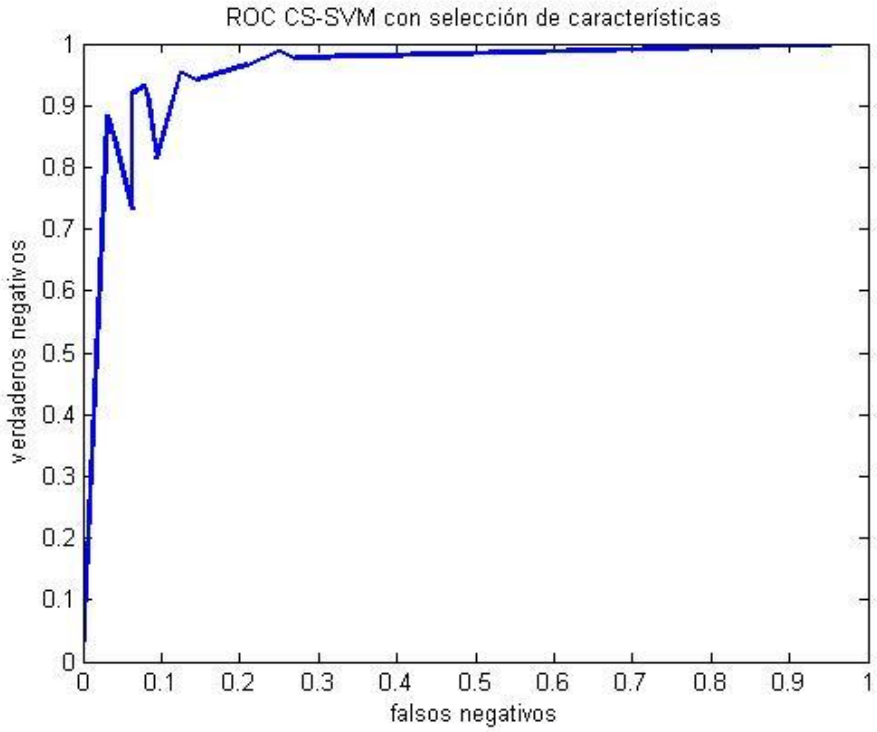

Ilustración 4 - Curva ROC para el clasificador *CS-SVM con selección de características*

El área bajo la curva ROC obtenido es $AUC = 0{,}9564$

La matriz de confusión para un valor de verdaderos negativos cercano a 95% es:

Tabla 12 - Matriz de confusión para el clasificador CC-SVM con selección de características mediante *Fisher ratio*

	Clasificados benignos (Negativos)	Clasificados malignos (Positivos)
Lesiones benignas	0.954	0,046
Melanomas	0.125	0,875

z-SVM

Se implementó otro algoritmo basado en *SVM* y orientado a problemas de desbalance de clases [7].

Este método se basa en un ajuste posterior del límite de decisión. Se entrena el clasificador de forma normal y luego se modifica el límite utilizando el parámetro z.

Luego de tener entrenado el clasificador, el vector de peso se puede escribir así:

$$\mathbb{w} = \sum_{i/y_i=1} \alpha_i y_i \Phi(\mathbb{x}_i) + \sum_{i/y_i=-1} \alpha_i y_i \Phi(\mathbb{x}_i) \quad \text{(Ec. 46)}$$

y la función de decisión puede ser reescrito de esta forma:

$$d(\mathbb{x}) = \sum_{i/y_i=1} \alpha_i y_i K(\mathbb{x}, \mathbb{x}_i) + \sum_{i/y_i=-1} \alpha_i y_i K(\mathbb{x}, \mathbb{x}_i) + b \quad \text{(Ec. 47)}$$

Nuevamente, como en *CS-SVM,* tomamos por un lado los *support vectors* correspondientes a la clase positiva y los correspondientes a la clase negativa, junto a sus correspondientes multiplicadores de Lagrange. Lo que este método hace, en su propósito de mejorar la detección de patrones de la clase minoritaria, es introducir un peso multiplicativo z asociado a los *support vectors* de la clase minoritaria:

$$d(\mathbb{x}, z) = z \sum_{i/y_i=1} \alpha_i y_i K(\mathbb{x}, \mathbb{x}_i) + \sum_{i/y_i=-1} \alpha_i y_i K(\mathbb{x}, \mathbb{x}_i) + b \quad \text{(Ec. 48)}$$

El parámetro z puede ser visto como un peso dado a los multiplicadores de Lagrange, logrando que la clasificación de los minoritarios mejore. De esta forma, z es una corrección al límite de decisión que habíamos calculado.

La implementación de este algoritmo es similar a las anteriores. Se realiza una búsqueda exhaustiva del parámetro del *kernel* y de z de manera de obtener, para un rango de verdaderos negativos de 95%, el menor rango de falsos positivos. Siempre utilizando validación cruzada con cinco grupos. Los resultados se muestran a continuación.

Tabla 13 - Resultados del clasificador z-SVM

	Grupo 1	Grupo 2	Grupo 3	Grupo 4	Grupo 5	Promedio
Verdaderos negativos	0.955	0.9535	0.9535	0.9419	0.9535	0.95148
Falsos negativos	0.25	0.375	0.125	0.25	0.1875	0.2375
Lambda	0.5	1.5	0.5	4	1.5	1.6
Z	1	1.0000087	1	1.0000001	1.0000075	1.00000326

6. Comparación de resultados:

Comparamos los diferentes resultados obtenidos en este trabajo, juntos a los que se muestran en [1], para la implementación de SVM mostrada en ese trabajo, no hay información que aclare con que serie de datos trabajaron, o los parámetros que utilizaron.

Tabla 14 - Comparación de resultados para diferentes clasificadores sobre los datos de lesiones en la piel.

Método	Verdaderos Negativos	Falsos Negativos	AUC
OC-SVM	0,967	0,234	0.9234
OC-SVC + Fisher	0,954	0,195	
OC-SVM + InfoGain	0,954	0,238	
CS-SVM	0.954	0.150	
CS-SVM + Fisher	0.954	0.125	0.9564
z-SVM	0.951	0.234	
AdaBoost - C4.5	0.95	0.0875	0.981
SVM	0.95	0.14	0.966

La mejor performance la logra el método AdaBoost - C4.5 implementado en [1] utilizando SMOTE para sintetizar datos de la clase minoritaria, mientras que los métodos de OC-SVM se alejan de los mejores resultados.

7. Conclusiones

OC-SVM no arrojó los resultados esperados, situándose entre los peores algoritmos probados en este trabajo, con más de un 20% de falsos negativos para un 95% de verdaderos negativos. Para estos datos no se puede dejar de lado una clase y tomar información solo de la clase mayoritaria, esto se debe a lo solapados que están los datos, y a los pocos datos con lo que contamos.

Las técnicas empleadas aquí para contrarrestar el desbalance de los datos mostraron mejoras en los resultados, siendo la combinación de estas técnicas (selección de características y métodos sensibles al costo) la que arrojó el mejor resultado, con una relación de un 12,5% de falsos negativos para un 95,4% de verdaderos negativos, a parte de un *AUC* de 0.9564.

Se pude observar una mejora en los resultados de la clasificación entre *C-SVM* y *CS-SVM* lo que quiere decir que estas técnicas si mejoran la clasificación en casos de desbalance de datos.

z-SVM mostró no adaptarse a los datos con los que trabajamos y no dio resultados buenos.

En ninguno de los casos se logró mejorar lo mostrando en [1] donde se utilizó SMOTE, pero en [8] se muestran dudas sobre este método ya que a veces se obtienen resultados muy buenos pero que no son confiables.

Si bien en este trabajo nos concentramos en luchar contra el desbalance de los datos, hay que decir que comparado a lo que se ve en la bibliografía, una relación de 5,4 muestras de la clase mayoritaria respecto de la minoritaria no es un gran desbalance, y que lo que más afecta a los resultados de la clasificación es la poca cantidad de datos con que contamos, y la gran cantidad de características.

8. Bibliografía

[1] Germán Capdehourat, Andrés Corez, Anabella Bazzano, and Pablo Musé, "Pigmented skin lesions classification using dermatoscopic images".

[2] Xinjian Guo, Yilong Yin, Cailing Dong, Gongping Yang, and Guangtong Zhou, "On the Class Imbalance Problem," 2008.

[3] Mike Wasikowski and Xue-wen Chen, "Combating the Small Sample Class Imbalance Problem Using Feature Selection," 2009.

[4] Haibo He and Edwardo Garcia, "Learning from Imbalanced Data," 2009.

[5] Cheng Weng and Josiah Poon, "A data complexity analysis on imbalanced datasets and an alternative imbalance recovering strategy," 2006.

[6] Dai Yuanhong, Chen Hongchang, and Peng Tao, "Cost-sensitive Support Vector Machine Based on Weighted Attribute," 2009.

[7] Tasadduq Imam, Kai Ming Ting, and Joarder Kamruzzaman, "z-SVM: An SVM for Improved Classification of Imbalanced Data," in *AI 2006: Advances in Artificial Intelligence*.: Springer, 2006.

[8] Marcelo Fiori, "Introducción al Reconocimiento de Patrones, Proyecto final," 2010.

[9] Bernhard Schlkopf and Alexander Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*.: The MIT Press, 2006.

[10] Richard Duda, Peter Hart, and David Stork, *Pattern Classification*.: Wiley-Interscience, 2000.

[11] Vojislav Kecman, *Learning and Soft Computing, Support Vector Machines, Neural Networks, and Fuzzy Logic Models*.: The MIT Press , 2001.