Análisis lineal discriminante (*LDA*)

Rafael Pérez Torres
Profesor: Dr. Wilfrido Gómez Flores,

LTI Cinvestav.

***Resumen*—El análisis lineal discriminante (*LDA*) es una técnica para la reducción de dimensionalidad que también puede ser utilizada para realizar clasificación lineal supervisada. *LDA* permite reducir la dimensionalidad de un conjunto de datos de l dimensiones proyectándolos en $c-1$ dimensiones, donde c es la cantidad de clases del conjunto de datos. Los datos proyectados pueden entonces describir una separación lineal, la cual es utilizada para realizar la clasificación de los mismos.**

En su forma general, *LDA* permite realizar clasificación binaria, sin embargo esto es suficiente para realizar incluso la clasificación multiclase al seguir un enfoque como *uno vs uno* o *uno vs todos*. El enfoque *uno vs todos* busca realizar c clasificaciones binarias en las que se obtiene el grado de pertenencia de la instancia a una clase en específico o al resto de ellas, asignando aquella clase en la que la pertenencia es mayor.

El presente documento describe la implementación de *LDA* para clasificación multiclase, utilizando el enfoque de *uno vs todos* sobre ocho datasets sintéticos, así como la presentación de los porcentajes de error obtenidos.

***Index Terms*—Reconocimiento de patrones, LDA**

I. Introducción

El análisis lineal discriminante, *LDA*, es una generalización del discriminante lineal de Fisher, creada por el estadístico, biólogo y matemático Ronald Fisher.

LDA es una técnica que puede ser empleada tanto para clasificación como para reducción de dimensionalidad de un conjunto de datos previo a la clasificación. El objetivo básico del *LDA* es la búsqueda de una combinación lineal de atributos que separen de la mejor forma a las clases.

Esta reducción de la dimensionalidad puede entenderse fácilmente observando la Figura 1. En ella se muestran dos ejemplos en el que se ha reducido la dimensionalidad de los datos, que obedecen a dos clases (roja y azul), a solamente una dimensión. Los datos son proyectados entonces de su espacio original, logrando describir sus características en apenas una sola dimensión. Sin embargo, tal como se muestra en la Subfigura 1(a), no todas las proyecciones resultan ser convenientes para realizar esta reducción ya que no se logra reflejar una mejor separabilidad de los datos. En cambio, la Subfigura 1(b) muestra un *buen* ejemplo de proyección en el que, a pesar de tener solamente una dimensión, se hace posible separar fácilmente al conjunto de datos en las dos clases originales.

Usando la misma Figura 1 es posible observar también que el *LDA* utiliza como características clave la media y varianza de cada una de las clases, intentando maximizar la relación de las varianza interclase (datos de distintas clase) así como la relación intraclase (datos de la misma clase), de tal forma que las proyecciones obtenidas posicionen a los datos en grupos *densos* y altamente separados.

Como dato adicional, es importante destacar que *LDA* asume que las distribuciones de los datos son gaussianas, obteniendo resultados no satisfactorios si los datos carecen de dicha característica.

II. Marco teórico

A través de una función de regresión lineal, como

$$f(x;w) = w^T x$$

es posible proyectar cada punto $\mathbf{x} = [x_1, x_2, ..., x_l]^T$ de un dataset hacia una línea paralela al vector de proyecciones $\mathbf{w}$ que pasa por el origen.

Entonces, variando este vector de proyecciones, o pesos si se hace la analogía con los clasificadores lineales abordados en clase, es posible obtener diferentes niveles de separación, tal y como se mostraba en las gráficas de la Figura 1. El objetivo del LDA es precisamente encontrar el vector de proyecciones $\mathbf{w}^*$ que maximice la separación entre las clases proyectadas.

De esta forma, la bondad de un vector de proyecciones viene definida por una medida de separación, como por ejemplo la diferencia entre la media de los datos de cada clase. Sin embargo, la media no siempre obtiene valores óptimos de separación por lo que se involucra otra característica de los datos tal como la varianza.

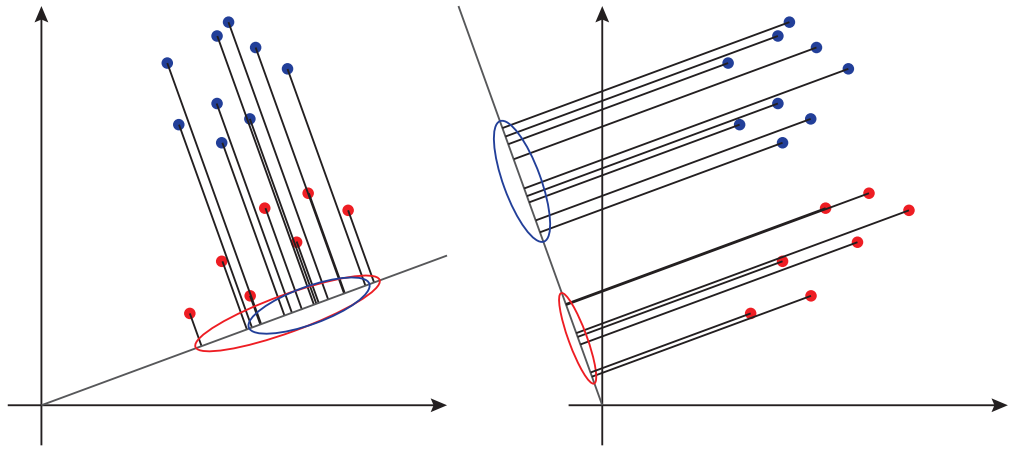
El criterio de Fisher, base para *LDA*, consiste precisamente en una función lineal $w^T x$ que considera tanto a la media como a la varianza de los datos, intentando maximizar (para dos clases) a:

$$separacion = \frac{|(\mu_1 - \mu_2)|^2}{s_1^2 + s_2^2} \tag{1}$$

donde μ_1 y μ_2 son los valores de las medias y s_1^2 y s_2^2 son los valores de las varianzas de las clases en la dimensión original.

Asumiendo que es posible encontrar el valor del vector de proyección $\mathbf{w}$, utilizar *LDA* para clasificación involucraría realizar las etapas de entrenamiento y la clasificación como tal:

- Entrenamiento: Las actividades se concentrarían en calcular el vector de proyección $\mathbf{w}$ y un bias.



(a) *Mala* proyección (b) *Buena* proyección

Figura 1. Ejemplo de proyecciones

- Clasificación. Las actividades se concentrarían en evaluar cada patrón desconocido con los vectores de proyección de cada clase y tomar aquel que arroje el valor más alto.

Las siguientes subsecciones describen el marco teórico de ambas tareas.

II-A. Entrenamiento del LDA

Ahora bien, para expresar lo anterior en términos del vector de proyección $\mathbf{w}$ es preciso reiterar que basta con *proyectar* cada punto en el nuevo espacio de dimensiones; esto se consigue multiplicando cada punto por este vector. De esta manera, la distancia entre las medias proyectadas puede ser expresada a través de:

$$|\tilde{m}_1 - \tilde{m}_2| = |\mathbf{w}^T(\mu_1 - \mu_2)| \tag{2}$$

Si se definen las matrices de dispersión S_i y S_w como:

$$S_i = \sum_{x \in \omega_i} (\mathbf{x} - \mu_i)(\mathbf{x} - \mu_i)^T \tag{3}$$

y

$$S_w = S_1 + S_2 \tag{4}$$

Entonces la varianza de los datos proyectados de cada clase ω_i puede ser expresada en términos de matrices de dispersión, como en:

$$\begin{aligned} \tilde{s}_i^2 &= \sum_{x \in \omega_i} (x - \tilde{\mu}_i)^2 && (5)\\ &= \sum_{x \in \omega_i} (\mathbf{w}^T x - \mathbf{w}^T \mu_i)^2 && (6)\\ &= \sum_{x \in \omega_i} \mathbf{w}^T (\mathbf{x} - \mu_i)(\mathbf{x} - \mu_i)^T \mathbf{w} && (7)\\ &= \mathbf{w}^T S_i \mathbf{w} && (8) \end{aligned}$$

Por lo tanto, la suma de las matrices de dispersión puede ser definida como:

$$\tilde{s}_1^2 + \tilde{s}_2^2 = \mathbf{w}^T S_w \mathbf{w} \tag{9}$$

siendo S_w conocida como la matriz de dispersión intra-clase.

Siguiendo este mismo enfoque, es posible definir la separación entre las medias de los datos proyectados:

$$\begin{aligned} (\tilde{m}_1 - \tilde{m}_2)^2 &= (\mathbf{w}^T \mu_1 - \mathbf{w}^T \mu_2)^2 && (10)\\ &= \mathbf{w}^T (\mu_1 - \mu_2)(\mu_1 - \mu_2)^T \mathbf{w} && (11)\\ &= \mathbf{w}^T S_B \mathbf{w} && (12) \end{aligned}$$

donde

$$S_B = (\mu_1 - \mu_2)(\mu_1 - \mu_2)^T \tag{13}$$

siendo S_B conocida como la matriz de dispersión inter-clase.

A través de este desarrollo matemático es posible expresar el criterio de Fisher (de la Ecuación 1) en términos de $\mathbf{w}$ como:

$$J(\mathbf{w}) = \frac{\mathbf{w}^T S_B \mathbf{w}}{\mathbf{w}^T S_W \mathbf{w}} \tag{14}$$

Ahora bien, obtener el valor óptimo de este vector de pesos puede ser realizado a través del cálculo de los eigenvalores y eigenvectores para

$$S_B \mathbf{w} = \lambda S_W \mathbf{w} \tag{15}$$

Sin embargo, para este caso particular puede ser resuelto de forma directa a través de:

$$\mathbf{w} = S_w^{-1}(\mu_1 - \mu_2) \tag{16}$$

El valor del *bias* es calculado directamente como *la media de las medias* de ambas clases:

$$\mathbf{bias} = \frac{\mu_1 + \mu_2}{2} \tag{17}$$

Hasta este punto los valores para realizar la clasificación han sido calculados. Sin embargo, es preciso notar que la clasificación sería binaria, por lo que si se requiere realizar clasificación multiclase es necesario adoptar un enfoque para definir qué datos estarían en la

clase ω_1 y en ω_2 en el clasificador, o en otras palabras conocer el origen de los datos para calcular las matrices de dispersión intraclase e interclase. Los dos enfoques disponibles son el *uno vs todos* y el *uno vs uno*.

- ***Uno vs todos***: Teniendo ω_i clases, es necesario establecer el ω_1 del clasificador como una clase en específico y el ω_2 como el resto de las clases.
- ***Uno vs uno***: Teniendo ω_i clases, es necesario establecer el ω_1 del clasificador como una clase en específico y el ω_2 como otra clase en específico.

Al implementar uno de estos enfoques las actividades del entrenamiento han sido finalizadas.

II-B. Clasificación

Dados un patrón desconocido x', los valores para el vector de proyección **w**, el **bias** y los valores de probabilidad $P(\omega_i)$ para cada una de las clases ω_i, la clasificación consiste en proyectar $\mathbf{x}'$ sobre la dirección de máxima separación y evaluar su cercanía hacia cada clase ω_i. El hiperplano de decisión queda definido como:

$$g(\mathbf{x}) = \mathbf{w}^T(\mathbf{x}' - \mathbf{bias}) + log\frac{P(\omega_1)}{P(\omega_2)} = 0 \qquad (18)$$

De tal forma que la etiqueta y' a asignar a $\mathbf{x}'$ es calculada a través de:

$$y' = \begin{cases} \omega_1 \text{ si } \mathbf{w}^T(\mathbf{x}' - \mathbf{bias}) > log\frac{P(\omega_1)}{P(\omega_2)} \\ \omega_2 \text{ en caso contrario} \end{cases} \qquad (19)$$

Si ambas clases son equiprobables, el valor de $log\frac{P(\omega_1)}{P(\omega_2)}$ sería 0.

Debido a que en esta asignación se implementa el enfoque *uno vs todos*, la clasificación debe ser lanzada para cada una de las clases ω_i, asignando la etiqueta de aquella clase de la que se obtuvo el valor más alto.

III. Metodología

La metodología para realizar la implementación de la técnica *LDA* con el enfoque *uno vs todos* es mostrada en la Figura 2, donde puede apreciarse el clásico proceso de entrenamiento previo a la clasificación.

Como también puede observarse, el proceso de entrenamiento recibe como argumentos de entrada las etiquetas ($Y_{training}$) así como los datos ($X_{training}$) pertenecientes al grupo de entrenamiento, entregando como respuesta los vectores de pesos o proyecciones (*Wi*) y los *bias* (*W0i*) pertenecientes a cada clase ω_i.

La salida del proceso de entrenamiento es utilizada como entrada para la tarea de clasificación, naturalmente especificando los datos a clasificar (X_{test}) y las etiquetas (Y_{test}) de dichos datos con la intención de obtener como respuesta las etiquetas que el clasificador ha asignado (Y_p) así como un porcentaje de error (*error*).

Se construyeron dos funciones en *Matlab* para realizar el entrenamiento y la clasificación de los datos, así como una función adicional para mostrar el espacio particionado obtenido por el clasificador en cada uno de los ocho datasets proporcionados.

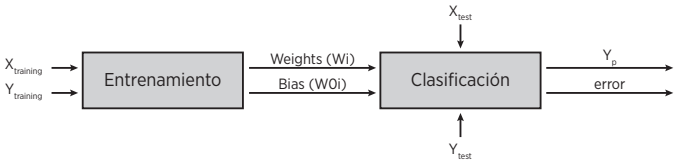


Figura 2. Metodologia

En Pseudocódigo 1 se muestra el algoritmo general para realizar el entrenamiento con *LDA*, el cual refleja los conceptos definidos en el marco teórico, específicamente en la Subsección II-A.

Pseudocódigo 1 Algoritmo de entrenamiento

Entrada: *Xtr*, *Ytr*
Salida: *Wi* (matriz de vectores de proyección), *W0i* (matriz de *bias*)
para Cada clase única ω_i en *Ytr* **hacer**
  c_inst = instancias de clase ω_i en *Xtr*
  o_inst = instancias no pertenecientes a clase ω_i en *Xtr*
  μ_1 = media de c_inst
  μ_2 = media de o_inst
  S_1 = covarianza de c_inst
  S_2 = covarianza de o_inst
  $S_w = S_1 + S_2$
  $Wi = S_w^{-1} * (\mu_1 - \mu_2)$
  $W0i = (\mu_1 + \mu_2)/2$
fin para

En Pseudocódigo 2 se muestra el algoritmo general para ejecutar la clasificación de los datos utilizando el enfoque *uno vs todos*, utilizando los conceptos definidos en la Subsección II-B.

Pseudocódigo 2 Algoritmo de clasificación

Entrada: *Xtt*, *Ytt*, *Wi*, *W0i*
Salida: *Yp*, *error* (matriz de *bias*)
para Cada instancia *i* en *Xtt* **hacer**
  para Cada clase única ω_i en *Ytt* **hacer**
    $P(\omega_i) = Wi' * (x_i - W0i(:, j))$
  fin para
  Yp_i = max($P(\omega_i)$)
fin para

Finalmente, en Pseudocódigo 3 se muestra el algoritmo utilizado para la creación de la gráfica con los espacios particionados, así como los puntos correcta e incorrectamente clasificados.

IV. Resultados

La experimentación fue realizada en un equipo de cómputo con un procesador Intel core i7 de 8 núcleos a 2.00 GHz con 6 GB de memoria en RAM. La Tabla I muestra los porcentajes de error obtenidos después de una iteración en cada uno de los 8 datasets proporcionados.

Las Figuras figs. 3 to 10 muestran el espacio particionado en cada uno de los datasets.

Pseudocódigo 3 Algoritmo de visualización

Entrada: *Xtt,Ytt,Ypred,Wi,W0i*
Salida: Una gráfica con el espacio particionado
1: $rango_x$ = crear 100 puntos entre min y max en X ($Xtt(1,:)$)
2: $rango_y$ = crear 100 puntos entre min y max en Y ($Xtt(2,:)$)
3: Crear una matriz *xy* con los puntos sintéticos
4: Convertir matriz *xy* a vector para clasificarlos con LDA, utilizando *labels* = classifyLDAmulti(*xy,c,Wi,W0i*)
5: Transformar vector *labels* a matriz (reshape)
6: Escalar y mostrar la matriz como el fondo (regiones particionadas) de la gráfica (imagesc)
7: Graficar puntos clasificados
8: Graficar los errores (puntos mal clasificados) según la diferencia entre *Ytt* y *Ypred*

Tabla I
PORCENTAJES DE ERROR OBTENIDOS POR *LDA*

Dataset	% de Error
clouds01	4.7222
clouds02	11.8
clouds03	8.08
clouds04	11.2782
clouds05	2.6667
clouds06	2.88
clouds07	5.5
clouds08	50

V. DISCUSIÓN DE RESULTADOS

Como puede observarse en la tabla de resultados y en las figuras de las regiones particionadas para cada dataset, *LDA* con el enfoque *uno vs todos* obtiene *buenos* resultados con porcentajes de error no mayores a 11.27 % (para los datasets proporcionados). Sin embargo, hay una excepción a este comportamiento al considerar el último dataset (10)), para el cual se obtiene un 50 % de error en la clasificación. Dicha situación puede ser originada debido a que las medias de las clases participantes están *mezcladas* o a que los datos de dichas clases no sigan una distribución gaussiana, como es solicitado por *LDA*.

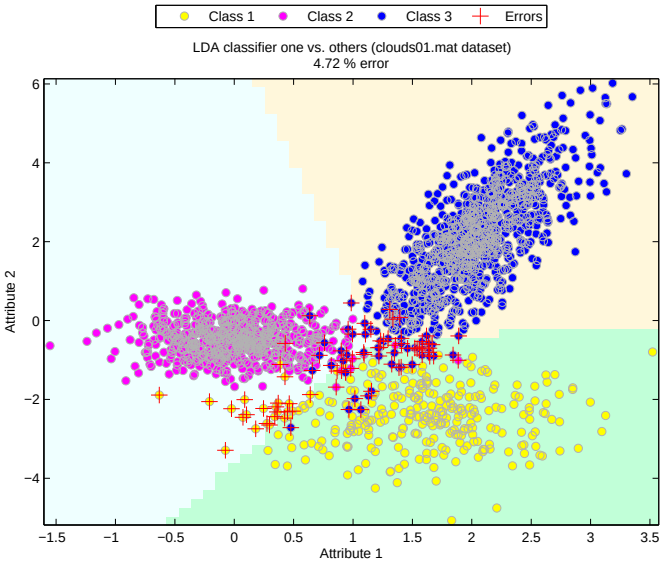


Figura 3. Espacio particionado en el dataset clouds01

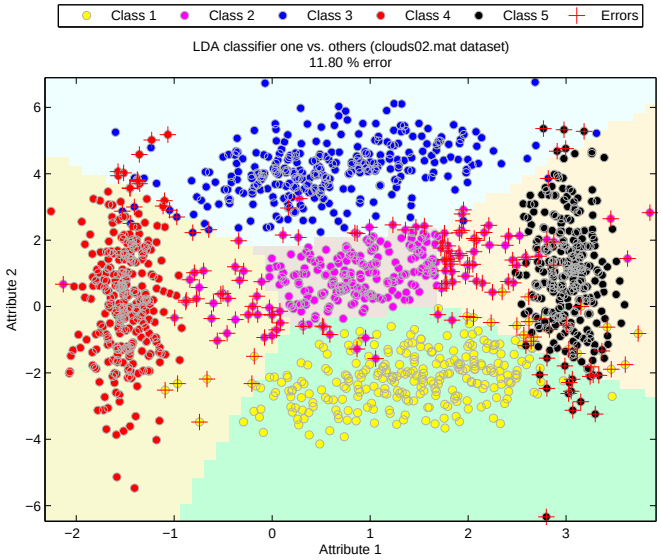


Figura 4. Espacio particionado en el dataset clouds02

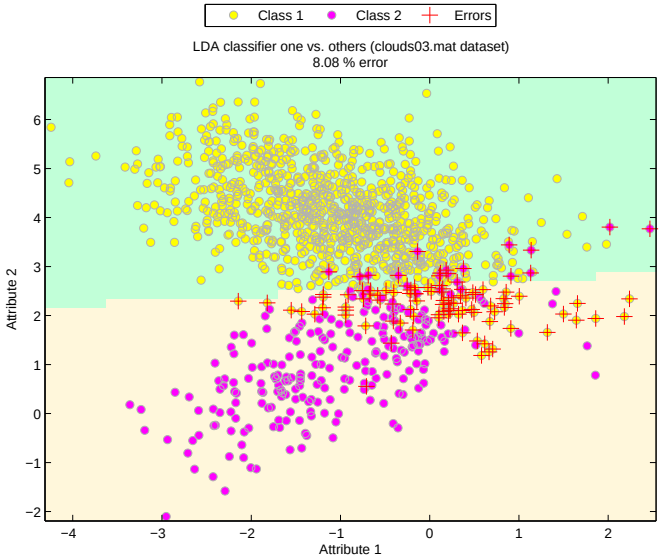


Figura 5. Espacio particionado en el dataset clouds03

Para el caso de datos linealmente separables, *LDA* ofrece resultados rápidos con el beneficio de poder proyectarlos en un espacio de menor dimensionalidad. Gracias a esto, si se requiere procesamiento adicional de los datos proyectados, ésto podría ser realizado necesitando menos poder de cómputo en comparación con los datos en su espacio de dimensiones original.

VI. CONCLUSIONES

La técnica de *LDA* permite reducir la dimensionalidad de los datos así como realizar labores de clasificación. Para esta última tarea, y considerando que *LDA*

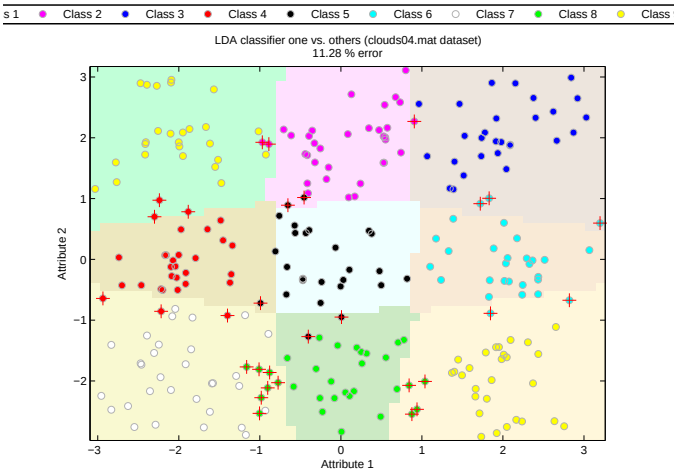


Figura 6. Espacio particionado en el dataset clouds04

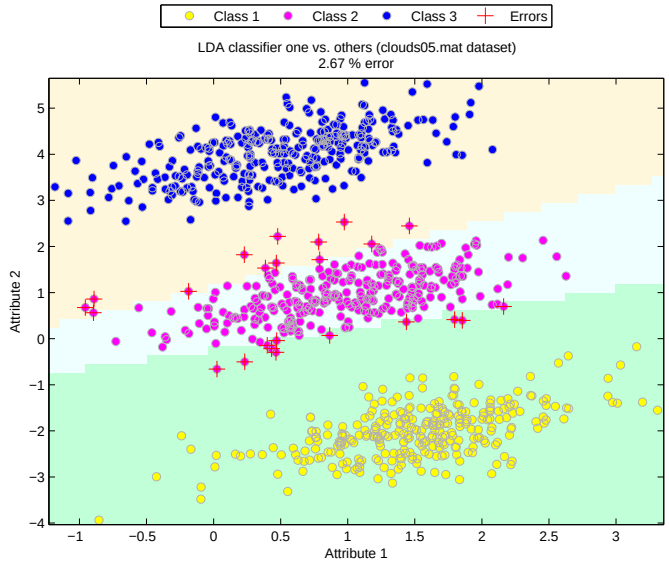


Figura 7. Espacio particionado en el dataset clouds05

proporciona clasificación binaria, es preciso seguir un enfoque que permita realizar la clasificación multiclase como el *uno vs todos* y el *uno vs uno*. Debido a que *LDA* se basa en la idea de alcanzar una separabilidad utilizando la varianza y medias de los datos, ofrecerá buenos resultados si los datos a clasificar observan una distribución gaussiana. En caso contrario, si los datos presentan valores de medias mezclados entre cada clase, los resultados no serán satisfactorios.

En este documento se ha presentado la descripción de una implementación de *LDA* para realizar la clasificación multiclase utilizando un enfoque *uno vs todos*, mostrando gráficas del espacio particionado por dicho clasificador y abordando además una pequeña discusión de sus resultados.

REFERENCIAS

[1] R. Duda, P. Hart, and D. Stork, *Pattern Classification*, 2002.

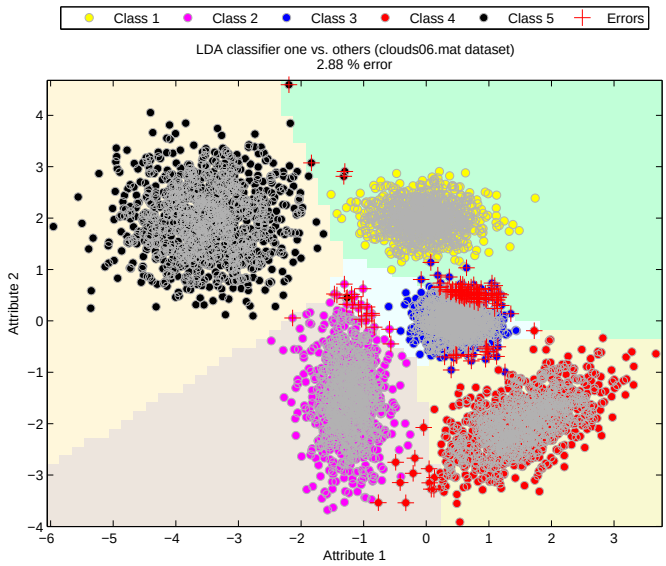


Figura 8. Espacio particionado en el dataset clouds06

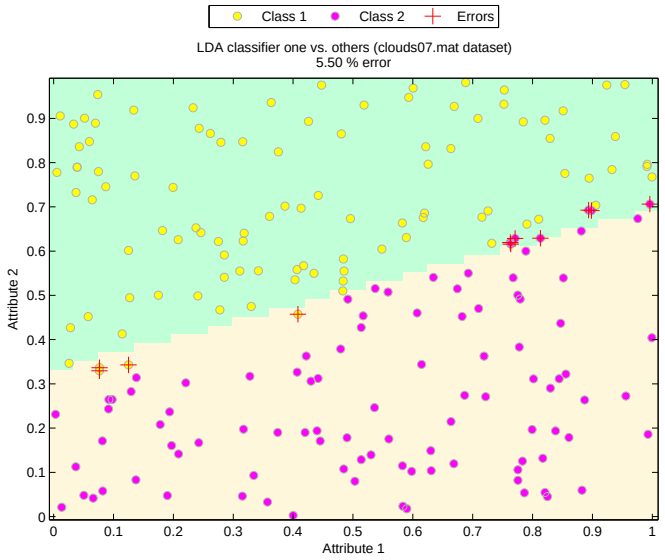


Figura 9. Espacio particionado en el dataset clouds07

[2] W. Gomez Flores, "Diapositivas de clase Linear Discriminant Analysis (LDA)," 2015.
[3] T. Li, S. Zhu, and M. Ogihara, "Using discriminant analysis for multi-class classification: an experimental investigation," *Knowledge and Information Systems*, vol. 10, no. 4, pp. 453–472, 2006. [Online]. Available: http://dx.doi.org/10.1007/s10115-006-0013-y
[4] Wikipedia, The free encyclopedia. (2015) Ronald A. Fisher. [Online]. Available: http://en.wikipedia.org/wiki/Ronald_Fisher

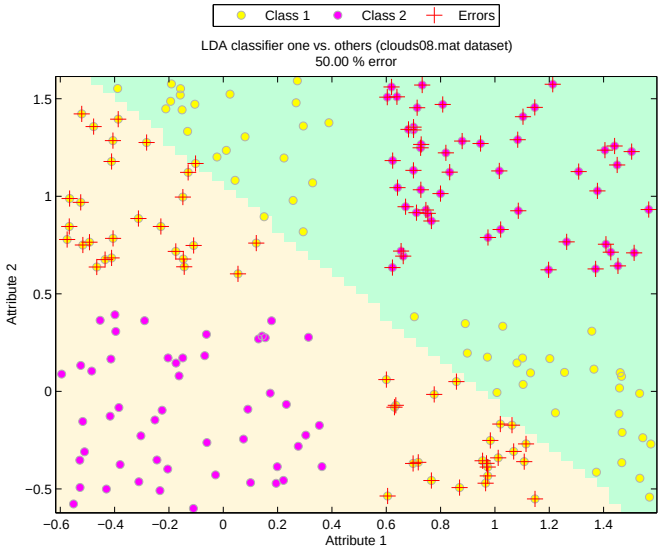


Figura 10. Espacio particionado en el dataset clouds08