

NPLS-DA

El modelo NPLS-DA, basado en el algoritmo NPLS fue generado utilizando un tensor de tercer orden construido a partir de los cubos hiperespectrales y fue transformado mediante un método de matrización. Para ello, siguiendo el procedimiento propuesto por Folch-Fortuny et al. (2016), se calcularon 5 características (media, desviación típica, simetría, curtosis, quinto momento) en cada longitud de onda obteniéndose una matriz reducida de 5 X 520 por cada cubo HS que inicialmente tenían 205 filas, 198 columnas y 520 longitudes de onda.

Número de píxeles en una imagen

$$n_p = \text{filas} \times \text{columnas} = 205 \times 198 = 40\,590$$

Media

$$\hat{x}_k = \frac{\sum x_{ik}}{n_p} \quad (1)$$

Donde

x_{ik} son los valores de reflectancia normalizada en el píxel i para una longitud de onda k.

n_p es número de píxeles en la imagen.

Desviación estándar

$$s_k = \sqrt{\frac{\sum (x_{ik} - \hat{x}_k)^2}{n_p - 1}} \quad (2)$$

Donde

x_{ik} son los valores de reflectancia normalizada en el píxel i para una longitud de onda k.

\hat{x}_k es la media de la reflectancia en la longitud de onda k.

n_p es número de pixeles en la imagen.

Simetría (tercer momento)

$$\mu_{3k} = \frac{\sum (x_{ik} - \hat{x}_k)^3}{n_p} \quad (3)$$

Donde

x_{ik} son los valores de reflectancia normalizada en el pixel i para una longitud de onda k.

\hat{x}_k es la media de la reflectancia en la longitud de onda k.

n_p es número de pixeles en la imagen.

Curtosis (cuarto momento)

$$\mu_{4k} = \frac{\sum (x_{ik} - \hat{x}_k)^4}{n_p} \quad (4)$$

x_{ik} son los valores de reflectancia normalizada en el pixel i para una longitud de onda k.

\hat{x}_k es la media de la reflectancia en la longitud de onda k.

n_p es número de pixeles en la imagen.

Quinto momento

$$\mu_{5k} = \frac{\sum (x_{ik} - \hat{x}_k)^5}{n_p} \quad (5)$$

x_{ik} son los valores de reflectancia normalizada en el pixel i para una longitud de onda k.

\hat{x}_k es la media de la reflectancia en la longitud de onda k.

n_p es número de pixeles en la imagen.

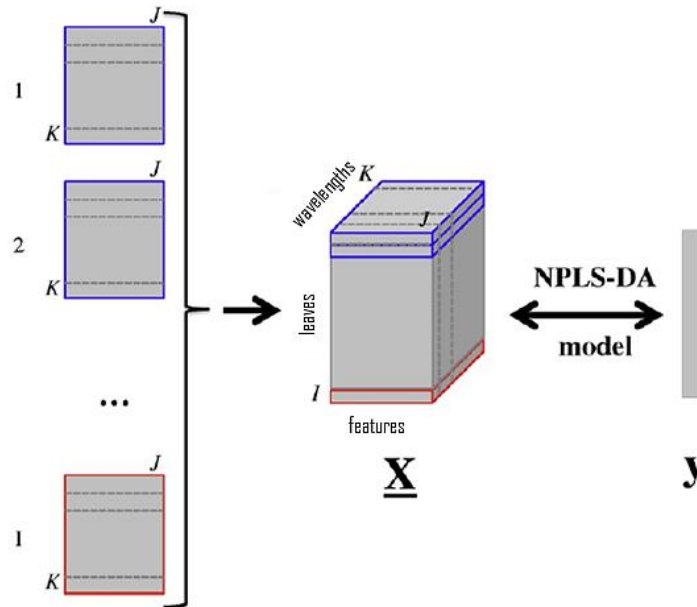


Figura 1 Estructura de tensor de tercer orden en NPLS-DA (Folch-Fortuny et al., 2016).

Una vez realizada la transformación de los cubos HS, un tensor de tercer orden con dimensiones 104 x 5 x 520 se formó con las 104 matrices generadas (figura 3.7). Este nuevo cubo de datos fue desplegado en el primer modo y se obtuvo la matriz final de dimensiones 104 filas x 2600 columnas (I x JK) (figura 3.8).

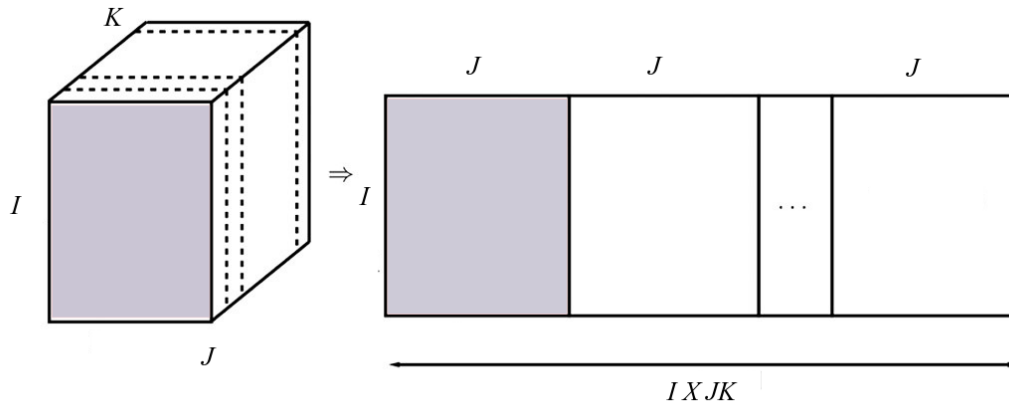


Figura 2 Tensor desplegado primer modo.

Durante la calibración del modelo se buscó el número mínimo de componentes para lograr la mejor predicción y luego, se realizó la prueba de validación externa utilizando la misma muestra utilizada en validación externa del modelo PLS-PLR.

El método NPLS-DA fue desarrollada en el lenguaje R.

Algoritmo NPLS

Utilizando Descomposición en valores singulares SVD, se obtiene las componentes principales a partir del producto vectorial, obteniendo los espacios latentes \mathbf{W}^k y \mathbf{W}^j que optimizan la varianza entre \mathbf{X} y \mathbf{y} . De la siguiente manera (Bro, 1996; Bro, 1998):

1.- Cálculo de \mathbf{Z}

$$\mathbf{Z} = \mathbf{X}^T \mathbf{y}$$

Donde,

\mathbf{X} es la matriz desplegada en el primer modo ($I \times JK$)

I es el número de muestras

J es el número de características calculadas

K es el número de bandas espectrales capturadas por la cámara

\mathbf{y} : Vector respuesta de I filas.

\mathbf{Z} : matriz de covarianza

2.- Cálculo de los vectores singulares izquierdos y derechos. SVD de matriz \mathbf{Z} . Se obtiene el primer vector singular izquierdo y el primer vector singular derecho ($\mathbf{W}^k, \mathbf{W}^j$).

$$\mathbf{Z} = \mathbf{U}\Sigma\mathbf{V}'$$

3.- Cálculo del vector \mathbf{t} de la matriz \mathbf{T} (Scores) modelo de mínimos cuadrados

$$\mathbf{t} = \mathbf{X}(\mathbf{W}^k \otimes \mathbf{W}^j)$$

4.- Calcula la \mathbf{b} de regresión. \mathbf{T} incluye todos los scores calculados hasta el momento.

$$\mathbf{b} = (\mathbf{T}'\mathbf{T})^{-1}\mathbf{T}'\mathbf{y}$$

5.- Cálculo de Residuos

$$\mathbf{y} = \mathbf{y}_0 - \mathbf{T}\mathbf{b}$$

$$\mathbf{X} = \mathbf{X} - \mathbf{T}\mathbf{W}^j(\mathbf{W}^k)^T$$

6.- Reemplaza y continua hasta el apropiado \mathbf{y}_0 .

Entrenamiento del modelo NPLS-DA

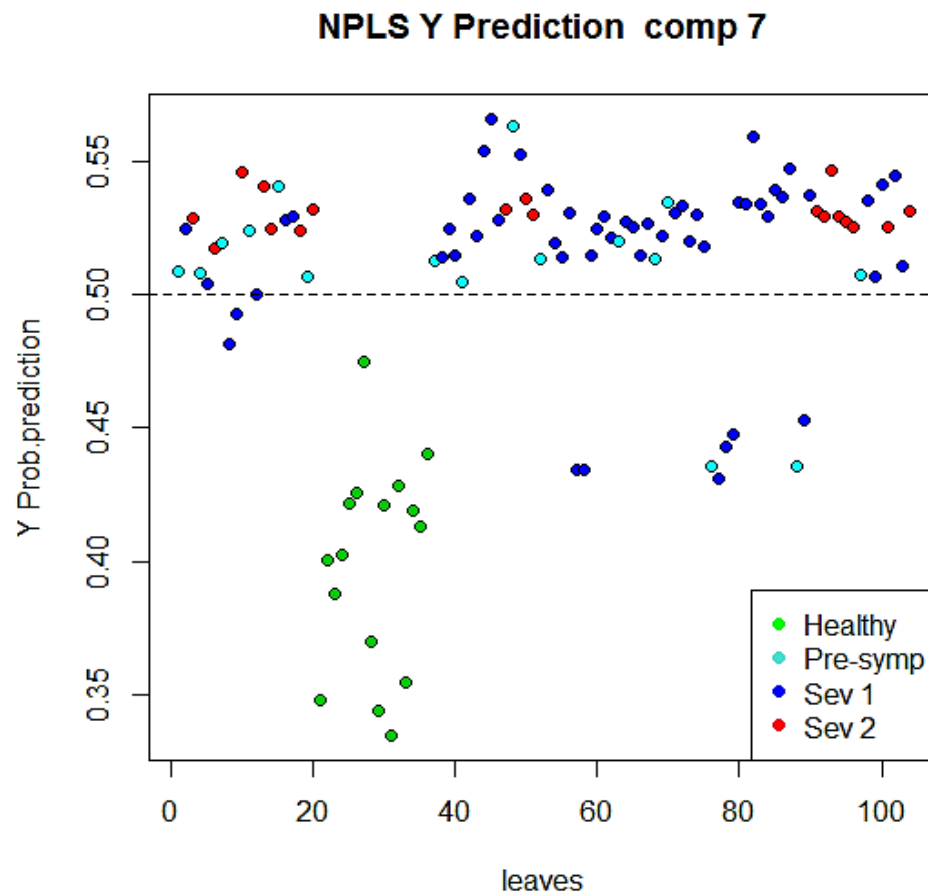


Tabla 1 Matriz de confusión del modelo NPLS-DA con datos de entrenamiento

| | Hojas Infectadas | Hojas no-infectadas | |
|---------------------------|---------------------|------------------------|----------------------|
| Resultado Test | TP | FP | Precisión |
| | 78 | 0 | 1 |
| | FN | TN | Valor Pred. negativo |
| | 10 | 16 | 0.62 |
| | Sensibilidad | Especificidad | Exactitud |
| | 0.89 | 1 | 0.90 |

Validación del modelo NPLS-DA

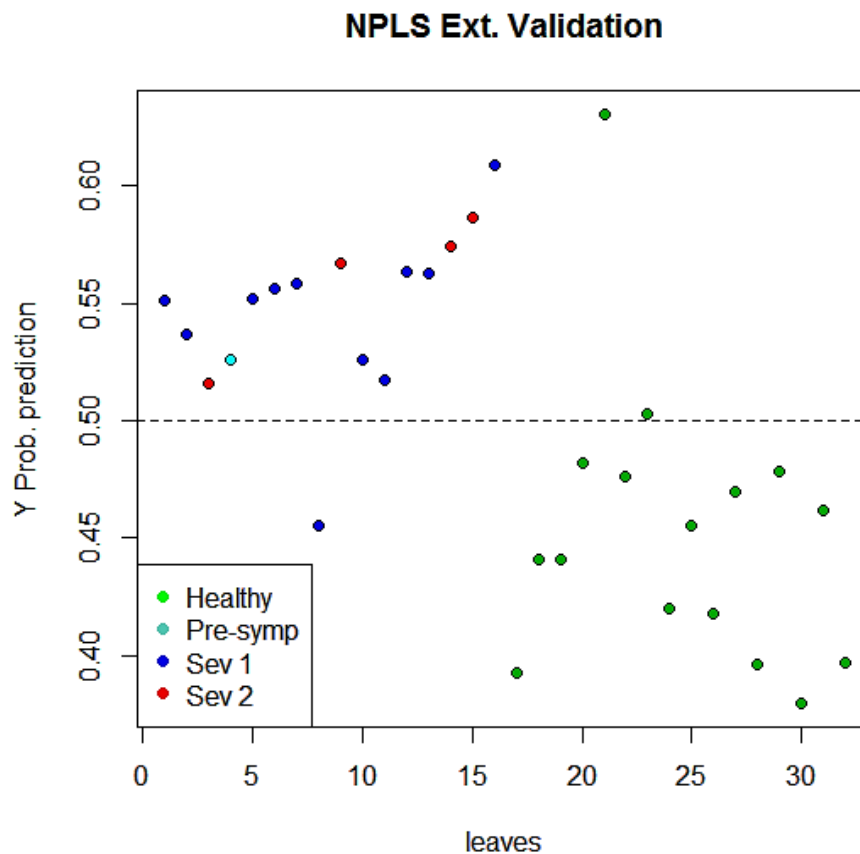


Figura 4 Predicción del modelo NPLS-DA en prueba de validación.

Tabla 2 Matriz de confusión para evaluación del modelo NPLS-DA

| | Hojas Infectadas | Hojas no-infectadas | |
|-------------------|---------------------|------------------------|----------------------|
| Resultado Test | TP | FP | Precisión |
| | 15 | 2 | 0.88 |
| | FN | TN | Valor Pred. negativo |
| | 1 | 14 | 0.93 |
| | Sensibilidad | Especificidad | Exactitud |
| | 0.94 | 0.88 | 0.91 |

Área bajo la curva ROC (AUC). El valor AUC se obtuvo mediante la función *auc()* del lenguaje R, el resultado fue 0.91.