

IDENTIFICAR COBERTURA VEGETAL DE SUELO CLASIFICANDO PIXELES EN IMÁGENES HIPERESPECTRALES CON SVM (MÁQUINA DE SOPORTE VECTORIAL)

Adrian Trueba Espinosa, Juan Carlos Moreno Sánchez, José Sergio Ruiz Castilla

atruebae@hotmail.com, cloudspaba@gmail.com, jsergioruizc@gmail.com
Universidad Autónoma del Estado de México, México
Avenida Jardín Zumpango S/N Fraccionamiento el Tejocote
Texcoco, Estado de México – México

Resumen

Se realizó un experimento para clasificar 4 clases de cobertura vegetal de suelo usando imágenes hiperespectrales, con 63 bandas. Se trabajó con un pixel representado por vector de 63 características (uno por banda). Se probaron nueve filtros y el análisis sin filtro. Para la clasificación se usó una máquina de soporte vectorial (SVM), con un kernel perceptron multicapa (MLP). Se obtienen precisiones aceptables, mejorando muchas encontradas en la literatura [1] [2]. Con este trabajo se demuestra que es mejor usar imágenes hiperespectrales que las comúnmente usadas. Los resultados sugieren que es necesario emplear otras técnicas o pre-proceso para mejorar las clasificaciones en este tipo de imágenes.

Abstract.

The experiment was performed by classification 4 labels of vegetable ground cover implements hyperspectral images. This image takes 63 bands hyperspectral. The principal feature of the work, it use the pixel like a vector with 63 values (one value by one band). The analysis feature 9 experiments with filters and one experiment without filter. For step of classification implement support vector machine (SVM), with kernel multilayer perceptron (MLP). The accuracy of result was admissible, and it was better than those found in the literature [1] [2]. With work shown that hyperspectral images are better than other image for vegetal ground cover. The result suggest it is necessary employ another machine learner's or technical of preprocess for improve rating up in such images.

Palabras clave

Imágenes hiperespectrales, Máquinas de Soporte Vectorial, Kernel Perceptron Multicapa.

Introducción.

Desde que existen los sensores remotos y la captura de imágenes, ha surgido la necesidad de conocer que hay en ellas. Se han realizado diversos estudios ejecutando algoritmos computacionales para: identificación y clasificación de las distintas coberturas que existen en la superficie terrestre, determinación de humedad del suelo, clasificación de la vegetación, mapas hidrotermales, estudios climáticos multitemporales, geológicos, demográficos entre otros. Las imágenes más empleadas son las SAR (Radar de Apertura Sintética), LANDSAT y SPOT. Estas dos últimas obtiene imágenes multiespectrales; que tienen entre 5 y 15 bandas. Los resultados de esos trabajos en su momento fueron aceptables, pero siempre existió una limitación por el número de bandas con las que se contaba; ya que reduce la posibilidad de una clasificación con mayor precisión. Este restrictivo se resuelve con sensores hiperespectrales; que obtienen imágenes con más bandas, generalmente entre 60 y 300. Se sabe que las imágenes hiperespectrales proporcionar mayor información para mejorar la clasificación por tener una firma espectral más específica en cada pixel [3] y [4]. En este trabajo se clasifican cuatro clases de cobertura de suelo, usando imágenes hiperespectrales. Se usan tres filtros con tres variantes, además se hace el análisis sin utilizar filtro.

El resto del paper está organizado de la siguiente manera: Sección 2 trabajos relacionados del tema tratado. La Sección 3 Metodología donde se exponen los algoritmos y métodos usados. En la Sección 4 se presentan los Resultados. En la Sección 5 se encuentra la Discusión del experimento y finalmente en la Sección 6 las Conclusiones del experimento.

2 Trabajos previos

Usan [5] imágenes de radar ERS y RADARSAT para clasificar: pasturas en uso, pasturas abandonadas, terrenos reforestados de 3 a 8 años, terrenos reforestados con árboles de 9 a 13 años, terrenos de reforestación con más de 13 años y bosque con vegetación

primaria. Logrando una precisión de 80% en las imágenes ERS y un 70% en las RADARSAT, emplean una ventana con movimientos y estadísticos de texturas, usando un clasificador clustering de máxima verisimilitud (maximum likelihood).

Con imágenes de satélite IKONOS [1], realiza la clasificación de las siguientes especies: conífera cryptomeria, ciprés hinoki, pino rojo, áreas de pino muerto, karamatsu, roble, castaño y bambú. El experimento usó dos variantes, en la primera se usaron valores del espectro, donde se obtuvieron precisiones de 25.4 % a 99.7 %. En la segunda se utilizaron valores espectrales, además, de valores derivados de las texturas de la imagen RGB, donde obtuvieron precisiones de 30.6% a 100%. En este trabajo, es importante destacar que se tienen clases diversas, con texturas muy parecidas. Lo cual genera precisiones bajas para los casos donde las texturas son similares.

No solo es necesario probar las técnicas de clasificación sino que [6] genera una nueva textura basada en el dominio de Fourier con el fin de obtener un gradiente y un sentido para realizar la clasificación entre viñedos, bosques, terrenos de cultivo y pomares, logrando precisiones del 81% al 95%.

Con el fin de no solo usar los valores de los píxeles, [2], emplean una red neuronal artificial difusa (Fuzzy ARTMAP neural classifier) de acuerdo a [7], introduciendo también el valor de la elevación en la imagen, logrando una precisión desde 73% hasta 97% al separar varios tipos de árboles entre ellos pino rojo y blanco. En estos resultados se observa que las redes neuronales podrían jugar un papel muy importante en la clasificación vegetal.

Con el fin de poder automatizar la clasificación [8] proponen una técnica para segmentar imágenes, empleando la segmentación con base en la media J (JSEG) referirse a [9], para clasificar bosque, cultivos, zonas urbanas y agua, empleando texturas y la transformada de Wavelet, logrando precisiones desde 85% hasta el 95%. En este trabajo se tienen buenas precisiones, pero es conveniente resaltar que se clasificaron clases muy contrastante, a diferencia de los trabajos donde se trata de clasificar árboles que a la vista son clases sumamente parecidos.

En otro trabajo, con el objetivo clasificar los bosques de coníferas de las especies *Picea crassifolia*. *Sabina przewalskii* y pastos [10] propone usar el índice de biomasa aérea,

considerando datos de texturas y del espectro de imágenes adquiridas por el satélite LiRDA, logrando una precisión de 87.54%.

Trabajando con el mismo índice de biomasa aérea [11], busca emplear la técnica de k distribuciones descritas en [12], con el objetivo de separar cinco clases de árboles: coníferas en general, la especie particular de coníferas Larch, Pino rojo y Abeto, logrando una precisión promedio de 85%.

Con el fin de buscar separar áreas en una imagen [13] trata de clasificar las siguientes clases: terraza urbanizada, nubes, árbol caducifolio, bosque latifolio, bosque mixto de árboles caducifolio, suelo desnudo, bosque de coníferas de hoja perenne, bosque mixto de árboles de hoja perenne, prados, sombras y agua. Para este trabajo se usaron los índices normalizados de vegetación, el índice normalizado de agua, obtuvieron un promedio de 92% y con un índice Kappa de .89.

Hasta el momento las técnicas solo generan imágenes con resultados planos, por esto [14] propone el uso de un mapa 2d, en el cual los datos de entrenamiento son obtenidos por un escalador, el cual irá dando información de lo que clasifica incluyendo la altura donde se encuentra, estos datos se dividen en regiones dependiendo la altura y el tipo de terreno, después estos datos son clasificados con la técnica de máxima verisimilitud, obteniendo un promedio de precisión del 80% y teniendo como resultado un mapa 2d.

Con la ayuda de una máquina de soporte vectorial y empleando la textura de Haralick, promedio y desviación estándar [15] logra una precisión de 50%. Con las siguientes especies vegetales: bosque virgen, bosque templado caducifolio, bambú plantaciones, bosque de melaleuca, pasto de la sabana y manglares. Cabe mencionar que se usaron imágenes infrarrojas por lo que solo tienen una banda con datos útiles, esto hace que la precisión sea muy pobre.

Retomando la transformada de Wavelet [16] introduce el índice de biomasa aérea (AGB) descrita en [17], y los índices de vegetación, en imágenes de satélite SPOT y LandSat con la tecnología de mapeo térmico avanzado de alta resolución radiométrico (AVHRR), logrando una precisión del 75%, al clasificar los árboles: beech (*Fagus sylvatica*)

hornbeam (*Carpinus betulus*) alder (*Alnus glutinosa*) oak (*Quercus castaneafolia*), maple (*Acer veltonia*) ironwood (*Parotia pérsica*).

Se ha avanzado en la precisión de los resultados, sin embargo, se considera que el uso de imágenes hiperespectrales (63 bandas), en conjunto con técnicas de filtrado por vecindarios y el uso de SVM con un Kernel perceptron multicapa como clasificador. Proporciona una mejor precisión.

3 Materiales y Métodos.

Materiales.

Se emplea una imagen hiperespectral de 63 bandas, 512 píxeles de ancho por 512 píxeles de largo, la imagen fue obtenida mediante la tecnología hiperespectral AHS-ATM explicada en [18]. Esta imagen será conocida como IMG_63, se encuentran 15 clases.

Con fines de validación experimental además de la imagen hiperespectral se tiene una imagen RGB llamada resRGB en la cual se detallan los límites de cada una de las clases que existen.

La computadora empleada cuenta con: procesador AMD FX 6300 Black Edition a 3.5 GHz, 16 Gb en RAM y un disco duro de 2 TB.

Definición de la variable

Se decidió usar al pixel como variable fundamental en este trabajo, esto soportado por el estudio realizado por [19] que demostró que al trabajar con píxeles de forma individual se generan resultados favorables. También [3] y [4] comentan que un pixel en una imagen hiperespectral tiene firma espectral única que bien puede ser usado para clasificar una imagen. Cabe destacar que no solo se usa un pixel ya que se usan filtros donde se contemplan los valores de los píxeles vecinos. Los valores de los píxeles son modificados acorde con un filtro por vecindario. Generando para cada experimento un vector experimental único compuesto por las 63 bandas de la imagen hiperespectral.

Caracterización de la imagen experimental

Se empleó en el estudio la imagen de [20], pero debido a las características del equipo de cómputo y en virtud de que la imagen completa no pudo ser procesada por falta de

memoria y desbordamiento de pila, se decidió dividir la imagen IMG_63, en 3 partes quedando con las siguientes dimensiones 172x512, 172x512 y 168x512 pixeles. De estas tres imágenes por cuestiones de tiempo de procesamiento de datos, se decidió trabajar con la imagen 1 de 172x512 pixeles que contiene 4 clases; Avena (café), Cebolla (gris), Experimental plot (azul) y Sin vegetación (verde), las aéreas de color negro no tienen interés. Esta imagen es llamada IMG_63_1, ver figura 1, (arriba imagen original y abajo imagen RGB con las clases ya caracterizada por colores). Esta imagen fue procesada en tres etapas nombradas: ConjuntosF, Conjuntos4Maquina y Maquina.

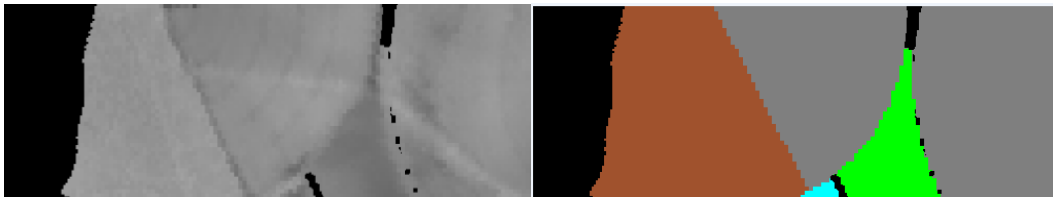


Figura 1. Arriba imagen experimental IMG_63_1 usada en el experimento, abajo en RGB clases ya clasificadas.

Etapas ConjuntosF.

Aquí la imagen es caracterizada pixel por pixel, eliminando los pixeles que no son de interés, creando una meta-matriz (Es una matriz que almacena solo información referente a otras matrices) de nombre etiquetaExisteResRgb que almacene al número de clases, los colores de cada clase, el número de pixeles que presenta cada clase y la cantidad de conjuntos que se crearan para esta clase. Del mismo modo se genera una índice-matriz (indexa otra matriz) de nombre matClaseResRgb en la cual, se guardan las coordenadas de cada pixel, la clase a la que pertenece y al conjunto que pertenece. Una vez que se tiene el índice matriz de todos los pixeles, se inicia el proceso de filtrado.

Definición del filtro.

Al trabajar con imágenes es conveniente filtrar las imágenes para normalizar los datos, y con ello mejorar las técnicas de clasificación y segmentación.

Uno de los filtros más usados es el filtro Gaussiano que se explica en [21] y [22] donde demuestran que al usar filtros antes de clasificar los pixeles (individuales o agrupados en texturas) mejoran la clasificación, otro filtro usado es el de Gabor explicado en [23] y [24]

en dichos trabajos se demuestra que los resultados son equiparables a los obtenidos en el del filtro Gaussiano.

De acuerdo a [25] y [26], para trabajar con un filtro Gaussiano empleando texturas es necesario dividir la imagen en ventanas. En esta investigación en vez de usar ventanas se emplean 3 topologías de distribución de pixeles: cuadrado, cruz y estrella (ver Figura 2) de acuerdo a lo propuesto por los parámetros de Markov referidos en [27], con la fórmula 1.

$$s = a(g_l + g_r) + b (g_t + g_b) \quad (1)$$

Donde g_l, g_r, g_t y g_b son los pixeles que se encuentran ubicados a partir del pixel central; el de arriba, abajo, izquierda y derecha, a y b son los parámetros de Markov, con magnitud 3. (Figura2). Cada filtro maneja una magnitud, que varía entre 1, 3 y 5.

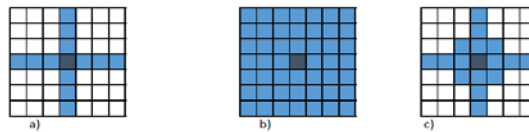


Figura 2.: Filtro con magnitud 3 a) Cruz. b) Cuadrado c) estrella. Los pixeles coloreados son los considerados.

En este trabajo cada pixel sufre una transformación, con base en un promedio de su vecindario, este valor se encuentra dado por la fórmula dos.

$$X_{new} = \frac{\sum_{i=1}^{tF} \frac{X_{old} + X_i}{2}}{tF} \quad (2)$$

Donde

tF= total de pixeles que conforman la figura sin incluir el pixel de interés.

Xold = pixel de interés.

Xnew= nuevo valor del pixel de interés.

X_i = pixeles vecinos al pixel de interés sin contar al pixel de interés.

El resultado del filtro es una nueva imagen hiperespectral, pero en la cual todos sus pixeles son afectados por el filtro obteniendo una nueva imagen IMG_63_FPT

Etapa Conjuntos4Maquina

La imagen IMG_63_FPT es dividida en 20 conjuntos, derivados de cada una de las clases, de esta manera si la imagen tiene 4 clases se obtienen 80 conjuntos, estos conjuntos son generados de forma aleatoria, el algoritmo de selección de pixeles distribuye los pixeles en subconjuntos con 20 pixeles, como mínimo, sin importar que para una clase con pocos pixeles no se llegue a los 20 conjuntos.

Conjunto de entrenamiento y prueba

Cada clase proporciona el 20 % de los subconjuntos de forma aleatoria para agregarlos a un súper-conjunto de entrenamiento y el 80 % restante es agregado a un súper-conjunto de prueba (estos 2 conjuntos albergan todos los pixeles de la imagen a analizar), si la clase solo cuenta con 1 subconjunto, este se agrega al súper-conjunto de entrenamiento, una vez que se tiene el súper-conjunto de entrenamiento este es desordenado para evitar que la técnica de clasificación se quede en un mínimo local.

Etapa Maquinas

El súper conjunto de entrenamiento se utiliza para entrenar a la SVM, para generar la clasificación de los pixeles en las diferentes clases que componen la imagen.

Clasificación de la imagen

Máquinas de soporte vector

Las máquina de soporte vector (Support Vector Machine SVM) es una técnica de clasificación que permiten separa solo 2 clases a la vez, esta forma de clasificar es llamada de tipo binario. El trabajo de SVM es clasificar buscando el mejor hiper-plano de separación que pueda existir entre las clase, el hiper plano se obtiene de maximizar la distancia que existe entre una clase y otra clase. De esta forma la SVM busca el margen más grande que existe entre 2 clases sin que existan puntos intermedios dentro de este margen, el margen se considera como la distancia máxima de los dos puntos que corren en paralelo al hiper-plano, estos puntos son llamados vectores de soporte [28].

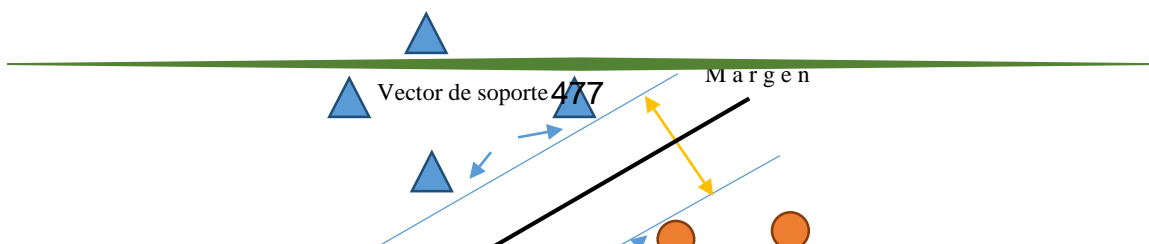


Figura 3 Diagrama de las partes de una SVM.

En la figura 3. Los triángulos representan la primera clase, los cuadrados representan la segunda clase, el hiper-plano de separación se encuentra en el centro y los vectores de soporte se encuentran en paralelo al hiper-plano y los vectores de soporte crean un margen de separación.

El SVM es una técnica de clasificación supervisada, por lo cual, debe de entrenar con un conjunto de puntos (vectores) x_i , junto con la clase y_i . Los puntos x_i contienen d dimensiones, cada vector está definido dentro de $x_i \in \mathbb{R}^d$, y su clase está definida por: $y_i = \pm 1$. Con esta información se tiene la siguiente ecuación del hiper-plano [29].

$$f(x) = (w \cdot x) + b \quad (3)$$

Donde $w \in \mathbb{R}^t$ además de que w, x y b pertenecen a números reales.

Para obtener el mejor hiper-plano de separación es necesario encontrar a w y b , esto se logra minimizando a $\|w\|$ de tal manera que para todos los puntos (x_i, y_i) .

$$f(x) = y_i(w \cdot x_i) + b \geq 1 \quad (4)$$

Los vectores de soporte x_i son aquellos para los que cumplan con la siguiente ecuación:

$$y_i(w \cdot x_i) + b = 1 \quad (5)$$

Por conveniencia matemática, el problema se puede generalizar al minimizar a $\frac{w \cdot w}{2}$. Este es un problema de programación cuadrática. La solución óptima w, b permite la clasificación de un vector z de la siguiente manera:

$$\text{Clase}(z) = \text{signo}((w \cdot z) + b) \quad (6)$$

Donde $z \in \mathbb{R}^d$ y la clase depende del signo resultante de esta ecuación.

Para resolver este problema cuadrático es computacionalmente más simple resolverlo de manera dual. Para obtener la dualidad se emplea el positivo de Lagrange y el multiplicador α_i , para multiplicarlo con la ecuación 7 y restar la función objetivo.

$$L_P = \frac{1}{2}(w \cdot w) - \sum_i \alpha_i (y_i((w \cdot x_i) + b) - 1) \quad (7)$$

Para buscar un punto estacionario L_P sobre w, b . Se establece el gradiente de L_P igual a 0, para obtener:

$$w = \sum_i \alpha_i y_i x_i \quad (8)$$

$$0 = \sum_i \alpha_i y_i$$

Gradiente $L_P = 0$ en problema por dualidad.

Sustituyendo dentro de L_P , se obtiene la dual L_D :

$$L_D = \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) \quad (9)$$

Despeje en L_P para obtener L_D .

Cuando se maximiza sobre $\alpha \geq 0$. En general, cualquier α_i que es 0 es el máximo. Las α_i diferentes a 0 dentro del problema dual define el hiper-plano lo que da a w como la suma de $\alpha_i y_i x_i$. De esta manera los valores donde x_i corresponden a una $\alpha_i \neq 0$ son los vectores de soporte.

Al derivar L_D con respecto a $\alpha_i \neq 0$, es 0 un grado óptimo. De esta manera se tiene como resultado:

$$y_i((w \cdot x_i) + b) - 1 = 0 \quad (10)$$

Derivada de L_D con respecto a $\alpha_i \neq 0$.

En partícula, esto le da el valor de b a la solución, mediante la adopción de cualquier i con un $\alpha_i \neq 0$.

SVM para multclasificación.

Para realizar la multclasificación es posible pensar que SVM es una técnica binaria, esto se puede solucionar empleando uno contra todos (one-against-all OAA). La formulación

inicial del método de OAA requiere la unanimidad entre todos los SVM's: un punto de datos sería clasificado bajo una cierta clase, si y sólo si, SVM de esa clase lo aceptó y SVM de todas las demás clases lo rechazan. [30].

Kernel.

Algunos problemas de clasificación binarios no tienen un hiper-plano simple como un criterio de separación útil. Para estos casos hay una variante del enfoque matemático Llamada kernel el cual transforma los datos en un espacio de función F , permitiendo que los datos sean separables. Para que los datos sean transformados a un espacio F , es considerada una función de mapeo ϕ , esta opera desde un espacio de entrada X , convirtiendo los datos en una función de espacio Hilber de acuerdo a [31].

$$\begin{aligned} \phi: X &\rightarrow F & (11) \\ x &\rightarrow \phi(x) \end{aligned}$$

Considerando que m es el número de muestras, con el cual se define la matriz del kernel k basada en el producto punto:

$$K = (k_{ij})_{1 \leq i \leq m, 1 \leq j \leq m} \quad (12)$$

Donde $k_{ij} = \phi^t(x_i)\phi(x_j)$

La dimensión $m \times m$ define a la K matriz con elementos $K_{ij} = k(x_i, x_j)$ y es llamado matriz kernel o matriz Gram.

La matriz kernel es simétrica y positiva semi-definida por lo tanto para todo $a_i \in \mathbb{R}, i = 1, \dots, m$, con esto se tiene:

$$\sum_{i,j=1}^m a_i a_j K_{ij} \geq 0 \quad (13)$$

De manera gráfica el kernel se encarga de adaptar los datos para permitir una clasificación lineal y simple.

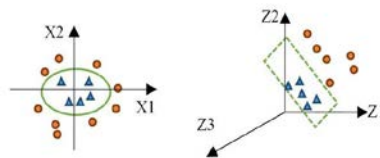


Figura 4 Ejemplo grafico de un kernel

En la figura 4. Se ve un ejemplo grafico de como un kernel genera un nuevo espacio F para permitir una clasificación por un hiper-plano, separando los triángulos de los círculos.

Procedimiento para obtener la clasificación

El súper-conjunto de entrenamiento es introducido a la técnica de clasificación, en este caso la técnica de clasificación es un máquina de soporte vectorial con un kernel dado por una perceptron multicapa de acuerdo a [32], el algoritmo Máquina busca los parámetros óptimos del kernel, estos se encuentran dados por una combinación que va desde 0 hasta 1 y desde 0 hasta -1 incrementando

estos dos parámetros en 0.1, por lo cual el algoritmo Máquina realiza 100 SVM por cada clase, empleando la técnica uno contra todos SVM (OAA) como lo recomiendan en [26]. Por cada clase se tiene como resultado un vector de soporte, este grupo de vectores de soporte son ocupados para etiquetar un mapa de clases, el cual, pasa por el algoritmo de DibujaRGB para colorear la imagen y obtener la precisión.

Todo este procedimiento desde la selección de pixeles, el algoritmo Máquina y la obtención de la precisión se repite 20 veces, por cada combinación de filtro de prueba y magnitud ya definidos. Con lo cual se hacen 180 ejecuciones, agregando 20 ejecuciones más donde, los datos no han pasado por ningún filtro, haciendo en total 200 ejecuciones tardando un tiempo de 30 días.

El criterio de selección de la SVM óptima está dado por la precisión el cual esta expresado en la ecuación 14.

$$\text{pre} = \frac{NA}{NA+NE} \quad (14)$$

Dónde:

pre = precisión

NA = número de aciertos.

NE= número de errores.

4 Resultados

En la Tabla 1 se muestran los resultados de precisión obtenidos por los 9 filtros que se emplearon en los experimento, También se presenta la prisi3n si no se usara un filtro.

Tabla 1 Precisiones obtenidas

No de pixeles	Clases	Precisi3n									
		Cruz 1	Cruz 3	Cruz 5	Cuadrado 1	Cuadrado 3	Cuadrado 5	Estrella 1	Estrella 3	Estrella 5	Sin filtro
24954	Avena	0.92	0.92	0.92	0.92	0.91	0.92	0.91	0.92	0.92	0.91
44143	Cebolla	0.71	0.71	0.70	0.70	0.71	0.70	0.71	0.71	0.71	0.71
48	Experimental Plot	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99
5288	Sin Vegetaci3n	0.93	0.92	0.92	0.93	0.93	0.92	0.92	0.93	0.92	0.92

En la Tabla 2. Muestra las medias y desviaci3n est3ndar que hay de las precisiones entre los diferentes filtros de la misma clase

Tabla 2. Media y desviaci3n est3ndar de los filtros

Clase	\bar{x}	S
Avena	0.9220	0.0032
Cebolla	0.7116	0.0032
Experimental Plot	0.9992	0.0005
Sin Vegetaci3n	0.9290	0.0038

Tabla 3. Sensibilidad (S) y especificidad (E) de los resultados obtenidos

Filtros	Avena		Cebolla		Experimental Plot		Sin Vegetaci3n	
	S	E	S	E	S	E	S	E
Cruz 1	0.876	0.646	0.638	0.785	1.000	0.001	0.900	0.046
Cruz 3	0.874	0.639	0.628	0.783	0.999	0.001	0.955	0.159
Cruz 5	0.875	0.642	0.627	0.783	0.999	0.000	0.955	0.159

Cuadrado 1	0.873	0.633	0.625	0.784	0.999	0.000	0.903	0.047
Cuadrado 3	0.875	0.640	0.637	0.783	0.999	0.001	0.960	0.181
Cuadrado 5	0.877	0.643	0.637	0.790	0.999	0.001	0.898	0.044
Estrella 1	0.875	0.632	0.623	0.781	0.999	0.000	0.900	0.048
Estrella 3	0.873	0.641	0.608	0.768	0.999	0.001	0.900	0.046
Estrella 5	0.877	0.652	0.639	0.787	0.999	0.001	0.948	0.139
Sin filtro	0.876	0.642	0.623	0.777	0.999	0.000	0.901	0.049

La sensibilidad se define como:

$$Sensibilidad = \frac{VP}{VP+FN} \quad (15)$$

Donde VP es verdaderos positivos y FN falsos negativos.

La especificidad se define como:

$$Especificidad = \frac{VN}{VN+FP} \quad (16)$$

Donde VN, serían los verdaderos negativos; y FP, los falsos positivos.

En la Tabla 3. Muestra la sensibilidad (S) y especificidad (E) de los resultados obtenidos en las todas las pruebas.

5 Discusión.

En cuanto a los resultados obtenidos por los tres filtros propuestos y las tres magnitudes respectivas, los datos sugieren que no hay diferencias significativas para aseverar que algún filtro en particular es mejor que otro. Y más importante es destacar que el experimento que se realizó sin filtro tiene resultados iguales que cuando se usa un filtro. Ver figura 4.

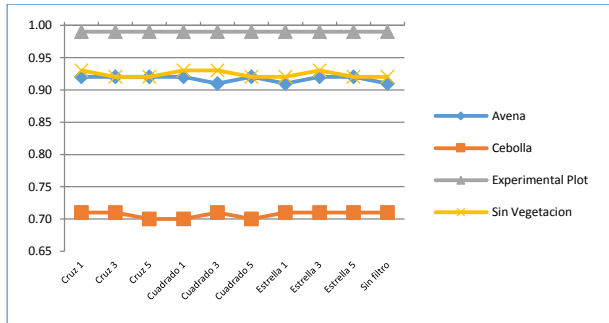


Figura 4 Diagrama de las partes de una SVM.

En la figura 4 se pueden observar las precisiones que se obtuvieron de cada filtro y clase a clasificar donde se observa claramente la poca variación que existe entre el valor de precisión obtenido por cada filtro.

Las desviaciones estándar que hay de las precisiones entre los diferentes filtros de la misma clase están por debajo de 0.0039. Esto es importante ya que estos resultados sugieren que no hay diferencias en las precisiones al utilizar un filtro determinado y más importante el experimento muestra que sin utilizar un filtro los resultados son prácticamente los mismos, la importancia de esto es que el filtro consume recursos computacionales y si estos son omitidos el tiempo de análisis se reduce considerablemente.

Los resultados que se presentan en la Tabla 1 hay que comentar que la precisión más alta de 0.99 del Experimental plot no es relevante debido a que este grupo cuenta con solo 48 pixeles, lo cual provoca un sesgo en la SVM, indicando que todos los pixeles no pertenecen y de esta manera logra una precisión casi perfecta.

La precisión que se obtuvo en un suelo sin vegetación fue en promedio de 0.92, estos resultados son reconocidos como los mejores, sin embargo, hay que considerar que un suelo sin vegetación no tiene una reluctancia de la luz muy discontinua, lo que explica la alta precisión. La precisión en la avena aunque en promedio es de 0.91 es equiparable a la de sin vegetación, esto es de llamar la atención ya que aquí la reflectancia si tiene variantes por combinar colores. Por otro lado al identificar cebolla la precisión en promedio es de 0.7, este resultado es muy bajo en comparación con los dos anteriores.

Pero al compararlos con los resultados presentados en [1], [10], [11], [13], [14], [15] y [16], los obtenidos mejoran las precisiones. El valor más bajo obtenido en este experimento fue de 70% y el más bajo en los experimentos mencionados fue de 25.4 %. Se mejora en un 44%. Sin embargo el 70% no es un resultado deseable, este resultado puede estar asociado a la técnica usada para multclasificación ya que la técnica SVM- OAA genera regiones de espacio de características de clasificación indecisas, debido a que los pixeles fueron clasificados por diferentes SVM's como pertenecientes a varias clases, esto también lo comenta [33], además [33] afirma que se genera un mapa con las clases encontradas con espacios de clasificación indecisa. Se considera que para poder mejorar las precisiones y contrarrestar la desventaja de la técnica SVM- OAA, habría que cambiar la máquina de aprendizaje con la técnica uno contra uno (one against one OAO) como lo sugiere [30].

Con respecto al uso de memoria RAM en la investigación se proponen 2 mejoras, la primera la adquisición de más memoria RAM y la segunda generar un algoritmo para elegir los puntos de clasificación, buscando un ajuste en estos para que los pixeles de entrenamiento sean más descriptivos por clase, además de permitir remplazos y usar lo mínimo de memoria RAM.

En la tabla 3 muestra los valores que se obtuvieron de los estadísticos sensibilidad y especificidad como indicadores que evalúan el grado de eficacia inherente a una prueba de clasificación. Considerando los resultados se puede decir que las clasificaciones no son las más adecuadas en cebolla y avena, para clasificar cobertura de suelo sin vegetación son aceptables. La variable experimental plot debe ser descartada por tener muy pocos pixeles para la clasificación. En general se puede decir que las SVM en multclasificación no son las más adecuadas para este tipo de trabajo.

6 Conclusiones.

Los pixeles derivados de imágenes hiper-espectrales, tienen características deseables para clasificar la cobertura vegetal del suelo.

La más baja precisión fue de 70% que al parecer no es buena, pero si se compara con las reportadas en la literatura se considera que mejoro a otras, sin embargo es necesario mejorarla.

La técnica OAA-SVM, no es adecuada para etiquetar imágenes con respecto a la cobertura de suelo, debido a que la cobertura de suelo contiene una gran cantidad de etiquetas, y genera un espacio de características indeciso al momento de etiquetar, quedando un pixel clasificado en varias clases o en ninguna. Se sugiere repetir este experimento con la técnica uno contra uno (one against one OAO) como lo sugiere [30].

Se recomienda repetir el experimento en más imágenes, para determinar si el uso de filtros puede ser excluido y mejorar los tiempos de uso de computadora.

Se requiere repetir este experimento buscando mejorar la precisión de las clases desfavorecidas en este experimento.

Es conveniente explorar otros clasificadores (redes neuronales artificiales con nuevas técnicas de aprendizaje), para evaluar la conveniencia de usar SVM.

7 Referencias bibliográficas

- [1] M. Kawamura, K. Tsujino y Y. Tsujiko, Characteristic Analysis of High Resolution satellite Imagery for Forest Species Discrimination, Japon: IEEE, 2004.
- [2] M. L. Pugh y A. M. Waxman, Classification of Spectrally-Similar Land Cover Using Multi-Spectral Neural Image Fusion and the Fuzzy ARTMAP Neural Classifier, New York, USA: IEEE, 2006.
- [3] M. Rojas, Caracterización de imágenes hiperespectrales utilizando Support Vector Machines y técnicas de extracción de características, España, 2009.
- [4] A. F. Paz, Procesamiento paralelo de imágenes hiperespectrales: Sistemasmulti, 2009.

- [5] C. d. C. Freitas, W. P. S.-J. Lelio y L. V. Dutra, Assessment of Operational Radar Satellite for Monitoring Land Cover Change in Amazonia, San Jose dos Campos SP, Brazil.: CNP, 2000.
- [6] R. Trias-Sanz, A Texture Orientation Estimator for Discriminating Between Forests, Orchards, Vineyards, and Tilled Fields, Paris, France.: IEEE, 2005.
- [7] G. A. Garpenfer, S. Grossberg, N. Markuzon, J. H. Reynolds y D. B. Rosen, Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps, Miami, USA: IEEE, 1991.
- [8] Z. Wang y R. Boesch, Color- and Texture-Based Image Segmentation for Improved Forest Delineation, Birmensdorf, Switzerland: IEEE, 2007.
- [9] Y. Deng y B. S. Manjunath, Unsupervised Segmentation of Color-Texture Regions in Images and Video, California, USA: PAMI, 2001.
- [10] H. Qisheng, Estimation of coniferous forest above-ground biomass using LiDAR and SPOT-5 data, Nanjing, China: National Natural Science Foundation of China, 2009.
- [11] W. Haipeng y O. Kazuo, A Simple Moment Method of Forest Biomass Estimation From Non-Gaussian Texture Information by High-Resolution Polarimetric SAR, Shanghai, China: IEEE, 2010.
- [12] E. Jakeman, On the statistics of K-distributed noise, Great Britain: e-Journal, 1980.
- [13] W. Lin, Q. Wang y S. Zha, Construction and Application of Characteristic Bands of typical Land Cover based on Spectrum-photometric Method, Wuhan, China: National Science Foundation of China, 2010.
- [14] J. Sanofer y p. R. Dee, Detection of Individual Tree Crowns in High Spatial Resolution using 2D Walking Ant Histogram, TamilNadu, India: Proceedings of the International Conference on Communication and Computational Intelligence, 2010.

- [15] H. Benelcadi, P.-L. Frison, C. Lardeux, A.-C. Capel, J.-B. Routier y J.-P. Rudant, CONTRIBUTION OF TERRASAR-X RADAR IMAGES TEXTURE FOR FOREST MONITORING, Paris, Francia: IEEE, 2012.
- [16] N. Ghasemi, M. R. Sahebi y A. Mohammadzadeh, Biomass Estimation of a Temperate Deciduous Forest Using Wavelet Analysis, IEEE, 2013.
- [17] J. Hyypää, H. Hyypää, M. Inkinena, M. Engdahla, S. Linkob y Y.-H. Zhuc, Accuracy comparison of various remote sensing data sources in the retrieval of forest stand attributes, Espoo, Finlandia: ScienceDirect, 2000.
- [18] P.-S. a. P. García-Sevilla, "Fucion of Multi-temporal and Multi-sensor Hyperspectral Data for Land-Use Classification," *Lecture Notes in Computer Science*, vol. 7887, pp. 724-731, Junio 2013.
- [19] A. Srivastava, A. Asati y M. Bhattacharya, Fast Hybrid Rough-Set Theoretic Fuzzy Clustering Technique with application to Multispectral Image Segmentation, Allahabad, Uttar Pradesh, India: ACM, 2010.
- [20] I. Piqueras-Salazar y P. García-Sevilla, «Fusion of Multi-temporal and Multi-sensor Hyperspectral Data for Land-Use Classification,» *Lecture Notes in Computer Science*, vol. 7887, pp. 724-731, June 2013.
- [21] J. F. Camapum y M. H. Fisher, Multiscale Color and Texture Invariants for Image Recognition, Brasilia, Brasil.: IEEE., 2001.
- [22] F. Sarría, A. Recio y J. Ruiz, Análisis de Imagenes Mediante Texturas: Aplicacion a la Clasificacion de Unidades de Vegetación, Valencia, España.: Departamento de Ingeniería Cartográfica, Geodesia y Fotogrametría., 2003.
- [23] W. Ali, F. Georgsson y T. Hellstrom, Visual Tree Detection for Autonomous Navigation in Forest Environment, Umea, Sweden.: IEEE., 2008.

- [24] Y. Yang y S. Newsam, Bag-Of-Visual-Words and Spatial Extensions for Land-Use Classification, California, USA.: ACM, 2010.
- [25] C. Kim y S.-H. Hong, The Characterization of a Forest Cover Through Shape and Texture Parameters from Quickbird Imagery, Seoul, Korea.: IEEE, 2008.
- [26] N. H. Chehade, J.-G. Boureau, C. Vidal y J. Zerubia, MULTI-CLASS SVM FOR FORESTRY CLASSIFICATION, USA & France.: IEEE, 2009.
- [27] M. Petrou y P. Garcia Sevilla, Image Processing Dealing Texture, England: Jhon Wiley & Sons Ltd, 2006.
- [28] T. Zhang, An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods A Review, AI Magazine, 2001.
- [29] T. Stibor, A Study of Detecting Computer Viruses in Real-Infected Files in the n - gram Representation with Machine Learning Methods, Fakult für InformatikTechnische Universit Munche ed., Alemania, 2010.
- [30] J. Milgram, M. Cheriet y R. Sabourin, "One Against One" or "One Against All": Which One is Better for Handwriting Recognition with SVMs?, Montrial, Canada., 2004.
- [31] G. Baudat y F. Anouar, Kernel-based Methods and Function Approximation, 2001.
- [32] T. W. Rauber y K. Berns, Kernel Multilayer Perceptron, Vit´oria, ES, Brazil: University of Esp´rito Santo, 2011.
- [33] B. Aisen, A Comparison of Multiclass SVM Methods, 2006.