A FAST METHOD TO COMPUTE ORTHOGONAL LOADINGS PARTIAL LEAST SQUARES

Constantinos Goutis

95-55

WORKING PAPERS

Universidad Carlos III de Madrid

Working Paper 95-55
Statistics and Econometrics Series 21
November 1995

Departamento de Estadística y Econometría
Universidad Carlos III de Madrid
Calle Madrid 126
28903 Getafe (Spain)
Fax (341) 624-9849

A FAST METHOD TO COMPUTE ORTHOGONAL LOADINGS PARTIAL LEAST SQUARES

Constantinos Goutis*

Abstract

We give a computationally fast method for the orthogonal loadings partial least squares. Our algorithm avoids the multiple regression computations at each step and yields identical scores and loadings to the usual method. We give a proof of the equivalence to the standard algorithm. We discuss briefly the computational advantages over both orthogonal scores and orthogonal loadings partial least squares.

Key words and phrases : Biased regression methods, bilinear forms, calibration, projection matrices.

* Departamento de Estadística y Econometría, Universidad Carlos III de Madrid,
tel. (341) 624-9852, e-mail: *costas@est-econ.uc3m.es.*

1 Introduction

There are two commonly used partial least squares regression algorithms[1,2,3]. The first one, the orthogonal scores algorithm, is the oldest one[4], whereas the second one, the orthogonal loadings algorithm, is more recent[5]. Their difference lies in the decomposition of the data matrix $\mathbf{X} = \sum_{i=1}^{r} \mathbf{t}_i \mathbf{p}_i^T$. The first one requires orthogonality of the vectors $\mathbf{t}_i$ whereas the second one of the vectors $\mathbf{p}_i$. As it turns out the two methods are equivalent for prediction purposes[1]. In the usual formulation, the first algorithm is easier computationally, though not always faster[6] and is typically used in computer packages[7,8]. This is because it does not require any multiple regression. The second one is easier to interpret and study theoretically[5].

The purpose of this note is to give a fast method to compute the scores and the loadings of the second algorithm. Our method avoids the multiple regression step and, as we will see, it requires significantly fewer computations than both the orthogonal scores and the orthogonal loadings algorithms. Furthermore, it is trivial to program. For simplicity we will consider only the univariate case though extensions to multivariate PLS seem possible.

2 Orthogonal Loadings PLS methods

Consider data in the form of a columnwise centered $n \times k$ matrix $\mathbf{X}$ and a centered $n \times 1$ vector $\mathbf{y}$. In the usual formulation[2] the PLS steps are as follows:

Set $\mathbf{E}_0 = \mathbf{X}$ and $\mathbf{f}_0 = \mathbf{y}$.
For $i = 1, 2, \ldots, r$:

$$\begin{aligned}
\mathbf{p}_i &= \frac{\mathbf{E}_{i-1}^T \mathbf{f}_{i-1}}{\sqrt{\mathbf{f}_{i-1}^T \mathbf{E}_{i-1} \mathbf{E}_{i-1}^T \mathbf{f}_{i-1}}} && (1)\\
\mathbf{t}_i &= \mathbf{E}_{i-1} \mathbf{p}_i && (2)\\
\mathbf{E}_i &= \mathbf{E}_{i-1} - \mathbf{t}_i \mathbf{p}_i^T && (3)\\
\mathbf{f}_i &= \mathbf{y} - \mathbf{T}_i (\mathbf{T}_i{}^T \mathbf{T}_i)^{-1} \mathbf{T}_i{}^T \mathbf{y}, && (4)
\end{aligned}$$

where $\mathbf{T}_i = (\mathbf{t}_1\, \mathbf{t}_2\, \ldots\, \mathbf{t}_i)$. We propose to replace the above algorithm by:

Set $\mathbf{E}_0 = \mathbf{X}$ and $\mathbf{g}_0 = \mathbf{y}$.
For $i = 1, 2, \ldots, r$:

$$\begin{aligned}
\mathbf{p}_i &= \frac{\mathbf{E}_{i-1}^T \mathbf{g}_{i-1}}{\sqrt{\mathbf{g}_{i-1}^T \mathbf{E}_{i-1} \mathbf{E}_{i-1}^T \mathbf{g}_{i-1}}} && (5)\\
\mathbf{t}_i &= \mathbf{E}_{i-1} \mathbf{p}_i && (6)\\
\mathbf{E}_i &= \mathbf{E}_{i-1} - \mathbf{t}_i \mathbf{p}_i^T && (7)\\
\mathbf{g}_i &= \mathbf{t}_i. && (8)
\end{aligned}$$

Clearly this can be written in a more concise way, by merging for example (6) and (8). However, we prefer the above form since it facilitates the comparison to (1)–(4). The two algorithms are equivalent in the sense that they decompose $\mathbf{X}$ into identical bilinear forms. Before proving this, we define some notation and state a few lemmas that will be needed in the proof. We will denote the range and the null space of a matrix $\mathbf{E}$ by $\mathcal{R}(\mathbf{E})$ and $\mathcal{N}(\mathbf{E})$ respectively and its Moore-Penrose inverse by $\mathbf{E}^+$. For $i = 1, 2, \ldots, r$ and $\mathbf{t}_i$, $\mathbf{E}_i$ and $\mathbf{f}_i$ given by (2)–(4), we define $\mathbf{K}_i$, $\mathbf{L}_i$, $\mathbf{M}_i$ and $\mathbf{N}_i$ to be the projection matrices onto $\mathcal{N}(\mathbf{T}_i)$, $\mathcal{R}(\mathbf{E}_i)$, $\mathcal{N}(\mathbf{T}_i) \cap \mathcal{R}(\mathbf{E}_{i-1})$ and $\mathcal{R}(\mathbf{T}_i) \cap \mathcal{R}(\mathbf{E}_{i-1})$ respectively. As a convention, we take $\mathbf{M}_0$ to be the identity matrix. With this notation we have the following two general lemmas. Their proofs are straightforward applications of matrix algebra.

Lemma 2.1 *Suppose that* $\mathbf{f}$ *and* $\mathbf{g}$ *are vectors and* $\hat{\mathbf{f}}$ *and* $\hat{\mathbf{g}}$ *are the corresponding projections onto* $\mathcal{R}(\mathbf{E})$ *for some matrix* $\mathbf{E}$. *Then*
(a) $\mathbf{E}^T\mathbf{f} = \mathbf{E}^T\hat{\mathbf{f}}$ *and*
(b) *if* $\mathbf{E}^T\mathbf{f} = \mathbf{E}^T\mathbf{g}$ *then* $\hat{\mathbf{f}} = \hat{\mathbf{g}}$.

Lemma 2.2 *For any matrix* $\mathbf{E}$

$$\left(\mathbf{E}^+\right)^T = \left(\mathbf{E}\mathbf{E}^T\right)^+\mathbf{E} \tag{9}$$
$$\left(\mathbf{E}^+\right)^T\mathbf{E}^+ = \left(\mathbf{E}\mathbf{E}^T\right)^+. \tag{10}$$

The next two Lemmas refer to the orthogonal loadings steps.

Lemma 2.3 *If* $\mathbf{E}_i$ *and* $\mathbf{E}_{i+1}$ *are given by consecutive steps of the method (1)–(4) then*

$$\mathcal{R}(\mathbf{E}_i) = \mathcal{R}(\mathbf{E}_{i+1}) + \mathcal{R}(\mathbf{t}_{i+1}) \tag{11}$$

and, hence, $R(\mathbf{E}_{i+1})$ *is a subspace of* $R(\mathbf{E}_i)$.

Proof. Since $\mathbf{p}_{i+1} \in \mathcal{R}(\mathbf{E}_i^T)$ and $\mathbf{E}_i^+\mathbf{E}_i$ is the projection matrix onto $\mathcal{R}(\mathbf{E}_i^T)$, it follows that $\mathbf{p}_{i+1} = \mathbf{E}_i^+\mathbf{t}_{i+1}$. Step (3) becomes

$$\mathbf{E}_{i+1} = \mathbf{E}_i - \mathbf{t}_{i+1}\mathbf{t}_{i+1}^T\left(\mathbf{E}_i^+\right)^T \tag{12}$$
$$= \mathbf{E}_i - \mathbf{t}_{i+1}\mathbf{t}_{i+1}^T\left(\mathbf{E}_i\mathbf{E}_i^T\right)^+\mathbf{E}_i. \tag{13}$$

where equality (13) follows from (9). Now from (10) and the normalisation $\mathbf{p}_{i+1}^T\mathbf{p}_{i+1} = 1$, we can see that the matrix $\mathbf{t}_{i+1}\mathbf{t}_{i+1}^T(\mathbf{E}_i\mathbf{E}_i^T)^+$ is idempotent and, hence, a (non-orthogonal) projection matrix onto its range which is $\mathcal{R}(\mathbf{t}_{i+1})$. This and the last step (13) show that the columns of $\mathbf{E}_{i+1}$ are the residuals from projections of the columns of $\mathbf{E}_i$ onto $\mathbf{t}_{i+1}$. The result follows immediately. □

Lemma 2.4 *If* $\mathbf{E}_i$, $\mathbf{E}_{i+1}$ *and* $\mathbf{f}_i$ *are given as in (1)–(4) then* $\mathbf{E}_{i+1}\mathbf{L}_i\mathbf{f}_i = 0$.

Proof. After expressing $\mathbf{E}_{i+1}$ and $\mathbf{L}_i$ in terms of $\mathbf{E}_i$ and $\mathbf{f}_i$ and substituting, we obtain

$$\begin{aligned}
\mathbf{E}_{i+1}\mathbf{L}_i\mathbf{f}_i &= \left(\mathbf{E}_i^T - \frac{\mathbf{E}_i^T\mathbf{f}_i\mathbf{f}_i^T\mathbf{E}_i\mathbf{E}_i^T}{\mathbf{f}_i^T\mathbf{E}_i\mathbf{E}_i^T\mathbf{f}_i}\right)\mathbf{E}_i\left(\mathbf{E}_i^T\mathbf{E}_i\right)^+\mathbf{E}_i^T\mathbf{f}_i && (14)\\
&= \mathbf{E}_i^T\mathbf{E}_i\left(\mathbf{E}_i^T\mathbf{E}_i\right)^+\mathbf{E}_i^T\mathbf{f}_i - \frac{\mathbf{E}_i^T\mathbf{f}_i\mathbf{f}_i^T\mathbf{E}_i\mathbf{E}_i^T\mathbf{E}_i\left(\mathbf{E}_i^T\mathbf{E}_i\right)^+\mathbf{E}_i^T\mathbf{f}_i}{\mathbf{f}_i^T\mathbf{E}_i\mathbf{E}_i^T\mathbf{f}_i} && (15)\\
&= \mathbf{E}_i^T\mathbf{f}_i - \frac{\mathbf{E}_i^T\mathbf{f}_i\mathbf{f}_i^T\mathbf{E}_i\mathbf{E}_i^T\mathbf{f}_i}{\mathbf{f}_i^T\mathbf{E}_i\mathbf{E}_i^T\mathbf{f}_i} = 0, && (16)
\end{aligned}$$

where Lemma (2.1a) was used to simplify (15). □

Now our main Theorem can be stated as follows:

Theorem 2.1 *The vectors* $\mathbf{p}_i$ *and* $\mathbf{t}_i$, $i = 1, 2, \ldots, r$, *given by the algorithm (1)–(4) are identical to the ones given by the algorithm (5)–(8).*

Proof. It suffices to show that for the method (1)–(4) the vector $\mathbf{E}_i^T\mathbf{f}_i$ is a multiple of the vector $\mathbf{E}_i^T\mathbf{t}_i$ for all i.

We have

$$\begin{aligned}
\mathbf{E}_i^T\mathbf{f}_i &= \mathbf{E}_i^T\mathbf{K}_i\mathbf{y} && \text{by (4)} && (17)\\
&= \mathbf{E}_i^T\mathbf{L}_i\mathbf{K}_i\mathbf{y} && \text{using Lemma (2.1a)} && (18)\\
&= \mathbf{E}_i^T\mathbf{L}_i\mathbf{L}_{i-1}\mathbf{K}_i\mathbf{y} && \text{since, by Lemma (2.3), } \mathbf{L}_i \text{ projects} && (19)\\
& && \text{onto a subspace of } \mathcal{R}(\mathbf{L}_{i-1}) && \\
&= \mathbf{E}_i^T\mathbf{L}_i(\mathbf{M}_i + \mathbf{N}_i)\mathbf{K}_i\mathbf{y} && && (20)\\
&= \mathbf{E}_i^T\mathbf{L}_i\mathbf{M}_i\mathbf{K}_i\mathbf{y} && \text{since } \mathcal{N}(\mathbf{T}_i) \cap \mathcal{R}(\mathbf{T}_i) = 0 && (21)\\
&= \mathbf{E}_i^T\mathbf{L}_i\mathbf{M}_i\mathbf{y} && \text{since } \mathbf{M}_i \text{ projects onto a subspace of } \mathcal{N}(\mathbf{T}_i) && (22)\\
&= \mathbf{E}_i^T\mathbf{M}_i\mathbf{y} && \text{using Lemma (2.1a).} && (23)
\end{aligned}$$

From Lemma (2.1b), it follows that $\mathbf{L}_i\mathbf{f}_i = \mathbf{L}_i\mathbf{M}_i\mathbf{y}$ and since this vector belongs to $\mathcal{R}(\mathbf{E}_i)$, it can be written as the sum of its projections onto $\mathcal{R}(\mathbf{M}_{i+1})$ and $\mathcal{R}(\mathbf{N}_{i+1})$. Therefore

$$\mathbf{L}_i\mathbf{f}_i = \mathbf{M}_{i+1}\mathbf{L}_i\mathbf{M}_i\mathbf{y} + \mathbf{N}_{i+1}\mathbf{L}_i\mathbf{M}_i\mathbf{y}. \qquad (24)$$

The first term of the above sum can be written as

$$\begin{aligned}
\mathbf{M}_{i+1}\mathbf{L}_i\mathbf{M}_i\mathbf{y} &= \mathbf{M}_{i+1}\mathbf{M}_i\mathbf{y} && \text{since } \mathbf{M}_{i+1} \text{ projects onto a subspace of } \mathcal{R}(\mathbf{E}_i) && (25)\\
&= \mathbf{M}_{i+1}\mathbf{y} && \text{since } \mathcal{R}(\mathbf{M}_{i+1}) \text{ is a subspace of } \mathcal{R}(\mathbf{M}_i). && (26)
\end{aligned}$$

On the other hand, by applying Lemma (2.3) recursively, we see that none of $\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_i$ belong to $\mathcal{R}(\mathbf{E}_i)$ whereas $\mathbf{t}_{i+1}$ does, so

$$\mathcal{R}(\mathbf{N}_{i+1}) = \mathcal{R}(\mathbf{T}_{i+1}) \cap \mathcal{R}(\mathbf{E}_i) = \mathcal{R}(\mathbf{t}_{i+1}) \cap \mathcal{R}(\mathbf{E}_i) = \mathcal{R}(\mathbf{t}_{i+1}), \qquad (27)$$

and the second term of the sum in (24) has the form $\alpha\,\mathbf{t}_{i+1}$ for some α. Premultiplying (24) by $\mathbf{E}_{i+1}^T$ and applying Lemma (2.4) we obtain

$$\begin{aligned} 0 &= \mathbf{E}_{i+1}^T\mathbf{M}_{i+1}\mathbf{y} + \alpha\,\mathbf{E}_{i+1}^T\mathbf{t}_{i+1} & (28)\\ &= \mathbf{E}_i^T\mathbf{M}_i\mathbf{y} + \alpha\,\mathbf{E}_i^T\mathbf{t}_i & (29)\\ &= \mathbf{E}_i^T\mathbf{f}_i + \alpha\,\mathbf{E}_i^T\mathbf{t}_i, & (30) \end{aligned}$$

where (29) is a simple change of label and (30) follows from (17)-(23). Hence we obtain $\mathbf{E}_i^T\mathbf{f}_i = -\alpha\,\mathbf{E}_i^T\mathbf{t}_i$ and the theorem is proved. □

3 Discussion

It is clear that the method presented here has advantages in terms of speed of computation. Indeed, the number of numerical operations that it performs is even smaller than that of the orthogonal scores algorithm since the latter needs an extra step to compute $\mathbf{p}_i$. The computational advantages over the classical orthogonal loading method are more substantial for large n and for a large number of factors. Compared to the orthogonal scores method, it is significantly faster for large k. The speeding of computation would help most when applying computer intensive methods such as cross-validation.

However, the mathematical equivalence of the two orthogonal loadings algorithms does not necessarily imply that the answers will be the same. There is always numerical error which can rapidly accumulate. Hence one must also examine the numerical stability. To do so, we implemented the method with some simulated data using a GAUSS program on a PC. Although we do not claim to be conclusive, it seemed to perform satisfactorily and we would feel fairly confident to use it.

REFERENCES

1. I. S. Helland, *Commun. Statist.- Simul. Comput.* **17**, 581-607 (1988).
2. H. Martens and T. Næs, *Multivariate Calibration*, Wiley, Chichester (1989).
3. T. Næs, C. Irgens and H. Martens, *Appl. Statist.* **35**, 195–206 (1986).
4. S. Wold, H. Martens and H. Wold, *Proc. Conf. Matrix Pencils*, 286–293 (1983).
5. T. Næs and H. Martens, *Commun. Statist.- Simul. Comput.* **14**, 545–576 (1985).
6. M. C. Denham, *Statist. Comput.* **5**, 191-202 (1995).
7. S. Wold, SIMCA3B, University of Umeå, Sweden (1981).
8. H. Martens, UNSCRAMBLER, Norwegian Food Research Institute, Norway (1984).