Chemometrics and Intelligent Laboratory Systems 140 (2015) 133–140

Contents lists available at ScienceDirect

Chemometrics and Intelligent Laboratory Systems

journal homepage: www.elsevier.com/locate/chemolab

A linearization method for partial least squares regression prediction uncertainty

Ying Zhang, Tom Fearn

Department of Statistical Science, University College London, Gower Street, London WC1E 6BT, UK

ARTICLE INFO

Article history:
Received 3 October 2014
Received in revised form 11 November 2014
Accepted 20 November 2014
Available online 27 November 2014

Keywords:
Multivariate calibration
Partial least squares regression
Mean squared prediction error
Linearization parametric bootstrap
Parametric bootstrap

ABSTRACT

We study a local linearization approach put forward by Romera to provide an approximate variance for predictions in partial least squares regression. We note and correct some problems with the original formulae, study the stability of the resulting approximation using some simulations, and suggest an alternative method of computation using a parametric bootstrap. The alternative method is more stable than the algebraic approximation and is faster when the number of predictors is large.

1. Introduction

Attaching a variance to the predictions made by a partial least squares (PLS) regression model is not straightforward because the factor scores on which the linear predictor is based are themselves non-linear functions of the data. Various approximate methods have been proposed, see Zhang and Garcia-Munoz [1] for a recent review, including at least two different approaches that involve local linearizations of the prediction formula. The method of Denham (Denham [2], Seerneels et al. [3], and Phatak et al. [4]) expands about the observed value of the dependent variable. A more recent method, due to Romera [5] expands about the observed variances and covariances of all the variables in the data. This is fundamentally different from Denham's approach in that it takes into account the variability in the predictors as well as that in the response variable. In trying to implement this latter approach as part of a comparative study of methodologies, we discovered some problems with the formulae presented in Romera [5]. The current paper corrects these formulae, studies their stability, and suggests an alternative computational approach using a parametric bootstrap that is more stable and is also faster when the dimension of the explanatory variables is large.

2. Theory

Suppose we have calibration and prediction sets of data generated from the following linear models

$$\dot{\mathbf{y}}_c = \beta_0 + \dot{\mathbf{X}}_c\boldsymbol{\beta} + \epsilon, \tag{1}$$

$$\dot{\mathbf{y}}_p = \beta_0 + \dot{\mathbf{X}}_p\boldsymbol{\beta} + \epsilon, \tag{2}$$

where $\dot{\mathbf{y}}_c$ and $\dot{\mathbf{y}}_p$ are calibration and prediction set response variables, $\dot{\mathbf{X}}_c$ $(n \times k)$ and $\dot{\mathbf{X}}_p$ $(n_p \times k)$ are calibration and prediction explanatory variable matrices, β_0 and $\boldsymbol{\beta}$ $(k \times 1)$ are intercept and regression coefficients, and ϵ is the error term that has a normal distribution with mean zero and variance σ_ϵ^2. The dot on, for example, $\dot{\mathbf{y}}_c$ denotes an un-centered variable, and its corresponding centered variable is $\mathbf{y}_c$. To apply PLS regression to such data Romera [5] employs an orthogonal scores algorithm.

2.1. Orthogonal scores algorithm

The orthogonal scores algorithm by Martens and Næs [6] is simple, stable and widely used. With the number of factors chosen to be a, the i-th step of the algorithm gives the results for the i-th factor, where $i = 1, \cdots, a$.

E-mail addresses: ying.zhang@ucl.ac.uk (Y. Zhang), t.fearn@ucl.ac.uk (T. Fearn).

http://dx.doi.org/10.1016/j.chemolab.2014.11.011
0169-7439/

2.1.1. Calibration

The algorithm starts from the centered calibration data matrix, $\mathbf{X}_{c_1} = \mathbf{X}_c$.

$$\mathbf{w}_i = \mathbf{X}_{c_i}'\mathbf{y}_c$$
$$\mathbf{t}_i = \mathbf{X}_{c_i}\mathbf{w}_i$$
$$\mathbf{p}_i = \mathbf{X}_{c_i}'\mathbf{t}_i/(\mathbf{t}_i'\mathbf{t}_i)$$
$$q_i = \mathbf{y}_c'\mathbf{t}_i/(\mathbf{t}_i'\mathbf{t}_i)$$
$$\mathbf{X}_{c_{i+1}} = \mathbf{X}_{c_i} - \mathbf{t}_i\mathbf{p}_i'$$

In the i-th step of the algorithm, the column vector $\mathbf{w}_i$ $(k \times 1)$ is the weight vector defined by the covariance between $\mathbf{X}_{c_i}$ and $\mathbf{y}_c$. The $n \times a$ score matrix $\mathbf{T} = (\mathbf{t}_1 \quad \mathbf{t}_2 \quad \cdots \quad \mathbf{t}_a)$ is orthogonal. The $k \times a$ weight matrix is $\mathbf{W} = (\mathbf{w}_1 \quad \mathbf{w}_2 \quad \cdots \quad \mathbf{w}_a)$, and the $k \times a$ x-loading matrix is $\mathbf{P} = (\mathbf{p}_1 \quad \mathbf{p}_2 \quad \cdots \quad \mathbf{p}_a)$. The y-loadings vector $\mathbf{q}$ is defined as an $a \times 1$ column vector. In the first step, if $\mathbf{w}_i$ were scaled to be of length one, the algorithm would become more stable, and it would be easier to compare scores, but the normalization would not change the regression coefficient estimate. Helland [7] shows that the PLS1 regression coefficient estimates can be written as

$$\hat{\boldsymbol{\beta}} = \mathbf{W}(\mathbf{P}'\mathbf{W})^{-1}\mathbf{q}. \tag{3}$$

The scores can also be written as $\mathbf{T} = \mathbf{X}_c\mathbf{W}(\mathbf{P}'\mathbf{W})^{-1}$.

2.1.2. Prediction

A prediction $\hat{y}_p$ can be produced via the score of $\mathbf{x}_p$ $1 \times k$. In contrast to the calibration, where $\mathbf{t}_i$ is a column of $\mathbf{T}$, the predictor score $\mathbf{t}_p$ is a row vector, $\mathbf{t}_p = (t_{p_1} \quad t_{p_2} \quad \cdots \quad t_{p_a})$, and the t_{p_i} are computed recursively as

$$t_{p_i} = \mathbf{x}_{p_i}\mathbf{w}_i$$
$$\mathbf{x}_{p_{i+1}} = \mathbf{x}_{p_i} - t_{p_i}\mathbf{p}'_i$$

with $\mathbf{x}_{p_1} = \dot{\mathbf{x}}_p - \bar{\mathbf{x}}$. Equivalently, $\mathbf{t}_p = \mathbf{x}_p\mathbf{W}(\mathbf{P}'\mathbf{W})^{-1}$. The prediction is $\hat{y}_p = \bar{y} + \mathbf{t}_p\mathbf{q}$.

2.2. A random sampling model for the data

We suppose that the $(k + 1) \times 1$ vector $\dot{\mathbf{c}} = (\dot{y} \quad \dot{\mathbf{x}})'$, comprising dependent and predictor variables from one case from either the calibration or prediction set, is randomly sampled from a distribution for which the covariance of $\dot{y}$ and $\dot{\mathbf{x}}$ is $\boldsymbol{\gamma} = (\gamma_1 \quad \gamma_2 \quad \cdots \quad \gamma_k)'$, and the variance matrix of $\dot{\mathbf{x}}$ is $\boldsymbol{\Sigma}$ with elements σ_{ij}, $1 \le i, j \le k$. These parameters can be put in a $k(k + 3)/2 \times 1$ vector $\boldsymbol{\phi} = (\boldsymbol{\gamma}' \quad vecut(\boldsymbol{\Sigma})')'$, where *vecut* denotes an operator that returns a column vector whose elements are taken in order along the rows, including the diagonal elements, from the upper triangular part of a symmetric matrix. Let the $k \times 1$ vector $\mathbf{s}_{xy} = \mathbf{X}_c'\mathbf{y}_c$ and the $k \times k$ matrix $\mathbf{S}_{xx} = \mathbf{X}_c'\mathbf{X}_c$ be the sample sums of squares and products for the calibration set. Then we denote by $\mathbf{b} = (\mathbf{s}'_{xy} \quad vecut(\mathbf{S}_{xx})')'$ the vector random variable made up of these quantities, and by $\mathbf{b}_0$ the actual observed value of the random variable for a particular calibration set. The random variable $\mathbf{b}$ is an unbiased estimator of $(n - 1)\boldsymbol{\phi}$.

2.3. Romera's approach

Romera [5] explores the dependence of regression coefficients $\hat{\boldsymbol{\beta}}$ on $\mathbf{b}$ via the y-loadings $\mathbf{q}$. The estimated y-loadings can be expanded about the observed value $\mathbf{b}_0$ of $\mathbf{b}$ according to the first-order Taylor expansion

$$\mathbf{q}_\mathbf{b} \approx \mathbf{q}_{\mathbf{b}_0} + \mathbf{J}(\mathbf{b} - \mathbf{b}_0).$$

The approximate variance of the estimated y-loadings $\mathrm{Var}(\mathbf{q}) \approx \mathbf{J}\mathrm{Var}(\mathbf{b})\mathbf{J}'$, where the Jacobian matrix $\mathbf{J}$ $(a \times k(k + 3)/2)$ is the first derivative of $\mathbf{q}$ with respect to $\mathbf{b}$ evaluated at $\mathbf{b}_0$, $\mathbf{J} = (\partial\mathbf{q}/\partial\mathbf{b})_{\mathbf{b}_0}$. Romera [5] then uses $\hat{\boldsymbol{\beta}} = \mathbf{W}\mathbf{q}$ which gives $\mathrm{Var}\left(\hat{\boldsymbol{\beta}}\right) = \mathbf{W}Var(\mathbf{q})\mathbf{W}'$, so the approximate variance of $\mathbf{x}_p\hat{\boldsymbol{\beta}}$ becomes

$$\mathrm{Var}\left(\mathbf{x}_p\hat{\boldsymbol{\beta}}\right) \approx \mathbf{x}_p\mathbf{W}\mathbf{J}\mathrm{Var}(\mathbf{b})\mathbf{J}'\mathbf{W}'\mathbf{x}'_p.$$

However, there are problems with $\mathrm{Var}\left(\hat{\boldsymbol{\beta}}\right) = \mathbf{W}Var(\mathbf{q})\mathbf{W}'$. As shown in Eq. (3), $\hat{\boldsymbol{\beta}} = \mathbf{W}(\mathbf{P}'\mathbf{W})^{-1}\mathbf{q}$ for the orthogonal scores algorithm, and not $\hat{\boldsymbol{\beta}} = \mathbf{W}\mathbf{q}$, which is the result of the PLS1 orthogonal loadings algorithm. There is also a second problem, in that the weight matrix $\mathbf{W}$ is dependent on $\mathbf{b}$, so $\mathbf{W}$ cannot be treated as fixed.

2.4. Corrected formulae

Linearizing around $\mathbf{b}_0$ we have the following approximate formula for the variance of $\mathbf{x}_p\hat{\boldsymbol{\beta}}$ for fixed $\dot{\mathbf{x}}_p$

$$\mathrm{Var}\left(\mathbf{x}_p\hat{\boldsymbol{\beta}}\right) \approx \mathbf{x}_p\left(\frac{\partial\hat{\boldsymbol{\beta}}}{\partial\mathbf{b}}\right)_{\mathbf{b}_0} Var(\mathbf{b}) \left(\frac{\partial\hat{\boldsymbol{\beta}}}{\partial\mathbf{b}}\right)'_{\mathbf{b}_0}\mathbf{x}'_p = V_L. \tag{4}$$

To calculate this we need expressions for $\mathrm{Var}(\mathbf{b})$ and for $\left(\partial\hat{\boldsymbol{\beta}}/\partial\mathbf{b}\right)_{\mathbf{b}_0}$. If we assume that the $\dot{\mathbf{c}}$ defined in Section 2.2 is normally distributed, both the distribution and the variance of $\mathbf{b}$ are known from standard normal theory. Appendix A gives the distribution of $\mathbf{b}$. The algebra for $\left(\partial\hat{\boldsymbol{\beta}}/\partial\mathbf{b}\right)_{\mathbf{b}_0}$ is in Appendix B.

2.5. Estimating $\mathrm{Var}\left(\hat{\boldsymbol{\beta}}\right)$ *by a parametric bootstrap*

An alternative approach that avoids all the algebra is to use a parametric bootstrap to estimate $\mathrm{Var}\left(\hat{\boldsymbol{\beta}}\right)$. For the m-th bootstrap sample $(m = 1, \ldots, M)$, a sum of squares and products matrix is drawn from the Wishart distribution in Appendix A and $\mathbf{b}_m$ is extracted from it. Now we need to calculate $\hat{\boldsymbol{\beta}}_m^B$ from $\mathbf{b}_m$, rather than from $\mathbf{X}_c$ and $\mathbf{y}_c$. The formula for doing this was given by Romera [5] and are presented in Appendix C. The variance of regression coefficients from the bootstrap algorithm is $Var\left(\hat{\boldsymbol{\beta}}^B\right) = \frac{n}{n+1}\frac{1}{M-1}\sum_{m=1}^{M}\left(\hat{\boldsymbol{\beta}}_m^B - \overline{\boldsymbol{\beta}}\right)\left(\hat{\boldsymbol{\beta}}_m^B - \overline{\boldsymbol{\beta}}\right)'$, where $\overline{\boldsymbol{\beta}} = \frac{1}{M}\sum_{m=1}^{M}\hat{\boldsymbol{\beta}}_m^B$ and the factor $\frac{n}{n+1}$ adjusts for the bias in the bootstrap (See Efron and Tibshirani [8]). The approximate variance of $\mathbf{x}_p\hat{\boldsymbol{\beta}}$ is

$$\mathrm{Var}\left(\mathbf{x}_p\hat{\boldsymbol{\beta}}\right) \approx \mathbf{x}_p\mathrm{Var}\left(\hat{\boldsymbol{\beta}}^B\right)\mathbf{x}'_p = V_B. \tag{5}$$

3. Numerical experiments

In this section, we use simulation studies to investigate how the linearization method and its bootstrap version perform under different conditions. Our purpose is not to carry out an extensive simulation study, but to demonstrate some of the properties of the method using a few simple simulations. Each of the N repetitions in the simulation generates a calibration set of size $n = 200$ and a prediction set of size $n_p = 200$ using the models in Eqs. (1) and (2) but with $\in$ set to zero in Eq. (2). Taking the additive noise component out of the predictions enables the performance of the variance formulae in Eqs. (4) and (5) to be seen more clearly. The explanatory variables are independently and normally distributed with mean 0 and variances $(\sigma_1^2 \quad \sigma_2^2 \quad \cdots \quad \sigma_k^2)$ in both calibration and prediction sets. The number of PLS factors is fixed to be a in each of the repetitions. Of course an extensive simulation study would need to explore both correlated predictors and the effect of extrapolation, but our purpose here is just to demonstrate some of the properties of the methods investigated using a few simple simulations.

For each of the $N \times n_p$ predictions in the simulation we calculate a squared prediction error and the estimated variances V_L and V_B given