

## Research Article

# The L-Curve Criterion as a Model Selection Tool in PLS Regression

**Abdelmounaim Kerkri<sup>1</sup>, Jelloul Allal,<sup>1</sup> and Zoubir Zarrouk<sup>2</sup>**

<sup>1</sup>*Laboratory of Stochastic and Deterministic Modelling (LMSD), Department of Mathematics, Faculty of Sciences, University Mohamed the First, Oujda 60000, Morocco*

<sup>2</sup>*Laboratory of Management and Economy of Organizations, Faculty of Social Sciences, University Mohamed the First, Oujda 60000, Morocco*

Correspondence should be addressed to Abdelmounaim Kerkri; krkabdelmounaim@gmail.com

Received 10 June 2019; Revised 5 September 2019; Accepted 3 October 2019; Published 30 October 2019

Academic Editor: Luis A. Gil-Alana

Copyright © 2019 Abdelmounaim Kerkri et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Partial least squares (PLS) regression is an alternative to the ordinary least squares (OLS) regression, used in the presence of multicollinearity. As with any other modelling method, PLS regression requires a reliable model selection tool. Cross validation (CV) is the most commonly used tool with many advantages in both precision and accuracy, but it also has some drawbacks; therefore, we will use L-curve criterion as an alternative, given that it takes into consideration the shrinking nature of PLS. A theoretical justification for the use of L-curve criterion is presented as well as an application on both simulated and real data. The application shows how this criterion generally outperforms cross validation and generalized cross validation (GCV) in mean squared prediction error and computational efficiency.

## 1. Introduction

Partial least squares regression is an alternative to ordinary least squares regression in the presence of multicollinearity; it was first developed by Wold [1], as an algorithm that calculates the principal components of a dataset with missing values. PLS regression is used in many fields such as chemometrics, where collinearity is a common problem; this problem also arises in many other fields such as genomics and computational biology. When using PLS regression, one is confronted with the problem of stopping rule, (the number of components to retain in the final model); cross validation is used in the literature as an optimal way of choosing the final PLS model.

Despite its efficiency, cross validation can fail to detect the optimal model; in this paper, we will propose the L-curve criterion as an alternative model selection tool. The first use of L-curve plots in regularization methods goes back to Lawson and Miller [2, 3], and Hansen was the first who used these plots to calculate regularization parameters [4]. A cubic spline-based algorithm was developed later [5], for discrete L-curves, as well as the adaptive pruning

algorithm [6] used in our paper. We will be comparing cross validation with an L-curve algorithm used commonly in regularization methods such as conjugate gradient algorithm and Tikhonov optimization, investing in the equivalence of PLS regression with an implementation of conjugate gradient algorithm called CGLS algorithm (conjugate gradient least squares algorithm). Previous research has tackled similar comparisons such as Gao et al. [7], which presented a comparison of general cross validation and L-curve criterion to assess the performance of a new method for determining the optimal regularization parameter. The comparison will include generalized cross validation as well, even though it has not been commonly used in PLS regression.

The first section will be a short presentation of PLS regression, the second one will be about cross validation and its possible drawbacks, the third will present the generalized cross validation method, and finally, we will present the L-curve algorithm proving that it is suitable for PLS regression and finally an application on both real and simulated data to conclude. Our comparison shows that L-curve algorithm outperforms both cross validation and generalized

cross validation in real data and gives promising results in the simulated datasets.

## 2. PLS Regression and Model Selection

Consider the minimization problem:

$$\min \|X\beta - y\|_\beta, \quad (1)$$

with  $X \in \mathbb{R}^{n \times m}$  is a matrix presenting the values of the predictive variables,  $y \in \mathbb{R}^n$  is the dependent variable, and  $\beta$  is the coefficient of the regression.

Instead of regular predictors, PLS regression uses a set of latent variables called scores  $t_k = X_k \omega_k$  (with  $X_k$  being the deflated matrix  $X$  during  $k$ th iteration of PLS regression). The latent variables or PLS components are iteratively calculated, based on the next composition:

$$X = t_1 p_1' + \cdots + t_m p_m' = T^P + E, \quad (2)$$

with  $E$  being the error,  $p_k$  a set of vectors called loadings, and  $\omega_k$  being the weight vector of length  $k$ .

When retaining all PLS components, the PLS estimator becomes the ordinary least squares estimator, that is why it is necessary to know the optimal number of components to include in the final model. The main goal behind PLS regression is to be able to predict future observations, that is where cross validation comes in handy, because it tests the predictive ability of the model regardless of its fit.

## 3. Cross Validation

The main idea behind cross validation is that the power of the model depends on its ability to predict future values. In  $K$ -fold cross validation, observations are randomly split into  $K$  equal subsets,  $S_1, \dots, S_K$  for  $K$  times; we fit the model on  $S - S_i$  ( $i = 1, \dots, K$ ) and use the estimated parameters to predict the values of  $S_i$ , which provides an estimation for the prediction error.

In PLS regression,  $K$  is often equal to  $n$  (the number of observations), and this cross validation is called leave one out. Here are the steps of cross validation in PLS according to Tenenhaus [8]:

*Step 1.* In the  $k$ th PLS iteration, the estimation error and the predictive error are calculated using the expression:

$$\begin{aligned} \text{RSS}_k &= \sum_{i=1}^n (y_i - \hat{y}_{[k],i})^2, \\ \text{PRESS}_k &= \sum_{i=1}^n (y_i - \hat{y}_{[k],-i})^2, \end{aligned} \quad (3)$$

where  $\hat{y}_{[k],i}$  is the estimation of  $y_i$  by the model that includes all observations, and  $\hat{y}_{[k],-i}$  by the model that excludes the  $i$ th observation.

*Step 2.* Calculate:

$$Q_k^2 = 1 - \frac{\text{PRESS}_k}{\text{RESS}_{k-1}}. \quad (4)$$

A PLS component is retained, if  $Q_k^2 > 0.0975$ .

Cross validation is computationally expensive especially when working with big data, not to mention that it performs poorly when the number of predictors is greater than the number of observations; there is also the risk of overfitting when the number of observations is large, that is what motivated the choice of a new model selection tool that will be an alternative for cross validation, one that takes into consideration the regularizing nature of PLS regression.

## 4. Generalized Cross Validation

Generalized cross validation (GCV) [9] serves the purpose of choosing the optimal parameter of a regularization method. The method is based on predictive error estimates, mainly used in the numerical applications of regularization methods, such as Tikhonov regularization and conjugate gradient regularization. The method is described as successful by Hansen [10], as it is based on statistical considerations. The chosen parameter in GCV minimizes the function [9]:

$$V(\lambda) = \frac{(1/n)\|y - A(\lambda)y\|^2}{((1/n)\text{Trace}(I - A(\lambda)))^2}. \quad (5)$$

This formula was first conceived for Ridge regression, where

$$A(\lambda) = X(X'X + n\lambda I)^{-1}X'. \quad (6)$$

The difference between the existing versions of the GCV is due to the estimation of the trace of the matrix in the denominator. To our knowledge, GCV has not yet been used with PLS regression, but it has been used in regularized conjugate gradient method [11].

The above formula of the GCV in the case of PLS regression becomes [11]

$$V(k) = \frac{(1/n)\|y - X\beta_k^{\text{PLS}}\|^2}{((1/n)\text{Trace}(I - A(k)))^2}, \quad (7)$$

with  $A(k)$  the projection of the matrix  $(X'X)$  on the Krylov subspace of dimension  $k$ :

$$A(k) = W_k'(X'X)W_k, \quad (8)$$

where  $W_k$  being the matrix formed by the elements of a basis of the Krylov subspace of dimension  $k$ .

Various estimations are proposed for the trace of the matrix  $(I - A(k))$ ; we will use one proposed by [11], and similar to that used in [6]. This results in the GCV function:

$$V(k) = \frac{n\|y - X\beta_k^{\text{PLS}}\|^2}{(n - k)^2}. \quad (9)$$

Using the expression above, our stopping rule will be the parameter  $k$  that minimizes  $V(k)$ , throughout all the iteration of the PLS algorithm. The work by Hansen et al. [6] has demonstrated the robustness of the L-curve criterion when compared with the GCV in the case of truncated singular

value decomposition and regularizing conjugate gradient iterations.

It is also worth noting that GCV is not as popular as the old fashion cross validation method proposed by Tenenhaus [8], at least in the field of chemometrics and its applications. The main drawback of the GCV method is that it is computationally expensive, as it requires fitting all possible models, which becomes quite hassle when dealing with massive datasets that sometimes contain thousands of predictive variables.

## 5. The L-Curve Algorithm

To understand the problem of parameter choice, we will consider solving  $Ax = b$ , where  $A$  is a matrix and  $b$  a vector. When  $A$  is ill conditioned, the solution becomes unstable due to the difficulty of finding an inverse to  $A$ , that is where regularization interferes, offering an alternative set of solutions  $x_1, \dots, x_k$  through an iterative process, with  $k$  being the regularization parameter that differs according to the chosen method.

The L-curve criterion is a graphical parameter choice method used in regularization algorithms such as Tikhonov and truncated singular value decomposition (principal component regression); these regularization methods have the main goal of stabilizing the solution by including additional information.

There are two types of L-curve algorithms, the first one is for continuous regularization methods, and the second one is for the discrete ones; the L-curve is discrete when the parameter  $k$  varies in a discrete interval. In PLS regression,  $k$  represents the dimension of the space chosen for the optimization; it also represents the number of latent factors retained in the model. The goal behind this criterion is to find the right balance between the two quantities plotted in log scale, the norm of the residual error and the norm of the solution; this right balance is the corner of the L-curve, which is the point with maximum curvature; Figure 1 presents an example of the L-curve on a test problem.

The corner is calculated analytically by fitting a function to a discrete set of points, Hansen and O'Leary [5] used a cubic spline to fit the plot or geometrically by using the method of triangles [12]. We have chosen an algorithm called the adaptive pruning algorithm (Algorithm 1) [6], which is an algorithm developed to calculate the corner of a discrete L-curve.

## 6. The Adaptive Pruning Algorithm [6]

Consider a discrete L-curve, with a set of points  $P_1, \dots, P_k$ ; specifically let there be three points  $P_j$ ,  $P_k$ , and  $P_l$  with  $j < k < l$ . We define the oriented angle  $\theta_{(j,k,l)}$ . The principle is to calculate all possible angles and associate the corner to the angle closest to  $-(\pi/2)$ .

To avoid finding a local corner, this algorithm defines a sequence of pruned L-curves; from each, we choose two candidates for the corner. This produces a smaller list of candidate points to be corners. The next step is to choose the best point that meets the angle criteria, which we consider

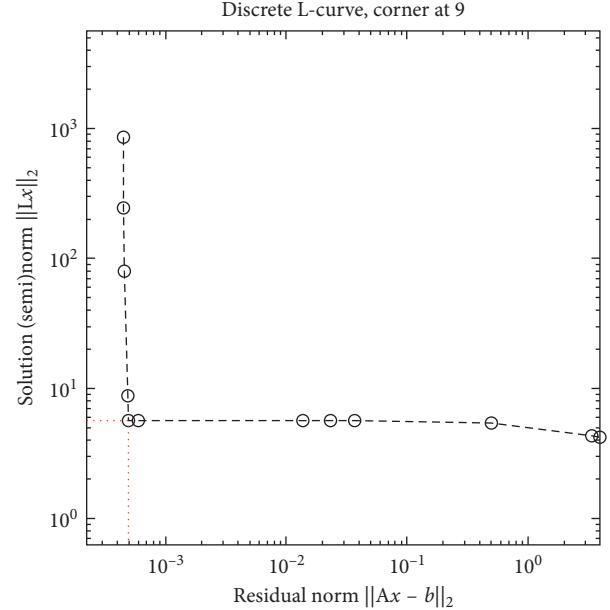


FIGURE 1: L-curve plot for a test problem.

our final corner for the L-curve; for more details, we recommend Hansen et al. [6].

This algorithm is implemented in the “regularization package” in MATLAB developed by Hansen [13]. The advantage of the adaptive pruning algorithm is that it takes into consideration the possibility of finding a local corner, and such an occurrence is common when data is of a high dimension.

## 7. The L-Curve Criterion and PLS Regression

The L-curve criterion uses the logic behind every regularization method, which is to find a compromise between the increasing size of the solution  $x_k$  in function of  $k$ , and the decreasing residual error in each parameter  $k$ :  $Ax - b$ , by plotting both quantities and finding the point of curvature.

The necessary assumptions for using the L-curve criterion is that the norm of residual error and the norm of the regularized solution are monotonic functions of the parameter  $k$  [10], and we will show next that PLS regression meets these assumptions.

**7.1. The Norm of the Solution.** In order to prove that the norm of the solution is a monotonic function of the parameter  $k$ , we will use the equivalence between PLS regression and the conjugate gradient algorithm applied to the normal equation.

**Proposition 1.** *Conjugate gradient algorithm performed on the normal equation:  $X'Xx = X'y$  with a starting vector  $x_0 = 0$  is equivalent to PLS regression performed on  $X$  and  $y$ .*

Proposition 1 means that PLS regression is a particular case of conjugate gradient applied on the normal equation, and we have written the proof in our previous paper [14]. We

- (1) Initialize  $\hat{P} = \min(5, P - 1)$
- (2) **Stage one:** while  $\hat{P} < 2(P - 1)$ 
  - (3)  $\hat{P} = \min(\hat{P}, P - 1)$
  - (4) Create a pruned L-curve consisting of the  $\hat{P}$  largest line segments.
  - (5) For each corner location routine
  - (6) Locate the corner  $P_k$  using the pruned L-curve.
  - (7) Add the corner to the list:  $L = L \cup \{P_k\}$
  - (8)  $\hat{P} = 2\hat{P}$
- (9) **Stage two:** if  $\#L = 1$  then  $k = k_1$ ; return.
- (10) Otherwise for  $i = 1, \dots, \#L - 1$ 
  - (11) Compute the slope  $\phi_i$  associated with point  $P_{k_i}$  in  $L$ .
  - (12) If  $\max\{\phi_i\} < (\pi/4)$  then  $k = \max\{k_i\}$ ; return.
  - (13) Otherwise let  $k = \min\{k_i : \phi_i > (\pi/4) \wedge \theta_{(k_{i-1}, k_i, k_{i+1})} < 0\}$ .

ALGORITHM 1: The algorithm as presented by Hansen et al. [6].

will use this link to justify the choice of L-curve and to prove that PLS meets its assumptions.

The next proposition is a direct result of the previous one, it is an application of theorem Heitens and Steifel, which will be helpful to prove that the norm of PLS is a monotonic function with respect to the parameter  $k$ .

**Proposition 2.** *The PLS estimators  $\beta_{PLS}^1, \dots, \beta_{PLS}^k$  are distinct, and they all minimize the expression:*

$$f(h) = (\beta_{OLS} - h)' X' X (\beta_{OLS} - h),$$

$$\beta_{PLS}^i = \sum_{j=0}^i \frac{f(\beta_{PLS}^j) - f(\beta_{PLS}^i)}{\|r_j\|_2^2} r_j, \quad (10)$$

with  $r_j = X' y - X' X \beta_{PLS}^j$ .

The residuals  $r_1, r_2, \dots, r_i$  are orthogonal according to Hestenes and Stiefel [15]; therefore, they form a basis for the Krylov subspace:

$$K_i(X' X, X' y) = \text{span}\{X' y, (X' X)X' y, \dots, (X' X)^{i-1} X' y\}, \quad (11)$$

which means that the norm of the PLS estimator in the  $i_{th}$  iteration is

$$\|\beta_{PLS}^i\|_2^2 = \sum_{j=0}^{i-1} \left( \frac{f(\beta_{PLS}^j) - f(\beta_{PLS}^i)}{\|r_j\|_2} \right)^2. \quad (12)$$

Since  $f$  is a decreasing function of  $k$ , we can easily demonstrate that the norm of the PLS estimator is an increasing function of parameter  $k$ ; therefore, it is monotonic.

## 7.2. The Norm of the Residual Error.

We have

$$\|(X' X)\beta_{PLS}^i - X' y\|_2 = \|(X' X)(\beta_{PLS}^i - \beta_{OLS})\|_2. \quad (13)$$

According to Hansen [10], the expression above for the residual error of conjugate gradient algorithm applied on the normal equation is a decreasing function of  $k$ , when the initial vector:  $x_0 = 0$ , which is equivalent to saying that the

TABLE 1: Chemical components of octane mixtures.

|                  |                        |
|------------------|------------------------|
| Straight run     | $0 \leq x_1 \leq 0.21$ |
| Reformate        | $0 \leq x_2 \leq 0.62$ |
| T. C. naphta     | $0 \leq x_3 \leq 0.12$ |
| C. C. naphta     | $0 \leq x_4 \leq 0.62$ |
| Polymer          | $0 \leq x_5 \leq 0.12$ |
| Alkylate         | $0 \leq x_6 \leq 0.74$ |
| Natural gasoline | $0 \leq x_7 \leq 0.08$ |

residual error of PLS regression is also a decreasing function of parameter  $k$ .

## 8. Application

The goal of this simulation is to compare the performance of three model selection tools in PLS regression, cross validation, generalized cross validation, and L-curve criterion used in regularization methods. The comparison will include both simulated and real datasets; the algorithm used in L-curve is the adaptive pruning algorithm, used in discrete regularization [6].

**8.1. Real Data.** We will use a dataset called Cornell (Table 1), presented in [8, 16]. These data measure the effect of 7 chemical components on octane motor rating, on a sample of 12 different mixtures.

Since we cannot calculate mean squared error of the regression coefficient, we will compare the values of  $R$  squared and mean squared error of the response variable in cross validation GCV and L-curve.

The comparison between CV, GCV, and L-curve on the real dataset Cornell (Table 2) shows that the last one gives a better model, with an improvement over the values of the mean squared error of  $y$ , and  $R$  squared, which means a better predictive power of the PLS model.

We note that L-curve selected the model that contains 5 components (Figure 2), GCV selected the model with four components, and cross validation selected the one with only the first three components.

The comparison of the elapsed time for each method shows the next result.

TABLE 2: The result of the comparison between cross validation (CV), generalized cross validation (GCV) and L-curve criterion on real data.

| R squared |      |         | MSEP of $y$ |      |         | The number of PLS component retained in the model |     |         |
|-----------|------|---------|-------------|------|---------|---|-----|---------|
| CV        | GCV  | L-curve | CV          | GCV  | L-curve | CV  | GCV | L-curve |
| 0.97      | 0.99 | 0.99    | 0.36        | 35.8 | 29.4    | 3   | 4   | 5       |

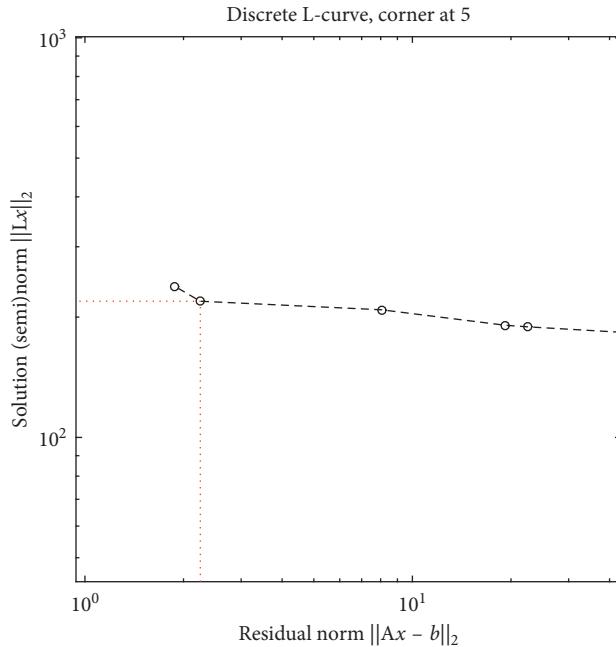


FIGURE 2: L-curve plot with the chosen corner in the dataset Cornell, by the adaptive pruning algorithm.

According to Table 3, GCV takes slightly more time to select the optimal model than the other two methods. The difference will increase as the dimension of the space variables increases, and this will be more palpable in the simulation.

**8.2. Simulated Data.** We compare cross validation, generalized cross validation, and L-curve algorithm on simulated datasets, and each has  $n = 100$  observations. We chose the number of variables for each dataset randomly, by selecting  $m$  in the interval [10, 90]. The goal is to explore the robustness of the methods in the space of variables.

The datasets are generated using a multivariate normal distribution, with zero mean and a covariance matrix that takes two values corresponding to medium and high collinearity.

$$C_{ij} = 0.8, 0.5. \quad (14)$$

The regression vector is generated from a uniform distribution in the interval [0, 3], and we consider three different signal-to-noise ratios for the standard deviation of the error.

$$\text{sntr} = \frac{\text{var}(X\beta)}{\text{var}(\sigma)} = 1, 5, 7. \quad (15)$$

We have 6 different datasets, generated 500 times each, and we estimate the PLS model for each one, using CV, GCV, and L-curve algorithm, to choose the number of components to keep in the final model. We compare the mean of 500 values of mean squared error (MSE), for both the regression coefficient and the estimated value of the dependent variable for each model selection method, and the results are summed up in Tables 4–6.

The MSE of the regression coefficient is calculated using the expression (Euclidian norm):

$$\text{MSE}(\beta_{\text{PLS}}^k) = \frac{1}{n} \|\beta_{\text{PLS}}^k - \beta\|^2, \quad (16)$$

$\beta$  being the real regression coefficient.

The first problem encountered in the result of cross validation is that in case of a signal-to-noise ratio equal to 1, the quantity of  $Q_k^2$  is negative in all cases throughout the 500 datasets. This means that it fails to select a model (in this case, we chose the first component for comparison). GCV seems to give the smallest MSE for the response variable, but it gives the highest one for the regression coefficient; cross validation yields a low regression coefficient MSE, but it inflates that of the response variable remarkably. In the meantime, L-curve gives reasonable results, giving a better MSE of the regression coefficient than GCV and giving almost a similar result for the MSE of the response variable.

TABLE 3: The result of the comparison between cross validation (CV), generalized cross validation (GCV), and L-curve criterion in PLS regression in terms of speed of execution in the case of real data.

|                           | CV   | GCV  | L-curve |
|---------------------------|------|------|---------|
| Elapsed time (in seconds) | 0.01 | 2.42 | 0.06    |

TABLE 4: The result of the comparison between cross validation (CV), generalized cross validation (GCV), and L-curve criterion in PLS regression in terms of mean squared error of both the estimator and the predicted  $y$ .

| Parameters $(m, C_{ij}, \text{stnr})$ | CV                 |            | GCV                |            | L-curve            |            |
|---------------------------------------|--------------------|------------|--------------------|------------|--------------------|------------|
|                                       | MSE of coefficient | MSE of $y$ | MSE of coefficient | MSE of $y$ | MSE of coefficient | MSE of $y$ |
| $(m, 0.5, 1)$                         | 108                | 217780     | 13414              | 63990      | 6786               | 80820      |
| $(m, 0.5, 5)$                         | 0.0445             | 433.20     | 25.31              | 117.58     | 11.00              | 149.5405   |
| $(m, 0.5, 7)$                         | 0.0414             | 112.4997   | 6.43               | 29.0744    | 2.90               | 38.8957    |
| $(m, 0.8, 1)$                         | 97                 | 559670     | 8679.3             | 151260     | 3596               | 199000     |
| $(m, 0.8, 5)$                         | 0.435              | 1000.9     | 135.98             | 269.7      | 64.43              | 343        |
| $(m, 0.8, 7)$                         | 0.0599             | 380.83     | 38.46              | 148.54     | 19.94              | 169.22     |

TABLE 5: The result of comparison between cross validation (CV), generalized cross validation (GCV), and L-curve criterion in PLS regression in terms of the number of components retained in the final model.

| $(m, C_{ij}, \text{stnr})$ | The number of PLS components retained in the final model |     |                   |
|----------------------------|--|-----|-------------------|
|                            | CV   | GCV | L-curve criterion |
| $(m, 0.5, 1)$              | 1*   | 6   | 4                 |
| $(m, 0.5, 5)$              | 1  | 6   | 4                 |
| $(m, 0.5, 7)$              | 2  | 6   | 4                 |
| $(m, 0.8, 1)$              | 1*   | 6   | 3                 |
| $(m, 0.8, 5)$              | 1  | 6   | 4                 |
| $(m, 0.8, 7)$              | 1  | 10  | 5                 |

\*A negative value of  $Q_k^2$ .

TABLE 6: The result of the comparison between cross validation (CV), generalized cross validation (GCV), and L-curve criterion in PLS regression in terms of speed of execution in the case of simulated datasets.

|                           | CV    | GCV   | L-curve |
|---------------------------|-------|-------|---------|
| Elapsed time (in seconds) | 11.75 | 75.04 | 0.047   |

For the comparison of the speed of the three methods in simulated data, we chose ten datasets having 100 observations each, with variables varying from 90 to 99. Table 6 presents the mean elapsed time for the ten datasets. GCV takes roughly 75.04 seconds to select the optimal model, making it the slowest one, followed by CV with 11.75 seconds, with L-curve consistently giving the lowest execution time with the value of 0.047.

## 9. Conclusion

In this paper we presented a model selection tool that is commonly used in regularization methods, and we used it in PLS regression, investing in the link that exists between partial least squares regression and the conjugate gradient algorithm. To assess the contribution of this model selection tool in PLS, we compared it with two versions of cross validation, one is basically the most popular method for model choice in PLS, due to its simplicity, the other is popular in the field of numerical analysis, and less common in the field of statistical modelling. L-curve seemed suitable

because it takes into account the balance between the size of the shrunk solution and the residual error. Our simulation study showed that L-curve outperforms the two methods in real data, and is much less computationally expensive in the case of artificial data, with results that are almost similar to the GCV method. Weighing in the main disadvantage of GCV, which is a high computational cost, L-curve seems to be the better choice when working with massive datasets, which is common in many fields of research that require the use of PLS regression.

## Data Availability

Previously reported data were used as real data to support this study and are available at [8, 16], which was cited in the text.

## Disclosure

An earlier version of this work has been presented as an oral communication in the International SM2A Conference in Meknes, Morocco, 2017.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## References

- [1] H. Wold, "Soft modelling by latent variables: the non-linear iterative partial least squares (NIPALS) approach," *Journal of Applied Probability*, vol. 12, no. S1, pp. 117–142, 1975.
- [2] C. L. Lawson and R. J. Hanson, *Solving Least Squares Problems*, Vol. 15, SIAM, Philadelphia, PA, USA, 1995.
- [3] K. Miller, "Least squares methods for ill-posed problems with a prescribed bound," *SIAM Journal on Mathematical Analysis*, vol. 1, no. 1, pp. 52–74, 1970.
- [4] P. C. Hansen, "Analysis of discrete ill-posed problems by means of the L-curve," *SIAM Review*, vol. 34, no. 4, pp. 561–580, 1992.
- [5] P. C. Hansen and D. P. O'Leary, "The use of the L-curve in the regularization of discrete ill-posed problems," *SIAM Journal on Scientific Computing*, vol. 14, no. 6, pp. 1487–1503, 1993.
- [6] P. C. Hansen, T. K. Jensen, and G. Rodriguez, "An adaptive pruning algorithm for the discrete L-curve criterion," *Journal of Computational and Applied Mathematics*, vol. 198, no. 2, pp. 483–492, 2007.
- [7] W. Gao, K. Yu, and Y. Wu, "A new method for optimal regularization parameter determination in the inverse problem of load identification," *Shock and Vibration*, vol. 2016, Article ID 732896, 16 pages, 2016.
- [8] M. Tenenhaus, *La régression PLS: théorie et pratique*, Editions Technip, 1998.
- [9] G. H. Golub, M. Heath, and G. Wahba, "Generalized cross-validation as a method for choosing a good ridge parameter," *Technometrics*, vol. 21, no. 2, pp. 215–223, 1979.
- [10] P. C. Hansen, *Rank-Deficient and Discrete Ill-Posed Problems: Numerical Aspects of Linear Inversion*, Vol. 4, SIAM, Philadelphia, PA, USA, 2005.
- [11] P. Favati, G. Lotti, O. Menchi, and F. Romani, "Generalized cross-validation applied to conjugate gradient for discrete ill-posed problems," *Applied Mathematics and Computation*, vol. 243, pp. 258–268, 2014.
- [12] J. L. Castellanos, S. Gómez, and V. Guerra, "The triangle method for finding the corner of the L-curve," *Applied Numerical Mathematics*, vol. 43, no. 4, pp. 359–373, 2002.
- [13] P. C. Hansen, "Regularization tools: a Matlab package for analysis and solution of discrete ill-posed problems," *Numerical Algorithms*, vol. 6, no. 1, pp. 1–35, 1994.
- [14] A. Kerkri, J. Allal, and Z. Zarrouk, "Highlighting the regularization properties of PLS regression using the conjugate gradient algorithm," *International Journal of Applied Mathematics and Statistics*, vol. 56, no. 5, pp. 63–74, 2017.
- [15] M. R. Hestenes and E. Stiefel, *Methods of Conjugate Gradients for Solving Linear Systems*, NBS, vol. 49, no. 1, Washington, DC, USA, 1952.
- [16] N. Kettaneh-Wold, "Analysis of mixture data with partial least squares," *Chemometrics and Intelligent Laboratory Systems*, vol. 14, no. 1–3, pp. 57–69, 1992.

