

Simplifying Complexity: Linearization Method for Partial Least Squares Regression

Herlin Simanullang^{1)*}, Sutarman²⁾, Open Darnius³⁾

¹⁾Postgraduated Mathematics, Universitas Sumatera Utara, Indonesia

^{2,3)}Department of Mathematics, Universitas Sumatera Utara, Medan, Indonesia

¹⁾herlinsimanullang@gmail.com , ²⁾sutarman@usu.ac.id , ³⁾opendarnius@gmail.com

Submitted : Jul 23, 2023 | **Accepted** : Jul 23, 2023 | **Published** : Jul 25, 2023

Abstract: This research investigates Romera's local linearization approach as a variance prediction method in partial least squares (PLS) regression. By addressing limitations in the original PLS regression formula, the local linearization approach aims to improve accuracy and stability in variance predictions. Extensive simulations are conducted to assess the method's performance, demonstrating its superiority over traditional algebraic methods and showcasing its computational advantages, particularly with a large number of predictors. Additionally, the study introduces a novel computational technique utilizing bootstrap parameters, enhancing computational stability and robustness. Overall, the research provides valuable insights into the local linearization approach's effectiveness, guiding researchers and practitioners in selecting more reliable and efficient regression modeling techniques.

Keywords: Partial least squares regression, Linearization Method, Orthogonal Score Algorithm

INTRODUCTION

Linearization is the linear approximation of a function at one point in mathematics and its applications. Linearization is a method used in dynamical systems to estimate the local stability of the equilibrium point in a system of nonlinear differential equations (Guckenheimer & Holmes, 2013; Mahfouf, 1999).

Regression is a statistical method for describing the relationship between one or more independent variables (X) and one or more response variables (Y). Which is expressed as $Y = \alpha + \beta x$. The least squares method is used to determine the regression interpretation based on the parameters α and β . The least squares method is widely regarded as the best estimating method in regression analysis, but it is extremely sensitive to data deviations from assumptions. If the assumption is violated, i.e., there is a high correlation between the independent variables (multicollinearity), the resulting estimator is still unbiased and consistent, but it is inefficient, so the variance of the regression coefficients is not minimized (overestimated). Meanwhile, if the number of independent variables is greater than the number of observations, the independent variable matrix structure becomes singular (Zhang & Garcia-Munoz, 2009).

The dependent variable (Y) was predicted using the independent variable (X) using least squares regression. The least squares method is used to estimate the parameters of a simple linear regression model. For example, if you have a regression data set Y_i, X_i with $i = 1, 2, 3, \dots, n$, the relationship Y_i and X_i in the regression equation looks like this:

$$y_i = \alpha + \beta x_i + \varepsilon_i \quad (1)$$

while the regression line equation looks like this:

$$\hat{y}_i = \alpha + \beta x_i \quad (2)$$

The error equation is now:

$$\varepsilon_i = y_i - \hat{y}_i \quad (3)$$

The least squares method is used to reduce the squared error, so:

$$SSE = \sum_{i=1}^n \varepsilon_i^2 = \sum_{i=1}^n (y_i - \hat{y}_i)^2 = \sum_{i=1}^n (y_i - \alpha - \beta x_i)^2 \quad (4)$$

This equation's derivative with respect to α and β is as follows:

1. Derivatives in relation to α :

$$\frac{d}{d\alpha} SSE = -2 \sum_{i=1}^n (y_i - \alpha - \beta x_i) = 0 \quad (5)$$

$$n\alpha + \beta \sum_{i=1}^n x_i = \sum_{i=1}^n y_i \quad (6)$$

2. Derivatives derived from β :

$$\frac{d}{d\beta} SSE = -2 \sum_{i=1}^n (y_i - \alpha - \beta x_i)x_i = 0 \quad (7)$$

$$\alpha \sum_{i=1}^n x_i + \beta \sum_{i=1}^n x_i^2 = \sum_{i=1}^n x_i y_i \quad (8)$$

We get and by substituting (5) – (8) as follows:

$$\alpha = \frac{(\sum Y_i)(\sum X_i^2) - (\sum X_i)(\sum X_i Y_i)}{n \sum X_i^2 - \sum Y_i^2} \quad (9)$$

$$\beta = \frac{n \sum X_i Y_i - (\sum X_i)(\sum Y_i)}{\sum X_i - (\sum X_i)^2} \quad (10)$$

Principal component analysis and multiple regression are combined in partial least squares regression (PLSR). The objective is to forecast a set of response variables (Y) from a set of predictor variables (X). This prediction is obtained by extracting a number of components from the predictor variables known as latent variables.

Least squares regression can make predictions if a Y vector and an X matrix have full power. If the number of predictors exceeds the number of observations and X is incomplete or singular, the least squares regression approach is no longer appropriate because X has a multicollinearity problem.

PLSR will retrieve the X components that are also relevant to Y . This is accomplished by simultaneously decomposing X and Y with the constraint that these components explain as much of the covariance between X and Y as possible. This decomposition process is followed by a regression stage in which the X decomposition results are used to predict Y .

If X is of size $N \times K$ (N is the number of observations and K is the number of predictor variables), it consists of vector X_k , $k = \{1, 2, \dots, K\}$, and Y is of size $N \times M$ (M is the number of response variables). The PLSR method generates several new components that will model X and Y . These new components are known as X scores, and they are recorded as t_a , $a = \{1, 2, \dots, A\}$. The X score is a linear combination of the original variables X_k with weights, as recorded by the vector w_{ka} , $a = \{1, 2, \dots, A\}$. The procedure can be stated as follows:

$$\begin{cases} t_{ia} = \sum x_{ik} w_{ka}, & i = 1, 2, \dots, N \\ T = XW \end{cases} \quad (11)$$

(Romera, 2010) made predictions using the formulation of partial least squares regression and approached it linearly. In this case, Romera suggested an alternative computational approach using bootstrap parameters.

The purpose of this study is to compare and evaluate the method used with the following suggestions and to assess the formulas described by (Romera, 2010). Hopefully, this research can be beneficial in addressing regression problems, particularly those related to partial least squares regression, as well as decision-making problems in uncertain scenarios.

METHOD

Regression Formulas

Given a calibration model and predictions from random data, the relationships can be described as follows:

Calibration Model:

$$\dot{y}_c = \beta_0 + \dot{X}_c \beta + \epsilon,$$

Prediction Model:

$$\dot{y}_p = \beta_0 + \dot{X}_p \beta + \epsilon,$$

Where:

\dot{y}_c is the un-centered calibration response variable.

\dot{y}_p is the un-centered prediction response variable.

\dot{X}_c ($n \times k$) is the matrix of centered calibration data variables.

\dot{X}_p ($n_p \times k$) is the matrix of centered prediction data variables.

β is the vector of regression coefficients (β_0 and β ($k + 1$) in this case).

ϵ is the error term with a normal distribution having a mean of 0 and variance σ_ϵ^2 .

The dot above variables (e.g., \dot{y}_c) indicates that they are un-centered variables, and they correspond to the centered variables for y_c .

Orthogonal Score Algorithm

The orthogonal scoring algorithm, developed by (Martens & Naes, 1992), is highly regarded for its simplicity, stability, and widespread utilization in various fields. When the factor number a is chosen, each step of the algorithm yields the outcome for the respective factor i , $i = 1, 2, \dots, a$. This step-wise approach facilitates the determination of results corresponding to each factor, offering valuable insights and analyses for a range of applications.

Calibration

The algorithm initiates its process from the midpoint of the matrix data calibration,

$$\begin{aligned} x_{c_1} &= x_c \\ w_i &= X'_{c_i} y_c \\ t_i &= X'_{c_i} w_i \\ q_i &= \frac{y'_{c_i} t_i}{t'_{c_i} t_i} \\ X_{c_{i+1}} &= X_{c_i} - t_i p'_i \end{aligned}$$

During the i -th step of the algorithm, the weight vector, denoted as $w_i (k \times l)$, is determined based on the covariance between the column vector X_{c_i} and y_c . The scores matrix of size $n \times a$, represented by $T = (t_1, t_2, \dots, t_a)$, and the x -loading matrix $x \times a$ ($p = (p_1, p_2, \dots, p_a)$) are calculated. Additionally, the y -loading vector q is defined as a column vector $a \times 1$.

In the first step, if the size of the weight vector w_i is only one element in length, the algorithm remains stable. This characteristic facilitates the comparison of scores and does not alter the estimated regression coefficients, even though its normalization does not undergo any changes. Notably, (Helland, 1988) demonstrated that the regression coefficient for partial least squares can be expressed as follows:

$$\hat{\beta} = W(P'W)^{-1}q$$

Furthermore, the score can be expressed through the following equation:

$$T = X_c W (P'W)^{-1}$$

Predictor

To estimate the predicted response variable, \widehat{y}_p , it can be generated from the x_p $1 \times k$ score. Unlike calibration, where t_i is a column of T , the predicted score, $t_p = (t_{p_1}, t_{p_2}, \dots, t_{p_a})$, is represented as a row vector, and t_{p_i} is calculated using the following steps:

1. Calculate t_{p_i} as:

$$t_{p_i} = x_{p_i} w_i$$

2. Update the next x_p score as follows:

$$x_{p_{i+1}} = x_{p_i} - t_{p_i} p'_i$$

where $x_{p_i} = \hat{x}_p - \bar{x}$, and $t_p = x_p W (P'W)^{-1}$.

The prediction for the response variable is then given by:

$$\widehat{y}_p = \bar{y} + t_p q$$

These calculations enable the estimation of the response variable \widehat{y}_p using the x_p scores and the corresponding model parameters. It is worth noting that the prediction process is distinct from the calibration procedure and involves updating the x_p scores iteratively to obtain the final prediction for the response variable.

Random Data Sample Model

Let \hat{c} be a vector $(k + 1) \times 1$ representing the interdependent and predictive variables of one case in the calibration or prediction set. It can be expressed as $\hat{c} = (\hat{y}, \hat{x})'$, where \hat{y} is the response variable and \hat{x} is a $k \times 1$ vector representing the predictor variables. The covariance between \hat{y} and \hat{x} is denoted by $\gamma = (\gamma_1, \gamma_2, \dots, \gamma_k)'$, and the variance-covariance matrix of \hat{x} is denoted by Σ with elements σ_{ij} for $1 \leq i, j \leq k$.

These parameters can be combined into an $ak(k + 3)/2 \times 1$ vector $\Phi = (\gamma' \text{vecut}(\Sigma))'$, where vecut is an operator that transforms a symmetric matrix into a column vector by stacking its upper triangular elements, including the diagonal, in column-major order.

Let's define the $k \times 1$ vector $s_{xy} = X_c'X_c$ as the sum of squares for the predictor variables in the calibration set. Additionally, let $b = (s'_{xy} \text{vecut}(S_{xx}))'$ be a vector, where s'_{xy} represents the observed value of the cross-product sum of squares and S_{xx} is the observed value of the sum of squares and cross-products matrix for the calibration set. The random variable b is an unbiased estimate of $(n - 1)\phi$, where n is the number of cases in the calibration set. This means that, on average, the vector b provides an accurate estimate of the true parameter vector ϕ , considering the calibration set's sample size (n) minus one.

This framework allows for the estimation of the parameters in the calibration or prediction set, enabling the analysis and modelling of the data in a statistically rigorous manner.

Romera's approach

(Romera, 2010) conducted a study to assess the accuracy of the regression coefficient estimate, $\hat{\beta}$ with respect to the vector b , and this estimation was based on the y -loading, q . The y -loading estimate was developed by observing the b_0 value of b using a first-order Taylor expansion. This can be represented as follows:

$$q_b \approx q_{b_0} + J(b - b_0)$$

Where q_b is the estimated y -loading based on the vector b , q_{b_0} is the y -loading estimate at b_0 , and J is the Jacobian matrix of size $a \times k(k + 3)/2$, representing the first derivative of q with respect to b , evaluated at b_0 . J can be mathematically expressed as $J(a \times \frac{k(k+3)}{2})$.

The estimated variance of the y -loadings, $\text{Var}(q)$, can be approximated as:

$$\text{ar}(q) \approx J\text{Var}(b)J'$$

Here, $\text{Var}(b)$ represents the variance of the vector b .

(Romera, 2010) then uses the relation $\hat{\beta} = Wq$, which provides the estimated regression coefficient $\hat{\beta}$ in terms of the y -loading vector q . To obtain an estimate of the variance of $x_p\hat{\beta}$, $\text{Var}(\hat{\beta})$, the following expression is used:

$$\text{Var}(x_p\hat{\beta}) \approx x_pWJ\text{Var}(b)J'W'x'_p$$

However, there are two main issues with this approach. Firstly, the equation (3) shows that $\hat{\beta} = W(P'W)^{-1}$ for the orthogonal scoring algorithm, and not $\hat{\beta} = Wq$, which is the outcome of the PLS1 orthogonal loading algorithm. Secondly, the weight matrix W depends on the vector b , which means that W cannot be considered correct when computing $\text{Var}(\hat{\beta})$.

These problems need to be addressed to ensure the accuracy and validity of the regression coefficient estimates and their associated variances in Romera's analysis. Further refinement and consideration of the algorithm and its dependencies are necessary for accurate results.

Estimation of $\text{Var}(\hat{\beta})$ with Bootstrap Parameters

In this alternative proof, the goal is to estimate the variance of the regression coefficient, $\text{Var}(\hat{\beta})$, using bootstrap parameters. The process involves generating bootstrap samples and deriving b_m values

from the Wishart Distribution. The estimated regression coefficient, $\hat{\beta}_m^\beta$, is then computed based on b_m , which may yield improved results compared to using X_c and y_c (Haff, 1979; Wentzell et al., 2017).

The formula representing the variance of the regression coefficient with the bootstrap algorithm is given by:

$$Var(\hat{\beta}^\beta) = \frac{n}{n+1} \frac{1}{M-1} \sum_{m=1}^M (\hat{\beta}_m^\beta - \bar{\beta})(\hat{\beta}_m^\beta - \bar{\beta})'$$

where $\hat{\beta}_m^\beta$ is the estimated regression coefficient from the m -th bootstrap sample, and $\bar{\beta}$ is the mean of these estimates computed as:

$$\bar{\beta} = \frac{1}{M} \sum_{m=1}^M \hat{\beta}_m^\beta$$

The factor $n/(n+1)$ plays a role in approaching the bootstrap parameter values.

The approximate variance of $x_p \hat{\beta}$ is then given as:

$$Var(x_p \hat{\beta}) \approx x_b Var(\hat{\beta}^\beta) x_p' = V_B$$

Where x_b is a vector representing the predictor variables for the bootstrap sample and V_B is the estimated variance of $x_p \hat{\beta}$ using the bootstrap algorithm.

This approach leverages bootstrap samples to obtain more robust estimates of the variance of the regression coefficient, thereby enhancing the accuracy and reliability of the statistical analysis. The use of bootstrapping techniques can provide valuable insights when dealing with complex datasets and addressing potential issues with the traditional algebraic methods.

RESULT

In this extensive simulation study, the goal is to investigate the linearization (Lin) and bootstrap (Linb) versions under different conditions. The simulations involve creating calibration sets and prediction sets with size $n = 200$. The explanatory variables are independent and normally distributed with mean 0 and variance $(\sigma_1^2, \sigma_2^2, \dots, \sigma_k^2)$ in both sets.

For each $N \times n_p$ predictions in the simulation, the squared prediction error is calculated, and the variances V_L and V_B are estimated using equations 4 and 5, respectively. The variance formula does not account for the contribution of \bar{x} and \bar{y} over repetitions of the calibration set. Thus, the contribution of \bar{y} , $\frac{\sigma_\epsilon^2}{n}$, is added to each variance estimate, resulting in Lin's variance formula becoming $\frac{\sigma_\epsilon^2}{n} + V_L$, and Linb's variance formula becoming $\frac{\sigma_\epsilon^2}{n} + V_B$.

In practice, only one value of σ_ϵ^2 is needed for estimation, and a fixed value can be used to focus on comparing the performance of V_L and V_B . The contribution of \bar{x} is accounted for as k/n^2 , and it is assumed to be negligible compared to other factors.

To check the performance of Lin and Linb, squared error plots are generated, and 2 times the variance is estimated for both V_L and V_B . The average of 20 bins defined by the abscissa variable x is taken. These bins are set using percentage points from the size of the chi-square random variable with the size and degree chosen freely to ensure accuracy in estimating both V_L and V_B . The number of samples is the same for each accuracy test per bin.

This simulation study aims to explore the properties of the linearization and bootstrap versions under different scenarios, including the correlation of predictors and their extrapolation effects. By comparing the performance of V_L and V_B through squared error plots and variance estimation, the study provides valuable insights into the suitability and accuracy of these variance formulas for various situations.

This simulation starts from two simulations with $k = 2$ and $a = 1$ so that the linearization is stable.

1. Simulation: $k = 2, a = 1, \sigma_1^2 = 25, \sigma_2^2 = 1, \beta_0 = \beta_1 = 1, \beta_2 = 0, \sigma_\epsilon^2 = 0,25, N = 10.000$

The first variable is a predictive variable with a non-zero regression coefficient. The first has a much higher variance than the second, which has a zero coefficient. The PLSR performed admirably, as expected, and both Lin and Linb performed admirably (Fig. 1). The plot against V_B

appears to be equally promising. Fig. 2 depicts how the estimated regression coefficient varies in relation to b . In the linear model, $\hat{\beta}_1$ is always close to 1, whereas $\hat{\beta}_2$ is dependent on two elements of b .

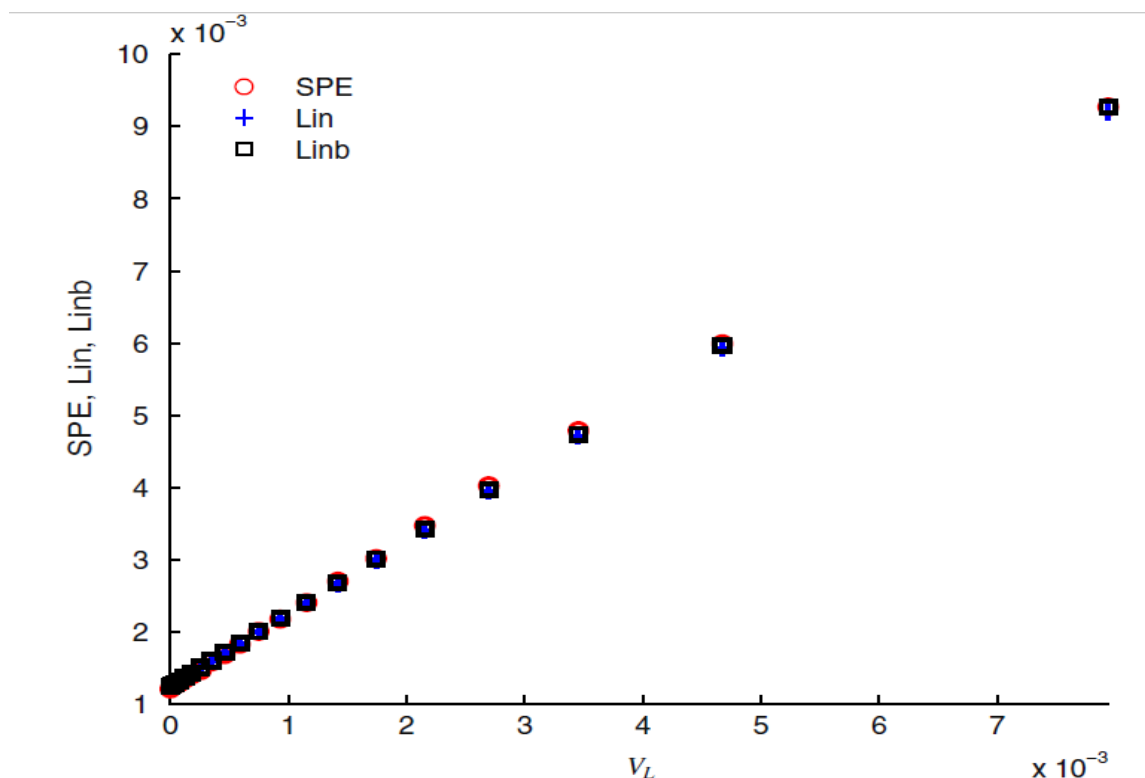
- Simulation: $k = 2, a = 1, \sigma_1^2 = 25, \sigma_2^2 = 1, \beta_1 = 0, \beta_0 = \beta_2 = 1, \sigma_\epsilon^2 = 0,25, N = 10.000$

In this more challenging scenario for PLSR, the first predictor has a larger variance but no regression contribution, while the second predictor, with a smaller variance, is related enough to the response variable to gain weight in the PLSR factor. The sign changes in $\hat{\beta}$, accompanied by the sign of the correlation, cause the breakdown of the linear approximation. As a result, both Lin and Linb methods fail, but in different ways, as illustrated in Figure 3.

Figures 1 and 2 provide insights into the reasons why Lin fails. Fig. 3 shows the distribution of $\hat{\beta}_1$ as a variation of b , which is bimodal, with modes switching as signs of changes in b_1 and b_4 . The local linearization method fails for a set of calibrations. The blue dotted line represents $\hat{\beta}$ calculated by changing b_1 and recalculating the PLSR algorithm. It can be observed that $\hat{\beta}_1$ and $\hat{\beta}_2$ vary with small changes in b_1 , making a linear approximation inadequate.

Linb's failure is less severe but still underestimates the SPE due to two reasons: bootstrapping undermines the true variance of $\hat{\beta}$, and the contribution from the bias in PLSR $\hat{\beta}$ cannot be ignored in this case.

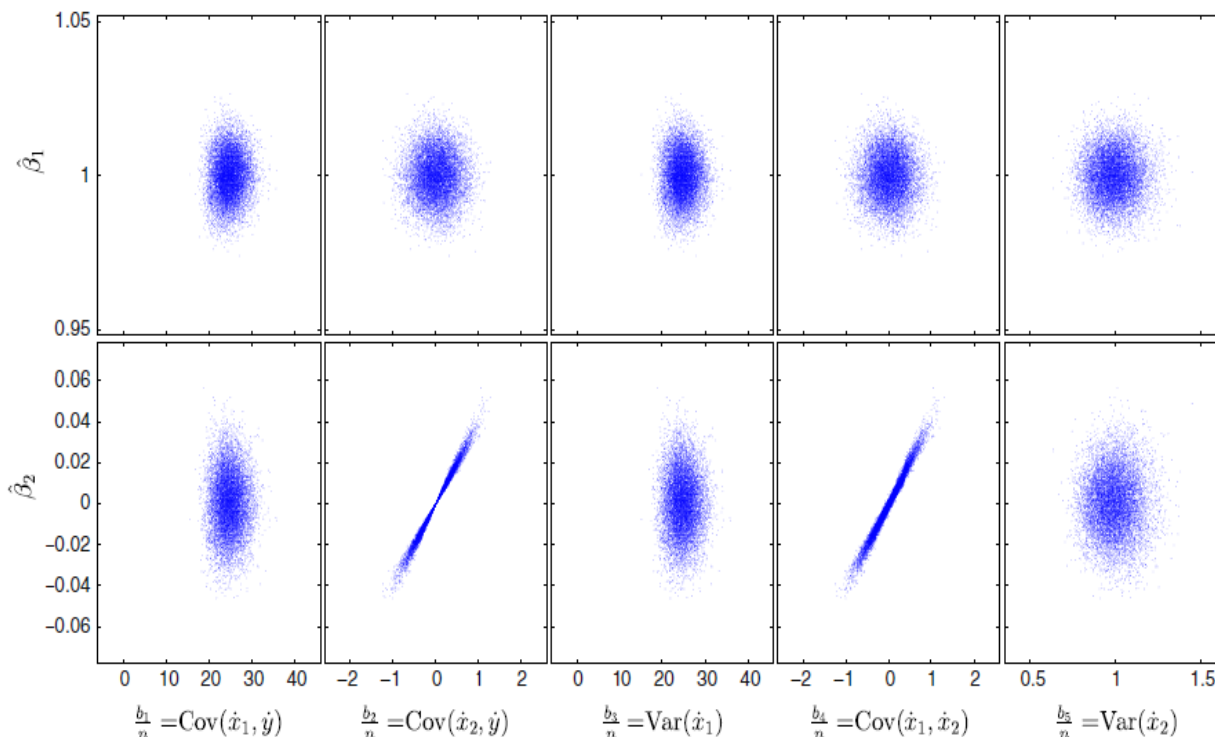
Fig. 1 LPSR estimates the new variance and predicted error versus V_L . $k = 2, a = 1, \sigma_1^2 = 25, \sigma_2^2 = 1, \beta_0 = \beta_1 = 1, \beta_2 = 0, \sigma_\epsilon^2 = 0,25$. SPE: Squared Prediction Error $(\hat{y}_p - \hat{y}_p)^2$. Lin: $V_L + \sigma_\epsilon^2/n$. Linb: $V_B + \sigma_\epsilon^2/n$.



The underestimation of the variance of predictions can be explained by considering the figures, particularly the top left of fig. 3. In the generalized iterative training set of combined response distributions and predictor variables where b_1 is zero-centered, $\hat{\beta}_1$ exhibits a bimodal distribution

with equal weight for each mode. The bootstrap estimation procedure will centre on the observation b_1 , which generally will not be zero. While in bootstrap, $\hat{\beta}_1$ still tends to have a bimodal distribution but with unequal weights in the two modes, leading to a smaller variance compared to $\hat{\beta}_1$ in the repeated training set. This accounts for 20% of the difference between Linb and SPE, while the remaining discrepancy is attributed to substantial bias in the PLSR $\hat{\beta}$.

Fig. 2 PLSR $\hat{\beta}$ counter $\frac{b}{n}$ when $k = 2, a = 1, \sigma_1^2 = 25, \sigma_2^2 = 1, \beta_0 = \beta_1 = 1, \beta_2 = 0, \sigma_\epsilon^2 = 0,25$.

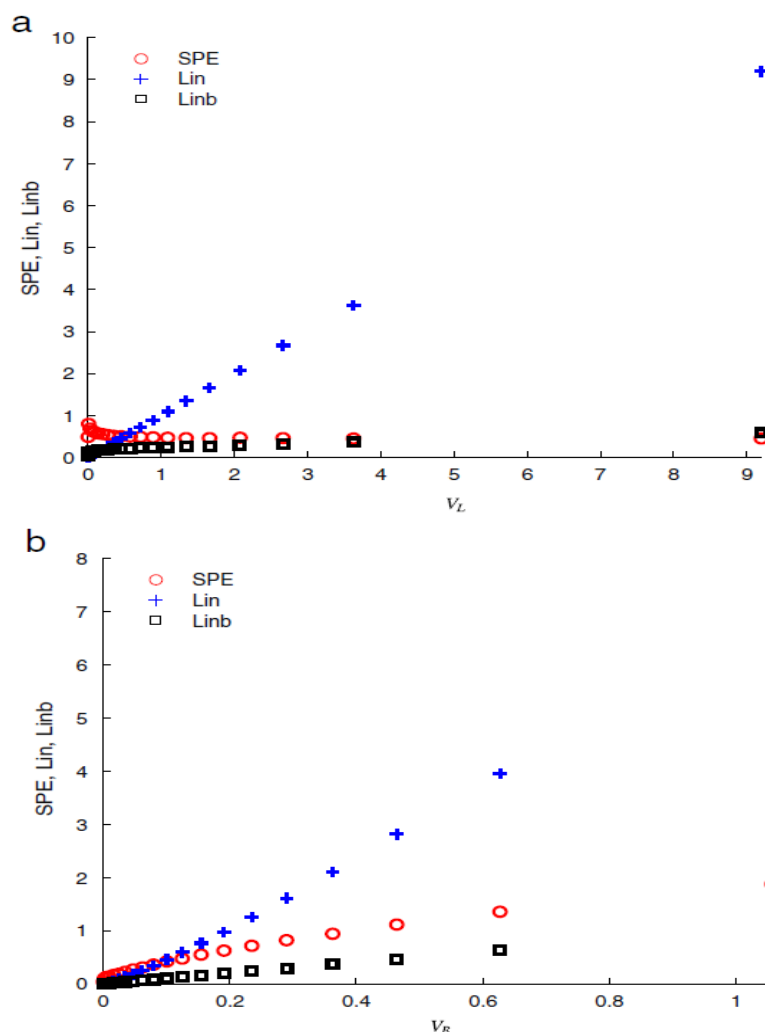


Overall, this complex case highlights the limitations and challenges faced by PLSR, Lin, and Linb methods when dealing with predictor variables with varying variances and regression contributions. Understanding these limitations is crucial in applying appropriate modelling techniques and interpreting the results accurately.

- Simulation: $k = 3, a = 2, \sigma_1^2 = \sigma_2^2 = 25, \sigma_3^2 = 1, \beta_0 = \beta_1 = \beta_2 = 1, \beta_3 = 0, \sigma_\epsilon^2 = 0,25, N = 10.000$

We deliberately chose the difficult case of PLSR in the previous simulation, and it is perhaps surprising that the linearization failed. However, that failure appears to set a dangerous precedent. We have two predictor variables with large variances and strong correlations with the responses and predictions of the three smaller variants, and no correlation in this simulation. Linb's bootstrap version is effective, but Lin's algebraic version is ineffective for some calibration sets. Fig. 3 depicts how the coefficient vector changes with b_4 (sum of squares of the first predictor) around the observed value for a calibration set. As previously stated, the linear approximation has a much narrower range of validity and results in an overly dirty variance of $\hat{\beta}$.

Fig. 3 PLSR estimates the new versus predicted variance and error (a) V_L dan (b) V_B . $k = 2$, $a = 1$, $\sigma_1^2 = 25$, $\sigma_2^2 = 1$, $\beta_1 = 0$, $\beta_0 = \beta_2 = 1$, $\sigma_\epsilon^2 = 0,25$. SPE: Squared Prediction Error $(\hat{y}_p - \hat{y}_p)^2$. Lin: $V_L + \sigma_\epsilon^2/n$. Linb: $V_B + \sigma_\epsilon^2/n$.



- Simulation: $k = 24$, $a = 7$, $\sigma_1^2 = 64$, $\sigma_2^2 = 49$, $\sigma_3^2 = 36$, $\sigma_4^2 = 25$, $\sigma_5^2 = 16$, $\sigma_6^2 = 9$, $\sigma_7^2 = 4$, $\sigma_8^2 = \dots$, $\sigma_{24}^2 = 1$, $\beta_0 = 1$, $\beta_1 = 8$, $\beta_2 = 7$, $\beta_3 = 6$, $\beta_4 = 5$, $\beta_5 = 4$, $\beta_6 = 3$, $\beta_7 = 2$, $\beta_8 = \dots$, $\beta_{24} = 1$, $\sigma_\epsilon^2 = 0,25$, $N = 10.000$

So far, the simulations have only seen a small number of predictive variables. There are $k = 24$ variables and $a = 7$ factors in this one. The first seven variables contain the majority of the x -variability and predictive power, implying that the PLSR problem is simple to solve. Lin's algebraic method fails again, yielding an extreme estimate of the calibration set's variance. Fig. 3 shows the Linb bootstrap version in good working order. Underestimates the mean squared error slightly, especially at the upper end of the scale. The difference due to bias is negligible; bootstrap estimates the variance of better $\hat{\beta}$. Linb is a bit faster to calculate in this example when N is reduced to 500, which is large enough to give the result. Because the Lin calculation involves a matrix of size $(k + 1)^2 \times (k + 1)^2$, namely 625×625 for $k = 24$. Linb is a much faster solution than Lin for much larger problems. Because little effort has been put into optimizing good code for computation, a more detailed comparison would be meaningless, but it seems reasonable to conclude that Linb is probably faster and more stable than Lin.

5. Use of a real data set.

This study was carried out through illustration/simulation with a real data set. For a real data set with k greater than 100, the real data already implemented in Linb yields a reasonable variance value in a reasonable amount of computation time. Aside from that, there isn't much to learn. The method's performance cannot be evaluated with a fixed calibration set. That's analogous to attempting to evaluate the truth of, say, the formula for the variance of the sample mean using a fixed data set. Others, on the other hand, do something with arbitrary samples but require a large data set to produce useful results.

CONCLUSION

Although (Romera, 2010) researched his linearization method, it can be concluded that it may not be a suitable approach in algebraic practice. The simulations show instances of bad failure, particularly in the calibration set. For a real data set, it is challenging to identify these bad cases easily, and the risk is that a linear approximation is highly improbable. On the other hand, the bootstrap version of the method has proven to be much easier to implement and more stable. It has also shown fairly good performance, and the simulations have demonstrated limited issues. The formulation considers variance, albeit ignoring bias, and at least the mean squared bias can be accounted for, as explained in the previous section.

REFERENCES

- Guckenheimer, J., & Holmes, P. (2013). *Nonlinear oscillations, dynamical systems, and bifurcations of vector fields* (Vol. 42). Springer Science & Business Media.
- Haff, L. R. (1979). An identity for the Wishart distribution with applications. *Journal of Multivariate Analysis*, 9(4), 531–544.
- Helland, I. S. (1988). On the structure of partial least squares regression. *Communications in Statistics-Simulation and Computation*, 17(2), 581–607.
- Mahfouf, J.-F. (1999). Influence of physical processes on the tangent-linear approximation. *Tellus A: Dynamic Meteorology and Oceanography*, 51(2), 147–166.
- Martens, H., & Naes, T. (1992). *Multivariate calibration*. John Wiley & Sons.
- Romera, R. (2010). Prediction intervals in Partial Least Squares regression via a new local linearization approach. *Chemometrics and Intelligent Laboratory Systems*, 103(2), 122–128.
- Wentzell, P. D., Cleary, C. S., & Kompany-Zareh, M. (2017). Improved modeling of multivariate measurement errors based on the Wishart distribution. *Analytica Chimica Acta*, 959, 1–14.
- Zhang, L., & Garcia-Munoz, S. (2009). A comparison of different methods to estimate prediction uncertainty using Partial Least Squares (PLS): A practitioner's perspective. *Chemometrics and Intelligent Laboratory Systems*, 97(2), 152–158.