Implementation in MATLAB of the Partial Least Squares algorithm for classification

Case study: fault detection and diagnosis on steel plates

Lorenzo Ferrari Lorenzo Leoni

Department of Engineering and Applied Sciences, University of Bergamo

May 3, 2023

Outline

Introduction to the PLS technique

Partial least squares (**PLS**), as known as **projection to latent structures**, is a dimensionality reduction technique for maximizing the **covariance** between the predictor (independent) matrix $X \in \mathbb{R}^{n \times m}$ and the predicted (dependent) matrix $Y \in \mathbb{R}^{n \times p}$ for each component of the reduced space $\mathbb{R}^{\alpha}$ with $\alpha \leq m$, where:

- n = number of observations;
- m = number of covariates (input variables);
- p = number of dependent variables (output variables);
- α = dimension of the reduced space in which X is projected.

Popular application of PLS

This technique is often used in **fault detection** and **isolation**. With PLS is possible to treat both regression and classification problems. The matrix X always contains the process variables (e.g. diameter and thickness of a gasket), while the matrix Y only (quantitative) quality variables (e.g. its mechanical seal) in the regression case, whereas in pattern classification the predicted variables are dummy variables (1 or 0) such as:

$$Y = \begin{bmatrix} \boldsymbol{y}_1 & \boldsymbol{y}_2 & \boldsymbol{y}_3 \end{bmatrix} = \begin{bmatrix} 1 & \dots & 1 & 0 & \dots & 0 & 0 & \dots & 0 \\ 0 & \dots & 0 & 1 & \dots & 1 & 0 & \dots & 0 \\ 0 & \dots & 0 & 0 & \dots & 0 & 1 & \dots & 1 \end{bmatrix}^\top \quad p = 3$$

where each column of Y corresponds to a *fault class*. The first n_j elements of column j are filled with a 1, which indicates that the first n_j rows of X are data from fault j. In classification case PLS is named **discriminant**.

NIPALS algorithm

The most popular algorithm used in PLS to compute the model parameters is known as **non-iterative partial least squares** (**NIPALS**). There are two versions of this technique:

- **PLS1**: each of the p predicted variables in modeled separately, resulting in one model for each class;
- **PLS2**: all predicted variables are modeled simultaneously.

The first algorithm is more accurate than the other, however it requires more computational time than PLS2 to find the α eigenvectors into which project the m covariates.

MATLAB code

The following MATLAB code implements the PLS2 algorithm:

```
E = X; % residual matrix for X
F = Y; % residual matrix for Y
[~, idx] = max(sum(Y.*Y));
% search of the j-th eigenvector
for j = 1:alpha
    u = F(:, idx);
    tOld = 0;
    for i = 1:maxIter
        w = (E'*u)/norm(E'*u); % support vector
        t = E*w; % j-th column of the score matrix for X
        q = (F'*t)/norm(F'*t); % j-th column of the...
            % loading matrix for Y
        u = F*q; % j-th column of the score matrix for Y
```

```
    if abs(tOld - t) < exitTol
        break;
    else
        tOld = t;
    end
end
p = (E'*t)/(t'*t); % j-th column of the...
    % loading matrix of X
% scaling
t = t*norm(p);
w = w*norm(p);
p = p/norm(p);
% calculation of b and the error matrices
b = (u'*t)/(t'*t); % j-th column of the...
    % coefficient regression matrix
E = E - t*p';
F = F - b*t*q';
```

```
    % calculation of W, P, T and B2
    W(:, j) = w;
    P(:, j) = p;
    T(:, j) = t;
    B2 = W*(P'*W)^-1*(T'*T)^-1*T'*Y;
end
Y_hat = X*B2; % computation of predictions
```

For each row of `Y_hat` the fault class is chosen by assigning 1 to the column whose value si greater than that of the others, 0 otherwise. Moreover, to increase the performances of PLS it is necessary **normalize** both X and Y before running the algorithm.

TO DO

TO DO