Supplementary material for A statistical view on "Predicting reaction performance in C–N cross-coupling using machine learning"

Tatyana Krivobokova[1] Universität Wien
Boris Maryasin[2] Universität Wien
Gianluca Finocchio[1] Universität Wien

September 14, 2022

1 Generalised linear models with continuous Bernoulli distribution

The density of the continuous Bernoulli distribution is given by

$$f(y;p) = \begin{cases} \frac{\log\{(1-p)/p\}}{1-2p} p^y (1-p)^{1-y}, & p \in (0,1),\ p \neq 0.5 \\ 2\, p^y (1-p)^{1-y}, & p = 0.5, \end{cases}$$

for $y \in [0,1]$. Rewriting

$$f(y;p) = \exp\left[y \log\left(\frac{p}{1-p}\right) + \log(1-p) - \log(1-2p) + \log\left\{\log\left(\frac{1-p}{p}\right)\right\}\right]$$

leads to the canonical parametrisation $f(y;\eta) = \exp\{y\eta - \kappa(\eta)\}$ with

$$\eta = \log\left(\frac{p}{1-p}\right) \quad \text{and} \quad \kappa(\eta) = \log\{\exp(\eta) - 1\} - \log(\eta).$$

[1]Department of Statistics and Operations Research, Oskar-Morgenstern-Platz 1, 1090 Wien
[2]Department of Organic Chemistry, Währinger Straße 38, 1090 Wien

This implies for $Y \sim f(y;\eta)$

$$\begin{aligned}\kappa^{'}(\eta) &= \frac{1}{1-\exp(-\eta)} - \frac{1}{\eta} = \mathrm{E}(Y) \\ \kappa^{''}(\eta) &= \frac{1}{\eta^2} - \frac{\exp(\eta)}{\{1-\exp(\eta)\}^2} = \mathrm{var}(Y).\end{aligned}$$

Let now $Y = (Y_1, \ldots, Y_n)$ be a vector independent random variables, such that Y_i is distributed with the density $f(y_i;\eta_i)$, $i = 1, \ldots, n$. Denote also $X \in \mathbb{R}^{n\times p}$ a matrix of covariates. To link the response vector Y and covariates X we employ a canonical link function, that is, $g = (\kappa^{'})^{-1}$, so that $g\{\mathrm{E}(Y_i)\} = g\{\kappa^{'}(\eta_i)\} = \eta = X_i\beta$, where X_i denotes the i-th row of matrix X. This link function is not available analytically for the continuous Bernoulli distribution. However, when the link is canonical, one does not need to know the form of g for the estimation of β. The Fisher scoring algorithm for estimation of β is given by

$$\hat{\beta}_{k+1} = \hat{\beta}_k + F(\hat{\beta}_k)^{-1} S(\hat{\beta}_k), \;\; k = 0, 1, 2, \ldots$$

for $F(\beta) = X^t \mathrm{diag}\{\kappa^{''}(X_1\beta), \ldots, \kappa^{''}(X_n\beta)\}X$ and $S(\beta) = X^t\{Y - \mathrm{E}(Y)\}$, where $\mathrm{E}(Y) = \{\kappa^{'}(X_1\beta), \ldots, \kappa^{'}(X_n\beta)\}^t$.

Once β is estimated, the estimator for the expectation is obtained via

$$\widehat{\mathrm{E}(Y_i)} = \kappa^{'}(X_i\hat{\beta}) = \frac{1}{1-\exp(-X_i\hat{\beta})} - \frac{1}{X_i\hat{\beta}}.$$

Function $\kappa^{'}(\cdot)$ is a monotone increasing function in $\hat{\beta}$, so that the interpretation of the regression coefficients is similar to that in linear regression.

2 Matrix of descriptors

First let us recall a general ANOVA model and its representation as a regression model. For sake of simplicity consider a two-factor ANOVA model; generalisation to four factors is straightforward, but requires more complicated notations. Assume there are two factors A and B, where factor A has I levels $A_1, \ldots, A_I$ and factor B has J levels $B_1, \ldots, B_j$. For each combination of these levels K values y_{ijk} is observed. The corresponding ANOVA model is given by

$$y_{ijk} = \mu_{ij} + \epsilon_{ijk} = \mu_0 + \alpha_i + \beta_j + (\alpha\beta)_{ij} + \epsilon_{ijk}, \quad i = 1, \ldots, I, \quad j = 1, \ldots, J, \quad k = 1, \ldots, K,$$

where μ_0 describes an overall mean, α_i is a deviation of i-th level of factor A from μ_0, β_j is a deviation of j-th level of factor B from μ_0 and $(\alpha\beta)_{ij}$ is a deviation of an interaction term. For identifiability, it is assumed that $\sum_i \alpha_i = \sum_j \beta_j = \sum_i (\alpha\beta)_{ij} = \sum_j (\alpha\beta)_{ij} = 0$. This model can be written as a regression model

$$\begin{aligned} y_{ijk} &= \mu_{ij} + \epsilon_{ijk} = \mu_0 + \alpha_1 x_1^A + \ldots + \alpha_{I-1} x_{I-1}^A + \beta_1 x_1^B + \ldots + \beta_{J-1} x_{J-1}^b \\ &+ (\alpha\beta)_{11} x_{1,1}^{AB} + \ldots + (\alpha\beta)_{I-1,J-1} x_{I-1,J-1}^{AB} + \epsilon_{ijk}, \end{aligned}$$

where

$$x_l^A = \begin{cases} 1, & \text{for } i = 1, \ldots, I-1 \\ -1, & \text{for } i = I \\ 0, & \text{otherwise} \end{cases}$$

$l = 1, \ldots, I-1$ and

$$x_m^B = \begin{cases} 1, & \text{for } j = 1, \ldots, J-1 \\ -1, & \text{for } j = J \\ 0, & \text{otherwise} \end{cases}$$

$m = 1, \ldots, J-1$ and $x^{AB}_{l,m} = x^A_l \cdot x^B_m$.

For example, for $I = 2$, $J = 3$ and $K = 1$ one has the data

	B_1	B_2	B_3
A_1	y_{11}	y_{12}	y_{13}
A_2	y_{21}	y_{22}	y_{23}.

Since per factor level combination there is only one observation available, the interaction term can not be estimated reliably and for the moment is assumed to be zero. The corresponding regression model without an interaction term results in

$$Y = \begin{pmatrix} y_{11} \\ y_{12} \\ y_{13} \\ y_{21} \\ y_{22} \\ y_{23} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & 0 \\ 1 & -1 & 0 & 1 \\ 1 & -1 & -1 & -1 \end{pmatrix} \begin{pmatrix} \mu_0 \\ \alpha_1 \\ \beta_1 \\ \beta_2 \end{pmatrix} + \begin{pmatrix} \epsilon_{11} \\ \epsilon_{12} \\ \epsilon_{13} \\ \epsilon_{21} \\ \epsilon_{23} \\ \epsilon_{33} \end{pmatrix} =: X\mu + \epsilon$$

From the identifiability conditions one finds $\alpha_2 = -\alpha_1$ and $\beta_3 = -\beta_1 - \beta_2$. To build interaction terms (in case $K > 1$), one just needs to multiply corresponding columns of matrix X.

Let us now show that a linear model based on chemical descriptors is equivalent to an ANOVA model. For simplicity, let us consider only two factors, extension to four factors is straightforward. Let D denote the (3960×74)-dimensional matrix of descriptors (both training and test sets, $3960 = 22 \times 15 \times 3 \times 4$), which is partitioned into a sub-matrix D_b that contains 10 descriptors of the factor base and a sub-matrix D_l that contains 64 descriptors of factor ligand. Let also Y denote a 3960-dimensional vector of yields. Then a linear model for the yield can be written as $Y = D\beta + \epsilon = D_b\beta_b + D_l\beta_l + \epsilon$, where $\beta = (\beta_b, \beta_l)$ is an unknown vector of coefficients.

Now, matrix D_b contains only three distinct rows, see the first five rows of D_b:

base1	base2	base3	base4	base5	base6	base7	base8	base9	base10
1.397	0.046	0.570	-0.157	-0.463	1.414	-0.939	-0.812	-1.056	-0.958
-0.890	-1.247	-1.406	-1.138	1.389	-0.707	-0.446	-0.597	-0.286	-0.422
-0.507	1.201	0.836	1.296	-0.926	-0.707	1.385	1.409	1.342	1.380
-0.890	-1.247	-1.406	-1.138	1.389	-0.707	-0.446	-0.597	-0.286	-0.422
-0.890	-1.247	-1.406	-1.138	1.389	-0.707	-0.446	-0.597	-0.286	-0.422

Hence, the row rank of D_b is three and therefore its column rank equals to three as well. Similarly, the rank of D_l equals to four. Therefore, one can leave only first three columns of D_b and first four columns of D_l in the linear model. Let us denote these reduced matrices by $\widetilde{D}_b$ and $\widetilde{D}_l$, respectively, so that the linear model becomes now $Y = \widetilde{D}_b\widetilde{\beta}_b + \widetilde{D}_l\widetilde{\beta}_l + \epsilon$, where $\widetilde{\beta}_b$ contains the first three components of β_b and $\widetilde{\beta}_l$ contains first four elements of β_l. Now, let us denote by B a 3×3 matrix, that consists of three distinct rows of $\widetilde{D}_b$ and by L a 4×4 matrix that consists of four distinct rows of $\widetilde{D}_l$. Both matrices are square, of full-rank and hence invertible. Therefore,

$$\widetilde{D}_b\widetilde{\beta}_b + \widetilde{D}_l\widetilde{\beta}_l = \widetilde{D}_b B^{-1} B\widetilde{\beta}_b + \widetilde{D}_l L^{-1} L\widetilde{\beta}_l =: X_b(B\widetilde{\beta}_b) + X_l(L\widetilde{\beta}_l) =: X_b\mu^b + X_l\mu^l,$$

where the i-th row of X_b is given by

$$X_{b,i} = \begin{cases} (1,0,0), & \text{if } \widetilde{D}_{b,i} = \widetilde{D}_{b,1} \\ (0,1,0), & \text{if } \widetilde{D}_{b,i} = \widetilde{D}_{b,2} \\ (0,0,1), & \text{if } \widetilde{D}_{b,i} = \widetilde{D}_{b,3}, \end{cases}$$

with $\widetilde{D}_{b,i}$ denoting the i-th row of $\widetilde{D}_b$. Matrix X_l is constructed in the same way and has four columns. With this, the linear model $Y = X_b\mu^b + X_l\mu^l + \epsilon$ corresponds

to the ANOVA model without an interaction term

$$\text{yield}_{ijk} = \mu_i^b + \mu_j^l + \epsilon_{ijk}, \quad i = 1, 2, 3, \quad j = 1, 2, 3, 4, \quad k = 1, \ldots, 330,$$

which can be re-parametrised to

$$\text{yield}_{ijk} = \mu_0 + \alpha_i + \beta_j + \epsilon_{ijk}, \quad i = 1, 2, 3, \quad j = 1, 2, 3, 4, \quad k = 1, \ldots, 330.$$

with the constraints $\sum_i \alpha_i = \sum_j \beta_j = 0$ for identifiability. As before, μ_0 is an overall mean, α_i is the deviation of the i-th level of factor base from μ_0 and β_j is the deviation of the j-th level of ligand from μ_0. This proves our claim. For the numerical verification see R files provided.

From these considerations it is easy to see that in general, independent of the model, the matrix of chemical descriptors is up to a linear transformation equal to a matrix X, which encodes with dummy variables levels of corresponding factors. Indeed, let $C = \text{blockdiag}(B^{-1}, L^{-1})$, which is 7×7 dimensional. Then, it holds that $\widetilde{D}C = (\widetilde{D}_b, \widetilde{D}_l)C = (X_b, X_l) = X$.

The original matrix of chemical descriptors contains only 19 descriptors of additive, which take 22 distinct values. It can easily be checked that the estimators based on the matrix with 19 columns and the estimators based on 22 columns are very close, see the R code provided. Therefore, all the results obtained with the original covariates matrix of chemical descriptors is nearly equivalent to an ANOVA model with four factors.

Since in practice a dummy matrix that corresponds to the given experimental design can be built directly, there is no need to use the matrix of chemical descriptors, which lacks three columns for additive and is ill-conditioned.

3 Generalised partial least squares

The PLS algorithm proposed by Wold (1966) fits a linear model between a given response vector $y \in \mathbb{R}^n$ and a set of latent variables obtained by projecting a given covariates matrix $X \in \mathbb{R}^{n \times p}$ onto a suitable latent subspace. By construction, for any integer $1 \leq s \leq p$, the coefficients $\widehat{\beta}_s$ estimated by PLS belong to the s-dimensional Krylov space $\mathcal{K}_s(X^\top y; X^\top X) := span\{X^\top y, (X^\top X)X^\top y, \ldots, (X^\top X)^{s-1}X^\top y\}$. One can see that the Krylov space depends on both X and y and accounts for the covariance between the features and the response. Let $\widehat{R}_s \in \mathbb{R}^{p \times s}$ be the orthogonal projection onto the s-dimensional latent space, that is, there is a unique $\widehat{\alpha}_s \in \mathbb{R}^s$ such that $\widehat{\beta}_s = \widehat{R}_s \widehat{\alpha}_s$. The estimated response is

$$\widehat{y}_s = X\widehat{\beta}_s = X\widehat{R}_s\widehat{\alpha}_s = \widehat{T}_s\widehat{\alpha}_s$$

and is the result of a linear model on the latent design $\widehat{T}_s := X\widehat{R}_s$. In this sense, PLS is a supervised dimension reduction method for linear models.

The idea behind our extension of PLS to generalized linear models is straightforward, although its technicalities are nontrivial: first, it is well-known that maximum likelihood estimation in GLMs can be achieved via iteratively-reweighted-least-squares (IRLS) by McCullagh (1989); second, one can think of PLS as a regularization tool for solving ill-posed least-squares problems. Therefore, an iteratively-reweighted-PLS (IRPLS) algorithm seems a natural procedure to regularize maximum likelihood estimation in ill-posed GLMs.

A pseudo-implementation is given in Algorithm 1 below. One inputs a pair (X, y) of observed data and an integer $1 \leq s \leq p$, which is a latent dimension parameter. Then, the iterations of the classical IRLS algorithm are performed, but instead of solving the required weighted-least-squares problem, we fit PLS with s latent

components. We run the algorithm for a maximum number $J \geq 0$ of iterations or until a certain threshold $\varepsilon \geq 0$ is hit by our stopping rule (likelihood ratio), both these parameters are fed to the algorithm together with the data input. Additional hyper-parameters determine whether the data should be centered/scaled and include an intercept in the coefficients. In our empirical findings, no more than $J = 20$ iterations and no less than $\varepsilon = 0.05$ are ever necessary. Even in high-dimensional settings, it is often the case that $s = 30$ latent components are sufficient.

Algorithm 1: IRPLS

Input: X, y, s, ε, J;

Output: $(\widehat{\beta}_s^{(j)})_{j=0}^{J}$;

$j \leftarrow 0$;

$\widehat{\beta}_s^{(j)} \leftarrow 0_p$;

$\widehat{D}^{(j)} \leftarrow \infty$;

while $\widehat{D}^{(j)} > 1 + \varepsilon$ **and** $j < J$ **do**

$\widehat{\eta}^{(j)} \leftarrow X\widehat{\beta}_s^{(j)}$, $\widehat{\mu}^{(j)} \leftarrow \kappa'(\widehat{\eta}^{(j)})$;

$\widehat{W}^{(j)} \leftarrow diag(\kappa''(\widehat{\eta}^{(j)}))$, $\widehat{Z}^{(j)} \leftarrow \widehat{\eta}^{(j)} + \widehat{W}^{(j),-1}(y - \widehat{\mu}^{(j)})$;

$\widehat{X}^{(j)} \leftarrow \widehat{W}^{(j),\frac{1}{2}}X$, $\widehat{y}^{(j)} \leftarrow \widehat{W}^{(j),\frac{1}{2}}\widehat{Z}^{(j)}$;

$\widehat{\mathcal{K}}_s^{(j)} \leftarrow \mathcal{K}_s^{(j)}(\widehat{X}^{(j),\top}\widehat{y}^{(j)}; \widehat{X}^{(j),\top}\widehat{X}^{(j)})$;

$$\widehat{\beta}_s^{(j+1)} \leftarrow \arg\min_{\beta \in \widehat{\mathcal{K}}_s^{(j)}} \left\| \widehat{y}^{(j)} - \widehat{X}^{(j)}\beta \right\|_2^2; \tag{1}$$

$\widehat{D}^{(j)} \leftarrow L(\widehat{\beta}_s^{(j+1)}|X, y)/L(\widehat{\beta}_s^{(j)}|X, y)$;

$j \leftarrow j + 1$;

end

We now inspect the IRPLS algorithm in more detail. We assume we can compute or approximate with arbitrary precision the likelihood $L(\cdot|X, y)$ given the data. In

particular, we have access to the auxiliary functions κ' and κ'' which characterize the GLM setting. For continuous Bernoulli variables, these functions are given explicitly in Section 1. The algorithm is initialized at zero. Given any iteration $j \geq 0$, some auxiliary quantities are defined: linear predictor $\widehat{\eta}^{(j)}$, predicted mean $\widehat{\mu}^{(j)}$, weight matrix $\widehat{W}^{(j)}$, pseudo-response $\widehat{Z}^{(j)}$, weighted design and pseudo-response $\widehat{X}^{(j)}$, $\widehat{y}^{(j)}$. Differently from the classical IRLS algorithm by (McCullagh, 1989), we solve the weighted-least-squares problem in Equation (1) via PLS, which means that the coefficients $\widehat{\beta}_s^{(j+1)}$ belong to the s-dimensional Krylov space $\widehat{\mathcal{K}}_s^{(j)}$. Let $\widehat{R}_s^{(j)} \in \mathbb{R}^{p \times s}$ be the orthogonal projection onto the s-dimensional latent space, so there is a unique $\widehat{\alpha}_s^{(j+1)} \in \mathbb{R}^s$ such that $\widehat{\beta}_s^{(j+1)} = \widehat{R}_s^{(j)} \widehat{\alpha}_s^{(j+1)}$. With latent design $\widehat{T}_s^{(j)} := \widehat{X}^{(j)} \widehat{R}_s^{(j)}$, the estimated response is $\widehat{y}_s^{(j)} = \widehat{X}^{(j)} \widehat{\beta}_s^{(j+1)} = \widehat{T}_s^{(j)} \widehat{\alpha}_s^{(j+1)}$. As we can see, the dimensionality reduction is performed at each iteration and the resulting coefficients are then fed to the next cycle. We run the algorithm until the likelihood ratio stops improving (up to a small threshold) or until we reach the maximum number of steps.

We perform model selection by dividing the data into training set (X_{train}, y_{train}) and testing set (X_{test}, y_{test}). After choosing a maximum number of PLS components, say $m_{pls} = 42$, we run different instances of IRPLS$(X_{train}, y_{train}, s)$ with different hyper-parameters over all $1 \leq s \leq m_{pls}$. Given one particular instance of the IRPLS algorithm, let $\widehat{\beta}_s^{(J)}$ and $\widehat{y}_s^{(J)}$ be the output coefficients and estimated response, respectively, for any given $1 \leq s \leq m_{pls}$. The performance of these coefficients is measured in terms of: correlation between responses $Cor(\widehat{y}_s^{(J)}, y_{test})$, root-mean-squared-error between responses $RMSE(\widehat{y}_s^{(J)}, y_{test})$. Among all coefficients $\widehat{\beta}_1^{(J)}, \ldots, \widehat{\beta}_{m_{pls}}^{(J)}$, those yielding the best prediction on the testing set are kept.

All computations in Algorithm 1 are explicit, with the exception of Equation (1). Our implementation in `R` is an adaptation of the `kernelpls.fit` function of the `pls` package, see (Liland et al., 2021). This is a numerically efficient procedure

since the Krylov spaces are never computed explicitly, but only as the result of successive matrix-vector multiplications taking at most $O(np)$ operations each. That is, in order to compute the s-dimensional PLS solution, at most $O(snp)$ operations are required. At convergence, after J maximum steps, our IRPLS algorithm has performed at most $O(Jsnp)$ operations. The above numerical properties are inherited from PLS being closely related to conjugate-gradient methods discussed in (Nemirovskii, 1986; Hanke, 1995), we refer to (Singer et al., 2016) for more details.

References

Hanke, M. (1995). *Conjugate gradient type methods for ill-posed problems*. Chapman and Hall/CRC.

Liland, K. H., Mevik, B.-H., and Wehrens, R. (2021). *pls: Partial Least Squares and Principal Component Regression*. R package version 2.8-0.

McCullagh, P. P. (1989). *Generalized linear models*. Monographs on statistics and applied probability 37. Chapman & Hall/CRC, Boca Raton, second edition. edition.

Nemirovskii, A. S. (1986). The regularizing properties of the adjoint gradient method in ill-posed problems. *USSR Computational Mathematics and Mathematical Physics*, 26(2):7–16.

Singer, M., Krivobokova, T., Munk, A., and de Groot, B. (2016). Partial least squares for dependent data. *Biometrika*, 103(2):351–362.

Wold, H. (1966). Nonlinear estimation by iterative least squares procedure. In David, F. N., editor, *Research papers in statistics: Festschrift for J. Neyman*, pages 411–444. Wiley.