Extensions and Variants of Linear Regression

Xiaogang Su, Ph.D.
Department of Mathematical Sciences
University of Texas at El Paso (UTEP)
xsu@utep.edu

April 2, 2018

Contents

Consider a supervised setting with response Y and predictors $\mathbf{x} = (X_1, \ldots, X_p)$. Without loss of generality, we assume that Y has been centered and each $\mathbf{X}_j$ has been normalized to have mean 0 and sd 1. In the following, we shall list a few closely related algorithms.

1 Computation of Regularization Path

In general, the nonsmooth nature of the lasso constraint makes the solutions nonlinear in $\mathbf{y}$. In the initial proposal of lasso by Tibshirani (1996), quadratic programming was employed to solve the optimization problem by using the fact that the condition $\sum_j |\beta_j| \leq s$ is equivalent to $\boldsymbol{\delta}_i^T \boldsymbol{\beta} \leq s$ for all $i = 1, 2, \ldots, 2^p$, where $\boldsymbol{\delta}_i$ is the p-tuples of form $(\pm 1, \pm 1, \ldots, \pm 1)$. Later, Osborne (2000a, 2000b) developed a compact descent method for solving the constrained lasso problem for any fixed s and a "homotopy" method that completely describe the possible selection regimes in the lasso solution. In the same vein, Efron et al. (2004) derived a parallel variant, called the least angle regression (LARS) . LARS facilitates a variable selection method in its own right. More importantly, the entire path of lasso solutions as s varies from 0 to $+\infty$ can be extracted with a slight modification on LARS.

Motivated by problems with a large p number of predictors, the coordinate descent (CD) method becomes popular later on. It is applicable not only to solving for LASSO regularizaiton path but also to the nonconvex SCAD and MCP regularization problems.

1.1 Properties of the LASSO Solution

Owing to the convex formulation, the constrained LASSO problem

$$\min_{\boldsymbol{\beta}} \ \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 s.t. \|\boldsymbol{\beta}\|_1 < t \quad \text{for } t > 0$$

is equivalent to its penalized Lagrangian form

$$\min_{\boldsymbol{\beta}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 \ + \ \lambda \|\boldsymbol{\beta}\|_1, \tag{1}$$

where the tuning parameter $\lambda > 0$ has a one-to-one correspondence to the upper bound $t > 0$ for the ℓ_1 norm.

Given a fixed $\lambda > 0$, the KarushKuhnTucker (KKT) conditions inform that the LASSO solution $\widetilde{\boldsymbol{\beta}}$ must satisfy

$$\mathbf{X}^T(\mathbf{y} - \mathbf{X}\widetilde{\boldsymbol{\beta}}) \ = \ \lambda \hat{\boldsymbol{\Sigma}}, \tag{2}$$

where the vector $\hat{\boldsymbol{\Sigma}} = (s_j) \in \mathbf{R}^p$ satisfies

$$s_j = \begin{cases} +1 & \text{if } \tilde{\beta}_j > 0 \\ [-1,\ 1] & \text{if } \tilde{\beta}_j = 0 \\ -1 & \text{if } \tilde{\beta}_j < 0 \end{cases} \tag{3}$$

for $j = 1, \ldots p$. Equation (2) is obtained by differentiating the Lagrangian (1). Since $|\beta|$ is not differentiable at 0 in the ℓ_1 penalty, the concept of subderivative or subgradient is introduced.

Definition 1.1. Let $f : \mathbb{R}^k \to \mathbb{R}$ be a convex function. For a point $\mathbf{x}_0$ with $|f(\mathbf{x}_0)| < \infty$, a vector $\mathbf{g} \in \mathbb{R}^k$ is called a subgradient of f at $\mathbf{x}_0$ if

$$f(\mathbf{x}) - f(\mathbf{x}_0) \ \geq \ \mathbf{g}^T(\mathbf{x} - \mathbf{x}_0)$$

for any $\mathbf{x} \in \mathbb{R}^k$. In the univariate scenario with $k = 1$, g is called a *subderivative* of f at x_0. The collection of all subgradients of f at $\mathbf{x}_0$ is called the *subdifferential* of f at $\mathbf{x}_0$, denoted by $\partial f(\mathbf{x}_0)$.

If a convex function f is differentiable at $\mathbf{x}_0$, then the subdifferential $\partial f(\mathbf{x}_0)$ would contain exactly one element, which is the gradient. It can be seen that a point $\mathbf{x}_0$ is a global minimum of a convex function f if and only if zero is contained in the subdifferential $\partial f(\mathbf{x}_0)$. By this definition, it can be easily checked that a subderivative of function $f(x) = |x|$ at $x_0 = 0$ is any number in $[-1, 1]$. Thus, vector $\hat{\boldsymbol{\Sigma}}$ in (3) is essentially the subdifferential of $\|\boldsymbol{\beta}\|_1$ at $\widetilde{\boldsymbol{\beta}}$.

It is routine to compute the entire regularization path $\{\widetilde{\boldsymbol{\beta}}(\lambda) : \lambda > 0\}$ and plot them versus λ in regularization methods. Same as in OLS, the fitted vector $\tilde{\mathbf{y}} = \mathbf{X}\widetilde{\boldsymbol{\beta}}$ in LASSO is always unique, but the coefficient estimator $\widetilde{\boldsymbol{\beta}}$ may not be. Tibshirani (2013) showed that $\widetilde{\boldsymbol{\beta}}$ for a given λ is unique as long as the columns of $\mathbf{X}$ are in general position, as defined below. This above uniqueness property holds even when $p > n$, although the number of nonzero coefficient estimates in $\widetilde{\boldsymbol{\beta}}$ is at most n.

Definition 1.2. The columns of $\mathbf{X}$ are *in general position* if any affine subspace in $\mathbb{R}^n$ of dimension $k < n$ contains at most $k + 1$ elements of the set $\{\pm\mathbf{x}_1, \pm\mathbf{x}_2, \ldots, \mathbf{x}_p\}$, excluding antipodal pairs of points (i.e., points differing only with opposite signs).

1.2 The Homotopy Algorithm

To compute the entire regularization path, let λ decrease from ∞ where the corresponding $\widetilde{\boldsymbol{\beta}}$ is 0. At any stage, let $\mathcal{A} \subset \{1, \ldots, p\}$ denote the active set such that $\tilde{\beta}_j \neq 0$ for $j \in \mathcal{A}$ and $\tilde{\beta}_j = 0$ for $j \notin \mathcal{A}$. The fitted vector is

$$\tilde{\mathbf{y}} \;=\; \mathbf{X}\widetilde{\boldsymbol{\beta}} \;=\; \mathbf{X}_{\mathcal{A}}\widetilde{\boldsymbol{\beta}}_{\mathcal{A}} \;=\; \tilde{\mathbf{y}}_{\mathcal{A}}(\lambda).$$

According to the KKT condition (2), The LASSO solution $\widetilde{\boldsymbol{\beta}}(\lambda)$ for a given $\lambda > 0$ must satisfy

$$\begin{cases} \mathbf{X}_{\mathcal{A}}^T\left(\mathbf{y} - \mathbf{X}_{\mathcal{A}}\widetilde{\boldsymbol{\beta}}_{\mathcal{A}}\right) &= \lambda\hat{\boldsymbol{\Sigma}}_{\mathcal{A}} \quad \text{with } \hat{\boldsymbol{\Sigma}}_{\mathcal{A}} = (\pm 1) \\ \left|\mathbf{x}_j^T\left(\mathbf{y} - \mathbf{X}_{\mathcal{A}}\widetilde{\boldsymbol{\beta}}_{\mathcal{A}}\right)\right| &< \lambda, \qquad \forall j \notin \mathcal{A}. \end{cases} \tag{4}$$

This leads to solution

$$\begin{cases} \widetilde{\boldsymbol{\beta}}_{\mathcal{A}}(\lambda) &= (\mathbf{X}_{\mathcal{A}}^T\mathbf{X}_{\mathcal{A}})^{-1}(\mathbf{X}_{\mathcal{A}}^T\mathbf{y} - \lambda\,\hat{\boldsymbol{\Sigma}}_{\mathcal{A}}) \\ \tilde{\beta}_j(\lambda) &= 0, \quad \forall j \notin \mathcal{A}. \end{cases} \tag{5}$$

From the first equation of (5), the LASSO solution $\widetilde{\boldsymbol{\beta}}(\lambda)$ is piecewise linear in λ as λ decreases from $+\infty$ to 0.

1.3 Coordinate Descent (CD)

The coordinate descent (CD) algorithm iteratively optimizes the objective function with respect to one (or a few) decision variables at a time. It is particularly attractive for high-dimensional problems that have a simple closed form univariate solution but lack one in higher dimensions.

Consider a regularization problem in the following general form

$$\min_{\boldsymbol{\beta}} \; \frac{1}{2n} \| \mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \sum_{j=1}^{p} \rho_{\boldsymbol{\lambda}}(|\beta_j|), \tag{6}$$

where the penalty function $\rho_{\boldsymbol{\lambda}}(\cdot)$ has a (low-dimensional) tuning parameter vector $\boldsymbol{\lambda}$; the factor $1/2n$, which does not affect the solution, provides some kind of standardization so that the tuning

parameters become comparable for different sample sizes, which can be useful, e.g., for cross-validation.

In LASSO, $\boldsymbol{\lambda} > 0 \in \mathbb{R}$ is a scalar and $\rho_\lambda(|\beta_j|) = \lambda|\beta_j|$. To find the univariate solution for β_j, fixing all other $\boldsymbol{\beta}_{(-j)}$'s at their current estimate. Rewrite

$$\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\| \;=\; \|\mathbf{y} - \mathbf{X}_{(-j)}\boldsymbol{\beta}_{(-j)} - \mathbf{x}_j\beta_j\| \;=\; \|\mathbf{r}_j - \mathbf{x}_j\beta_j\|,$$

where $\mathbf{r}_j = \mathbf{y} - \mathbf{X}_{(-j)}\boldsymbol{\beta}_{(-j)}$ are some kind of partial residuals. Rewrite

$$\mathbf{r}_j = \mathbf{y} - \mathbf{X}\boldsymbol{\beta} + \beta_j\mathbf{x}_j = \mathbf{r} + \beta_j\mathbf{x}_j$$

with $\mathbf{r} = \mathbf{y} - \mathbf{X}\boldsymbol{\beta}$ being the ordinary residual. A direct inspection of the objective (quadratic) function in (6) leads to the univariate solution of β_j:

$$\tilde{\beta}_j \;=\; \mathcal{S}_\lambda(\mathbf{r}_j^T\mathbf{x}_j/n) \;=\; \mathcal{S}(b_j;\lambda) \;\; \text{with } b_j = \mathbf{r}_j^T\mathbf{x}_j/n, \tag{7}$$

where $\mathcal{S}(\cdot;\lambda)$ is the soft threshold operator

$$\mathcal{S}(x;\lambda) = \text{sgn}(x)\,(|x| - \lambda)_+ = \begin{cases} x - \lambda & \text{if } x > \lambda \\ 0 & \text{if } |x| \le \lambda \\ x + \lambda & \text{if } x < -\lambda. \end{cases} \tag{8}$$

For SCAD, two tuning parameters $\boldsymbol{\lambda} = (a, b)^T \in \mathbb{R}^2$ are involved in the penalty function. The univariate solution is given by

$$\tilde{\beta}_j \;=\; \begin{cases} \mathcal{S}(b_j;\, a) & \text{if } |b_j| \le 2a \\ \dfrac{\mathcal{S}\left(b_j;\, ab/(b-1)\right)}{1 - 1/(b-1)} & \text{if } 2a < |b_j| \le ab \\ b_j & \text{if } |b_j| > ab \end{cases} \tag{9}$$

The MCP penalty also has two parameters $\boldsymbol{\lambda} = (a, b)^T \in \mathbb{R}^2$. with univariate solution given by

$$\tilde{\beta}_j \;=\; \begin{cases} \dfrac{\mathcal{S}(b_j;\, a)}{1 - 1/b} & \text{if } |b_j| \le ab \\ b_j & \text{if } |b_j| > ab \end{cases} \tag{10}$$

In a CD algorithm, the univariate solution is quickly computed as the coordinate-wise minimizer of the objective function. CD iterates over $j = 1, \ldots, p$ for a total of M iterations. At the j-th step of the m-th iteration, CD (Breheny and Huang, 2011) proceeds with the following operations:

(i). Compute $b_j = \mathbf{r}_j^T\mathbf{x}_j/n = \mathbf{r}^T\mathbf{x}_j/n + \beta_j^{(m)}$;

(ii). Update $\beta_j^{(m+1)}$ with the univariate solution formula in (7), (9), or (10);

(iii). Update $\mathbf{r} \;:=\; \mathbf{r} - (\beta_j^{(m+1)} - \beta_j^{(m)})\mathbf{x}_j$.

The last step ensures that the residual vector $\mathbf{r}$ is computed with all the latest estimates. The CD approach can be used to compute the entire regularization path for LASSO, SCAD, and MCP. For LASSO, its piecewise linear property of its path can not be well utilized in CD. Instead, one picks a number of λ values and computes the solution at each value. To speed up, the solution from the neighboring λ value is used as the starting value for compute the next solution, referred to as the 'warm start'.

2 Ridge Regression and PCR

The singular value decomposition (SVD) of the design matrix $\mathbf{X}$ can provide further insights into the nature of ridge regression. We shall establish, following HTF (2009), a connection among OLS regression, ridge regression, and principal component regression (PCR) analysis.

2.1 Perspective with SVD of X

Assume the variables are standardized or centered so that the matrix

$$\hat{\mathbf{\Sigma}} = \mathbf{X^T X}/n \tag{11}$$

gives either the sample variance-covariance matrix or the sample correlation matrix among predictors X_j's. The SVD of the $n \times p$ design matrix $\mathbf{X}$ has the form

$$\mathbf{X} = \mathbf{UDV^T}, \tag{12}$$

where $\mathbf{U}$ and $\mathbf{V}$ are $n \times p$ and $p \times p$ orthogonal matrices such that the columns of $\mathbf{U}$ form an orthonormal basis of the column space of $\mathbf{X}$ and the columns of $\mathbf{V}$ form an orthonormal basis of the row space of $\mathbf{X}$; the $p \times p$ diagonal matrix $\mathbf{D} = \text{diag}(d_j)$ with diagonal entries d_j, $d_1 \geq d_2 \geq \cdots \geq d_p \geq 0$ being the singular values of $\mathbf{X}$. It follows that

$$\mathbf{X^T X} = \mathbf{VD^2V^T}, \tag{13}$$

which provides the spectral decomposition of $\mathbf{X^T X}$. Thus the eigenvalues of $\mathbf{X^X}$ are $\lambda_j = d_j^2$. The spectral decomposition for $\hat{\mathbf{\Sigma}}$ in (11) would be

$$\hat{\mathbf{\Sigma}} = (1/n) \cdot \mathbf{VD^2V^T}, \tag{14}$$

with eigenvalues d_j^2/n and eigenvectors $\mathbf{v}_j$ (i.e., the j-th column of $\mathbf{V}$).

Using the SVD of $\mathbf{X}$ in (12), the least squares fitted vector $\widehat{\mathbf{y}}^{\text{LS}}$ can be rewritten as

$$\begin{aligned}
\widehat{\mathbf{y}}^{\text{LS}} = \mathbf{X}\widehat{\boldsymbol{\beta}}^{\text{LS}} &= \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} \\
&= \mathbf{UDV}^T(\mathbf{VD}^T\mathbf{U}^T\mathbf{UDV}^T)^{-1}\mathbf{VD}^T\mathbf{U}^T\mathbf{y} \\
&= \mathbf{UDV}^T(\mathbf{VD}^{-2}\mathbf{V}^T)^{-1}\mathbf{VD}^T\mathbf{U}^T\mathbf{y} \\
&= \mathbf{UDV}^T\mathbf{VD}^2\mathbf{V}^T\mathbf{VD}^T\mathbf{U}^T\mathbf{y} \\
&= \mathbf{UU}^T\mathbf{y} \\
&= \sum_{j=1}^{p}(\mathbf{u}_j^T\mathbf{y}) \cdot \mathbf{u}_j
\end{aligned} \tag{15}$$

Note that $\mathbf{U}^T\mathbf{y}$ or $\mathbf{u}_j^T\mathbf{y}$'s are the coordinates of $\mathbf{y}$ with respect to the columns of $\mathbf{U}$. Recall that $\mathbf{u_j}$'s are the orthonormal basis spanning the column space of $\mathbf{X}$ and also the normalized sample principal components.

Another method to deal with multicollinearity is called *principal components regression* (PCR). In this approach, y is regressed on the first m principal components by rejecting the last $(p - m)$

components that explain a relatively small portion of variation in $\mathbf{X}$. Thus the fitted response vector would be

$$\widehat{\mathbf{y}}^{\text{PCR}} = \sum_{j=1}^{m} (\mathbf{u}_j^T \mathbf{y}) \cdot \mathbf{u}_j. \tag{16}$$

The implicit assumption is that the response tends to vary most in the directions where the predictors have large variations.

The fitted vector based on ridge regression, after similar simplification, is

$$\begin{aligned} \widehat{\mathbf{y}}^{\text{R}} = \mathbf{X}\widehat{\boldsymbol{\beta}}^{\text{R}} &= \mathbf{X}(\mathbf{X}^T\mathbf{X} + k\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y} \\ &= \mathbf{U}\mathbf{D}(\mathbf{D}^2 + k\mathbf{I})^{-1}\mathbf{D}\mathbf{U}^T\mathbf{y} \\ &= \sum_{j=1}^{p} \frac{d_j^2}{d_j^2 + k} (\mathbf{u}_j^T \mathbf{y}) \cdot \mathbf{u}_j. \end{aligned} \tag{17}$$

Thus, similar to least squares regression, the ridge solution computes the coordinates of $\mathbf{y}$ with respect to the orthonormal basis $\mathbf{U}$ and then shrinks them by the factor $d_j^2/(d_j^2 + k)$. Note that $d_j^2/(d_j^2 + k) \leq 1$ as $k \geq 0$. With this strategy, a greater amount of shrinkage is applied to basis vectors or principal component vectors corresponding to smaller $d_j^2 = \lambda_j$. Instead of rejecting low-variance directions as in PCR, ridge regression keeps all principal component directions but weighs the coefficients by shrinking low-variance directions more.

2.2 Weighted Orthogonal Components Regression (WOCR)

WOCR (Su, Wonkye, Wang, and Yin, 2017+) generalizes on basis of OLS regression, RR, and PCR. Suppose that columns of matrix $\mathbf{U} = (\mathbf{u}_1, \ldots, \mathbf{u}_p)$ form an orthonormal basis of the column space of $\mathbf{X}$. Note that PCA is just one way of extracting oothonormal basis $\{\mathbf{u}_j\}_{j=1}^p$; the Gram-Schimt process, partial least squares components, and continuum regression are among the alternative ways. Somehow, it matters whether $\mathbf{y}$ plays a role in finding $\mathbf{u}_j$. This would affect the model complexity as measured by the degrees of freedom. For simplicity, let's assume that extraction of the orthogonal components $\mathbf{u}_j$ has nothing to do with $\mathbf{y}$ for the time being.

WOCR specifies the fitted vector in a general form as follows

$$\widehat{\mathbf{y}} = \sum_{j=1}^{p} w_j \gamma_j \mathbf{u}_j, \tag{18}$$

where $\gamma_j = \langle \mathbf{u}_j, \mathbf{y} \rangle$ is the coefficient of $\mathbf{u}_j$ and $\mathbf{w} = (w_j)$ are the weight vector. OLS regression, RR, and PCR are special cases of WOCR models with $w_j = 1$, $d_j^2/(d_j^2 + \lambda)$, and $I(d_j^2 \geq t)$ for some $t > 0$, respectively.

The next important component in specifying WOCR is to parameterize the weights in $\mathbf{w}$ in a principled way. The key motivation stems from the observation that, compared to the original regressors in $\mathbf{X}$, the orthogonal components in $\mathbf{U}$ are naturally ordered in some way. One ordering can be attributable to the specific variation d_j^2 explained by each PC component $d_j \mathbf{u}_j$. This ordering is utilized in RR and PCR. Another natural ordering is based on the coefficients $\{|\gamma_j|\}_{j=1}^m$.

It is intuitive to assign more weights to more important components. To do so, w_j can be specified as a monotone function in the ordering measure and parameterized with a low-dimensional

vector $\boldsymbol{\lambda}$. Among many other choices, the usage of sigmoid functions will be advocated in this article because they provide a smooth approximation to the 0-1 threshold indicator function that is useful for the component selection purpose and they are also flexible enough to adjust for achieving improved prediction accuracy. In general, we denote $w_j = w_j(\boldsymbol{\lambda})$. The vector $\boldsymbol{\lambda}$ in the weight function are essentially the tuning parameters.

While there are arguments saying that the response tends to be more correlated with the leading principal components, this is usually not the case in many real-life data. WOCR can provide a convenient solution to deal with this issue: one simply bases the ordering of $\mathbf{u}_j$ on the regression coefficients γ_j and defines the weights w_j via a monotone function of $|\gamma_j|$ or, preferably, γ_j^2. However, doing so will induce dependence on the response to the weights. As a result, the associated DF has to be computed differently, as established in Proposition 1.

Proposition 1. *Suppose that the WOCR model (18) has orthogonal components* $\mathbf{u}_j$ *constructed independently of* $\mathbf{y}$ *and weights* $w_j = w(\gamma_j^2; \boldsymbol{\lambda})$, *where* $w(\cdot)$ *is a smooth monotonically increasing function and* $\boldsymbol{\lambda}$ *is the parameter vector. Its degrees of freedom (DF) can be estimated as*

$$\widehat{DF} = \sum_{j=1}^{m} (2\gamma_j^2 \dot{w}_j + w_j), \tag{19}$$

where $\dot{w}_j = dw(\gamma_j^2; \boldsymbol{\lambda})/d(\gamma_j^2)$.

Clearly both PCR and RR can be benefited from this reformulation. As a variant of RR, the weight now becomes $w_j = w(\gamma_j^2; \lambda) = \gamma_j^2/(\gamma_j^2 + \lambda)$ and hence $\dot{w}_j = \lambda/(\gamma_j^2 + \lambda)^2$. It follows that the estimated DF is

$$\widehat{\mathrm{DF}} = \sum_{j=1}^{m} (\gamma_j^4 + 3\lambda\gamma_j^2)/(\gamma_j^2 + \lambda)^2.$$

The best tuning parameter $\hat{\lambda}$ can be obtained by minimizing GCV. Using similar notations as earlier, we denote this RR variant as $\mathrm{RR}(\gamma; \lambda)$. It is worth noting that $\mathrm{RR}(\gamma; \lambda)$ is, in fact, not a ridge regression model. Its solution can no longer be nicely motivated by a regularized or constrained least square optimization problem as in the original RR. But what really matters in these methods is the predictive power. By directly formulating the fitted values $\hat{\mathbf{y}}$, the WOCR model (18) facilitates a direct and flexible model specification that focuses on prediction.

For PCR, the weight becomes $w_j = \pi(\gamma_j^2; a, c)$. Hence, $\dot{w}_j = a w_j(1 - w_j)$ and

$$\widehat{\mathrm{DF}} = \sum_{j=1}^{m} w_j(2ar_j^2 + 1 - 2aw_j r_j^2).$$

Depending on whether or not we want to select components, we may fix a at a larger value or leave it free. This results in two PCR variants, which we denote as $\mathrm{PCR}(\gamma_j^2; c)$ and $\mathrm{PCR}(\gamma_j^2; a, c)$, respectively.

Table 1 summarizes the WOCR models that we have discussed so far. Among them, $\mathrm{RR}(d_j; \lambda)$ and $\mathrm{PCR}(d_j^2; c)$ resemble the conventional RR and PCR.

One additional advantage of WOCR is its pre-tuning feature. With the formulation, all the involved tuning parameters in WOCR can be directly estimated by minimizing a model selection criterion beforehand. On the basis of available SSE and EDF, the model selection criterion now

Table 1: WOCR Variants of ridge regression (RR) and principal components regression (PCR) models, both based on the normalized principal components $\{\mathbf{u}_j : j = 1, \ldots p\}$.

Model	Component Ordering	Weights	Tuning Parameter	Suggested WOCR Objective Function
RR$(d;\lambda)$	d_j	$w_j = d_j^2/(d_j^2+\lambda)$	λ	GCV(λ)
RR$(\gamma;\lambda)$	γ_j^2	$w_j = \gamma_j^2/(\gamma_j^2+\lambda)$	λ	GCV(λ)
PCR$(d;c)$	d_j	$w_j = \text{expit}\{a(d_j - c)\}$ with fixed a	c	BIC(c)
PCR$(d;a,c)$	d_j	$w_j = \text{expit}\{a(d_j - c)\}$	a, c	GCV(a,c)
PCR$(\gamma;c)$	γ_j^2	$w_j = \text{expit}\{a(\gamma_j^2 - c)\}$ with fixed a	c	BIC(c)
PCR$(\gamma;a,c)$	γ_j^2	$w_j = \text{expit}\{a(\gamma_j^2 - c)\}$	a, c	GCV(a,c)

becomes an objective function of the tuning parameters. For example, the GCV for Model RR$(d;\lambda)$ is given, up to some irrelevant constant, by

$$\text{GCV}(\lambda) \;\propto\; \frac{SSE}{(n-EDF)^2} = \frac{\| \mathbf{y} \|^2 - \sum_{j=1}^m (w_j^2 - 2w_j)\,\gamma_j^2}{(n - \sum_{j=1}^m w_j)^2}, \tag{20}$$

The best tuning parameter $\hat{\lambda}$ can then be estimated as

$$\hat{\lambda} = \underset{\lambda}{\text{argmin}}\, \text{GCV}(\lambda).$$

Depending on the analytic purpose, a criterion is recommended for each WOCR model in Table 1. In general, GCV is suggested for predictive purposes, in which scenarios AIC can be used as an alternative. AIC is equivalent to GCV if $\lim_{n\to\infty} p/n = 0$, both being selection-efficient in the sense prescribed by **?**. On the other hand, if selecting components is desired, using BIC is recommended.

Once $\hat{\lambda}$ is available, plug it into the WOCR model (18) and rewrite the fitted vector in the form of $X\widehat{\boldsymbol{\beta}}$. Then the response for a new observation with $\mathbf{x}_0$ can be conveniently predicted as $\hat{y}_0 = \mathbf{x}_0^T \widehat{\boldsymbol{\beta}}$.

3 The LAR Algorithm

In this section, the homotopy (Osborne, Presnell, and Turlach, 2000) or LARS (Efron et al., 2004) algorithm for solving LASSO is first introduced with greater details. Then some properties of the LASSO solution are outlined.

We start with an (incremental) forward stagewise regression method.

Algorithm I: (Incremental) Forward Stagewise Regression

1. Start with residual $\mathbf{r} = \mathbf{y}$ and $\beta_1, \ldots, \beta_p = 0$.
2. Find the predictor x_j most correlated with $\mathbf{r}$.

3. Update $\beta_j \longleftarrow \beta_j + \delta_j$, where $\delta_j = \varepsilon \cdot \text{sign}\langle \mathbf{r}, \mathbf{x}_j \rangle$ and $\varepsilon > 0$ is a damping factor.

4. Set $\mathbf{r} \longleftarrow \mathbf{r} - \delta_j \cdot \mathbf{x}_j$ and repeat steps 2 and 3 many times.

Note that, if $\delta_j = \langle \mathbf{r}, \mathbf{x}_j \rangle$ is used instead, this gives usual forward stagewise regression, which is different from forward stepwise.

As manifested below in Algorithm II, if we replace predictors by some base learners such as regression trees, the incremental forward stagewise becomes analogous to least squares boosting.

Algorithm II: Least Squares Boosting

1. Start with function $F(\mathbf{x}) = 0$ and residual $\mathbf{r} = \mathbf{y}$.

2. Fit a CART regression tree to $\mathbf{r}$ giving $f(\mathbf{x})$

3. Set $F(\mathbf{x}) \longleftarrow F(\mathbf{x}) + \varepsilon f(\mathbf{x})$ and update $\mathbf{r} := \mathbf{y} - F(\mathbf{x})$;

4. Repeat step 2 & 3 many times.

The least angle regression (LAR) algorithm is motivated by the forward stagewise regression and offers a computational shortcut to stagewise regression. After a slight modification, the entire solution path of LASSO can be obtained as well.

The LARS method iteratively builds up the fitted response vector $\widehat{\boldsymbol{\mu}}$ with updating steps, analogous to boosting and stagewise regression. The main steps of LARS are first briefly outlined, with some details following up. Initially all coefficients are set to zero. The predictor that has highest correlation with the current residual, which is the response itself in this stage, is identified. A step is then taken in the direction of this predictor. The length of this step, which corresponds to the coefficient for this predictor, is chosen such that some other predictor (i.e., the second predictor entering the model) and the current predicted response have the same correlation with the current residual. Next, the predicted response moves in the direction that is equiangular between or equally correlated with these two predictors. Moving in this joint direction ensures that these two predictors continue to have a common correlation with the current residual. The predicted response moves in this direction until a third predictor has the same correlation with the current residual as the two predictors already in the model. A new joint direction that is equiangular between these three predictors is determined and the predicted response moves in this direction until a fourth predictor having the same correlation with the current residual joins the set. This process continues till all predictors have entered the model.

More specifically, start with $\widehat{\boldsymbol{\mu}}_0 = \mathbf{0}$. Let $\widehat{\boldsymbol{\mu}}_{\mathcal{A}}$ denote the current LARS estimate for the predicted vector, where $\mathcal{A}$ is the active set of indices corresponding to predictors that have the great absolute correlations with the current residuals, i.e., predictors in the current model. The sample correlation between the residuals and values of each predictor indexed in $\mathcal{A}$ is given as

$$\mathbf{c} = \mathbf{X}^T(\mathbf{y} - \widehat{\mathbf{y}}_{\mathcal{A}})$$

owing to normalization. Thus,

$$\mathcal{A} = \{j : \ |c_j| = C\} \ \text{ with } C = \max_j(|c_j|).$$

Let $s_j = \text{sign}(c_j)$ for $j \in \mathcal{A}$ be the sign of the correlation between X_j in the active set and the current residuals and let $\mathbf{X}_\mathcal{A} = (s_j \mathbf{x}_j)$, for $j \in \mathcal{A}$, be the design matrix containing all signed active predictors. Compute matrices

$$\begin{aligned} \mathbf{G}_\mathcal{A} &= \mathbf{X}_\mathcal{A}^T \mathbf{X}_\mathcal{A}, & (21) \\ A_\mathcal{A} &= (\mathbf{j}_\mathcal{A}^T \mathbf{G}_\mathcal{A}^{-1} \mathbf{j}_\mathcal{A})^{-1/2}, & (22) \\ \mathbf{w}_\mathcal{A} &= A_\mathcal{A} \mathbf{G}_\mathcal{A}^{-1} \mathbf{j}_\mathcal{A}, & (23) \end{aligned}$$

where $\mathbf{j}_\mathcal{A}$ is the vector of 1's of dimension equal to $|\mathcal{A}|$, the cardinality of $\mathcal{A}$. Then the equiangular vector

$$\mathbf{u}_\mathcal{A} = \mathbf{X}_\mathcal{A} \mathbf{w}_\mathcal{A} \ \text{ with } \ \| \mathbf{u}_\mathcal{A} \| = 1 \tag{24}$$

makes equal angles, less than 90°, with each column of $\mathbf{X}_\mathcal{A}$, i.e.,

$$\mathbf{X}_\mathcal{A}^T \mathbf{u}_\mathcal{A} = A_\mathcal{A} \mathbf{j}_\mathcal{A}. \tag{25}$$

Also compute the correlations between each predictor with the equiangular vector $\mathbf{u}_\mathcal{A}$, given as

$$\mathbf{a} = \mathbf{X}^T \mathbf{u}_\mathcal{A}.$$

Then, the next updating step of the LARS algorithm is

$$\widehat{\boldsymbol{\mu}}_{\mathcal{A}+} = \widehat{\boldsymbol{\mu}}_\mathcal{A} + \hat{\gamma} \mathbf{u}_\mathcal{A}, \tag{26}$$

where

$$\hat{\gamma} = \min_{j \in \mathcal{A}^c}^+ \left\{ \frac{C - c_j}{A_\mathcal{A} - a_j}, \ \frac{C + c_j}{A_\mathcal{A} + a_j} \right\} \tag{27}$$

and the $\min^+$ means that the minimum is taken over only these positive components. Let $\hat{j}$ be the minimizing index in (27). Then $X_{\hat{j}}$ is the variable added to the active set and the new maximum absolute correlation becomes $C - \hat{\gamma} A_\mathcal{A}$.

Two notable remarks are in order. First, LARS is rather thrifty in computation, simply requiring a total of p steps. Secondly, surprisingly, with a slight modification of LARS, one can obtain a sequence of lasso estimates, from which all other lasso solutions can be obtained by linear interpolation. The modification is that, if a non-zero coefficient turns into zero, it will be removed from the active set of predictors and the joint direction will be recomputed. The entire regularization path of LASSO is piece-wise linear. Figure 1 plots the solution paths of LASSO vs. forward stagewise. We can see how similar they are.

There are seemingly two motivations for the LAR procedure. The first is the similarity of forward stagewise and LASSO; the second is to provide a statistical interpretation of the homotopy method of Osborne, Presnell, and Turlach (2000), who first found the efficient optimization algorithm for obtaining the entire LASSO path. The LAR algorithm is presented below in Algorithm III.

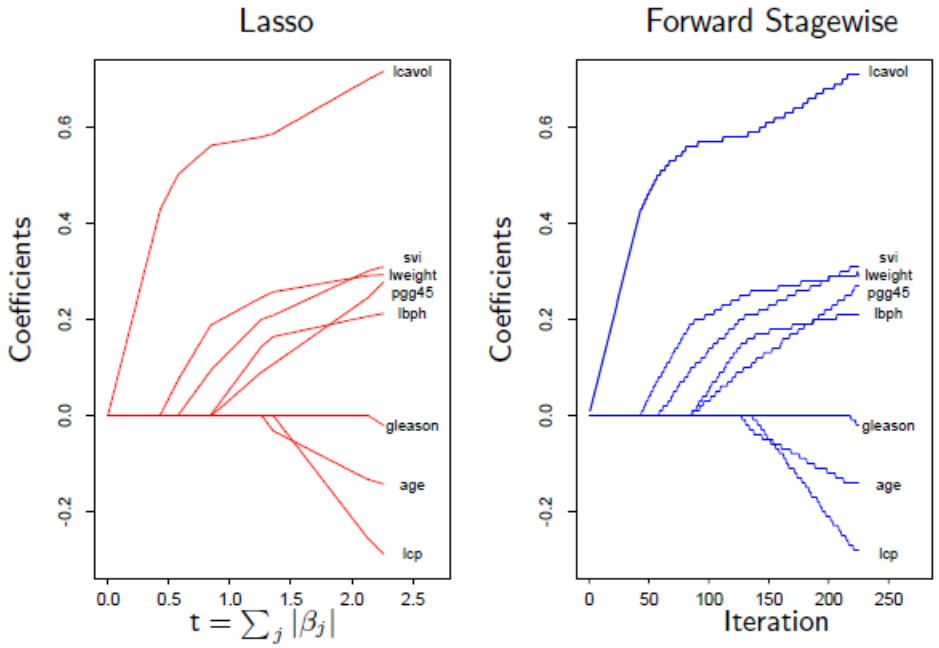

Figure 1: Comparing LASSO vs. Forward Stagewise on a Prostate Cancer Data. The figure is taken from Prof. Hastie's presentation.

Algorithm III: Least Angle Regression (LAR)

1. Start with function residual $\mathbf{r} = \mathbf{y}$ and $\beta_1, \ldots, \beta_p = 0$.
2. Find predictor $\mathbf{x}_j$ most correlated with $\mathbf{r}$.
3. Increase β_j from 0 towards its least-squares coefficient $\langle \mathbf{x}_j, \mathbf{r} \rangle$, until some other competitor $\mathbf{x}_k$ has as much correlation with the current residual as does $\mathbf{x}_j$.
4. Move (β_j, β_k) in the direction defined by their joint least squares coefficient of the current residual on $(\mathbf{x}_j, \mathbf{x}_k)$, until some other competitor $\mathbf{x}_l$ has as much correlation with the current residual.
5. Continue in this way until all p predictors have been entered. After p steps, we arrive at the full least-squares solution, in which case $\text{corr}(\mathbf{r}, \mathbf{x}_j) = 0, \forall\, j$.

An illustration of LAR when $p = 2$ can be given in Figure 2. Note that the current correlations are

$$\mathbf{c}(\hat{\boldsymbol{\mu}}) = \mathbf{X}^T(\mathbf{y} - \hat{\boldsymbol{\mu}}) = \mathbf{X}^T(\hat{\mathbf{y}} - \hat{\boldsymbol{\mu}}),$$

where $\hat{\mathbf{y}} = \mathrm{P}_{\mathbb{V}}\mathbf{y}$ with $\mathbb{V} = \mathcal{C}(\mathbf{X})$ is the full least squares solution. In the figure, $\hat{\mathbf{y}} = \bar{\mathbf{y}}$. This means

that there is no need of seeing the original response vector $\mathbf{y}$ in the illustration. In addition, $\mathbf{u}_2$ is the unit vector bisecting the angle between $\mathbf{x}_i$ and $\mathbf{x}_2$.

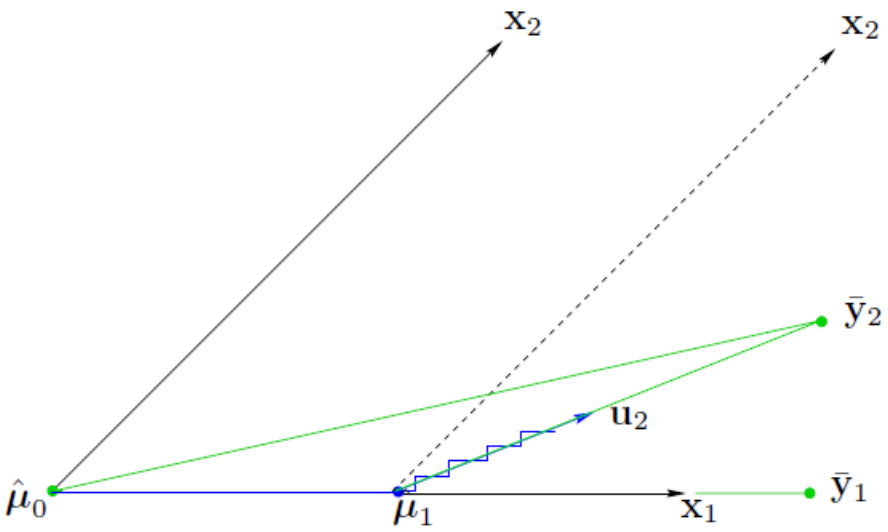

Figure 2: Illustration of LAR when $p = 2$. The figure is taken from Efron, Hastie, Johnstone, and Tibshirani (*Annals of Statistics*, 2004).

To obtain the LASSO solution path, the only modification needed for LAR is that, if a coefficient crosses zero, drop that predictor (from the active predictor set), recompute the best direction and continue.

4 Partial Least Squares

Partial least squares has been popular in chemometrics where the number of predictors, p, is relatively high compared with the number of observations, n; the predictors are correlated, i.e., the multi-collinearity scenario. It is also useful when there is more than one response variable and they are correlated. But here we consider scenarios that involves one single response.

In its general form PLS creates orthogonal score vectors (also called latent vectors or components) by maximising the *covariance* between different sets of variables. PLS is similar to Canonical Correlation Analysis (CCA) where latent vectors with *maximal correlation* are extracted. PLS regression then regresses Y on the first few PLS components as done in PCR.

PLS constructs a set of linear combinations of the inputs for regression, but unlike principal components regression it uses both $\mathbf{y}$ and $\mathbf{X}$ for this construction. Again, we assume that $\mathbf{y}$ has been centered and each $\mathbf{x}_j$ has been standardized to have mean 0 and SD 1. The optimization problem involved for the first direction $\mathbf{a}_1$ in PLS is given below, together with that involved in

canonical correlation analysis (CCA) and PCA:

$$\begin{cases} \text{PCA:} & \max_{\|\mathbf{a}\|=1} \text{var}(\mathbf{Xa}); \\ \text{CCA:} & \max_{\|\mathbf{a}\|=\|\mathbf{b}\|=1} \{\text{corr}\,(\mathbf{Xa}, \mathbf{Yb})\}^2; \\ \text{PLS:} & \max_{\|\mathbf{a}\|=\|\mathbf{b}\|=1} \{\text{cov}\,(\mathbf{Xa}, \mathbf{Yb})\}^2 \end{cases}$$

for multivariate response data $\mathbf{Y} \in \mathbb{R}^{n\times q}$. Since both correlations and covariances can be positive or negative, it is necessary to use the squared quantities. When $q = 1$ (i.e., one single response variable), $\| \mathbf{b} \| = 1$ yields $b = 1$ and hence $\mathbf{Yb} \equiv \mathbf{y}$, in which case

$$\text{cov}\,(\mathbf{Xa}, \mathbf{y}) = \text{var}(\mathbf{Xa}) \{\text{corr}\,(\mathbf{Xa}, \mathbf{y})\}^2.$$

Thus PLS seeks directions that take both variation in $\mathbf{X}$ and correlation with $\mathbf{y}$ into consideration.

Concerning the constrained optimization problem, i.e., $\max_{\mathbf{a}}\{\text{cov}(\mathbf{Xa}, \mathbf{y})\}^2$, note that $\text{cov}(\mathbf{Xa}, \mathbf{y}) = \mathbf{a}^T\mathbf{X}^T\mathbf{y}$. Consider the Lagrangian

$$L \;=\; \mathbf{a}^T\mathbf{X}^T\mathbf{y}\mathbf{y}^T\mathbf{X}\mathbf{a} - \lambda(\mathbf{a}^T\mathbf{a} - 1).$$

Setting $\partial L/\partial \mathbf{a} = \mathbf{0}$ leads to

$$\mathbf{X}^T\mathbf{y}\mathbf{y}^T\mathbf{X}\mathbf{a} = \lambda\mathbf{a},$$

which implies that the solution $\mathbf{a}$ would be the first eigenvector with eigenvalue λ of matrix $\mathbf{X}^T\mathbf{y}\mathbf{y}^T\mathbf{X}$. But matrix

$$\mathbf{X}^T\mathbf{y}\mathbf{y}^T\mathbf{X} = \left(\mathbf{X}^T\mathbf{y}\right)\left(\mathbf{X}^T\mathbf{y}\right)^T$$

is of rank-one. Its first eigenvector corresponding to the only positive eigenvalue (all others being 0) is $\mathbf{X}^T\mathbf{y}$. Therefore, the solution is simply given by

$$\mathbf{a}_1 := \mathbf{X}^T\mathbf{y} \quad \text{and} \quad \mathbf{a}_1 := \mathbf{a}_1/\,\|\,\mathbf{a}_1\,\|$$

with normalization so that $\| \mathbf{a}_1 \| = 1$. Keep in mind that we are ultimately seeking a regression of $\mathbf{y}$ on $\mathbf{X}$ via several of its linear combinations. Here we regress $\mathbf{y}$ on the first linear PLS component

$$\mathbf{z}_1 = \mathbf{X}\mathbf{a}_1 = \sum_{j=1}^{p} \langle \mathbf{x}_j, \mathbf{y}\rangle \mathbf{x}_j.$$

The resultant predicted value becomes $\hat{\mathbf{y}}^{(1)} = P_{C(\mathbf{z}_1)}\mathbf{y}$. For regression purpose, the normalization step $\| \mathbf{a}_1 \| = 1$ really does not matter.

Next, we remove from $\mathbf{X}$ the information explained by the linear combination $\mathbf{z}_1 = \mathbf{X}\mathbf{a}_1$. This is done by subtracting from $\mathbf{X}$ its projection on $\mathbf{z}_1$, i.e., updating $\mathbf{X}$ via

$$\mathbf{X} \;:=\; \mathbf{X} - P_{C(\mathbf{z}_1)}\mathbf{X} \;=\; P_{C^\perp(\mathbf{z}_1)}\mathbf{X}, \tag{28}$$

where, as before, $P_{C(\mathbf{z}_1)}$ is the projection matrix on the column space of $\mathbf{z}_1$:

$$P_{C(\mathbf{z}_1)} = \frac{\mathbf{z}_1\mathbf{z}_1^T}{\| \mathbf{z}_1 \|^2} = \frac{\mathbf{X}\mathbf{a}_1\mathbf{a}_1^T\mathbf{X}^T}{\| \mathbf{z}_1 \|^2}. \tag{29}$$

Noting the matrix $\mathbf{X}$ on the left hand side of the numerator in (29), the deflated matrix $\mathbf{X}$ in (28) can be written as $\mathbf{XA}$ for some matrix $\mathbf{A}$. As a result, each $\mathbf{z}_j$ must be linear in the original $\mathbf{X}$. So is the fitted vector $\hat{\mathbf{y}}^{(m)} = \mathbf{X}\hat{\boldsymbol{\beta}}_{PLS}^{(m)}$ at the m-th step. These linear coefficients $\hat{\boldsymbol{\beta}}_{PLS}^{(m)}$ can be recovered from the sequence of PLS transformations, which are useful for making prediction. For example, given a new observation with feature vector $\mathbf{x}_0$, its predicted value can be simply written as $\hat{y}_0 = \mathbf{x}_0^T\hat{\boldsymbol{\beta}}_{PLS}^{(m)}$ if the first m PLS components are used.

PLS regression is done iteratively in an incremental manner, which renders it an excellent method in dealing with ultra-high dimensional scenarios ($p \gg n$). Specifically, setting $\hat{\mathbf{y}} = \mathbf{0}$, compute the PLS direction $\mathbf{a} := \mathbf{X}^T\mathbf{y}$ and optionally normalize. The PLS component is given by $\mathbf{z} := \mathbf{Xa}$. Then regress $\mathbf{y}$ on $\mathbf{z}$ so that update $\hat{\mathbf{y}} := \hat{\mathbf{y}} + P_{C(\mathbf{z})}\mathbf{y}$. Next, deflate $\mathbf{X}$ so that $\mathbf{X} := P_{C^\perp(\mathbf{z})}\mathbf{X} = (\mathbf{I} - P_{C(\mathbf{z})})\mathbf{X}$. Now go back to the beginning and iterate. The PLS regression procedure is described in Algorithm IV.

Algorithm IV: Partial Least Squares (PLS)

1. Set $\hat{\mathbf{y}}^{(0)} = \bar{y}\mathbf{j} = \mathbf{0}$ and $\mathbf{x}_j^{(0)} = \mathbf{x}_j$ for $j = 1, \ldots, p$.
2. For $m = 1, 2, \ldots, p$, do
   (a) Obtain $\mathbf{z}_m = \sum_{j=1}^p \langle \mathbf{x}_j, \mathbf{y}\rangle \mathbf{x}_j = \mathbf{X}\mathbf{X}^T\mathbf{y}$.
   (b) Update $\hat{\mathbf{y}}^{(m)}$ as
   $$\hat{\mathbf{y}}^{(m)} := \hat{\mathbf{y}}^{(m-1)} + \frac{\langle \mathbf{z}_m, \mathbf{y}\rangle}{\langle \mathbf{z}_m, \mathbf{z}_m\rangle}\mathbf{z}_m.$$
   (c) Orthogonalize each $\mathbf{x}_j$ with respect to $\mathbf{z}_m$. For $j = 1, 2, \ldots, p$,
   $$\mathbf{x}_j := \mathbf{x}_j - \frac{\langle \mathbf{z}_m, \mathbf{x}_j\rangle}{\langle \mathbf{z}_m, \mathbf{z}_m\rangle}\mathbf{z}_m$$
3. Output the fitted vector $\{\hat{\mathbf{y}}^{(m)}\}_{m=1}^p$ and the PLS components $\{\mathbf{z}_m\}_{m=1}^p$.

Denote $\hat{\phi}_{mj} = \langle \mathbf{x}_j^{(m-1)}, \mathbf{y}\rangle$ in step 2(a) of Algorithm IV. $\hat{\boldsymbol{\phi}}^{(m)} = (\hat{\phi}_{mj})_{j=1}^p$ is called the mth PLS direction $\mathbf{a}_m$ solves the following optimization problem:

$$\max_{\mathbf{a}} \left\{\text{corr}(y, \mathbf{x}^T\mathbf{a}) \cdot \text{var}(\mathbf{x}^T\mathbf{a})\right\} \quad \text{subject to} \quad \| \mathbf{a} \| = 1 \text{ and } \mathbf{a}^T\boldsymbol{\Sigma}_{xx}\mathbf{a}_k = 0, \text{ for } k = 1, \ldots, m-1.$$

Here $\boldsymbol{\Sigma}_{xx}$ denotes the sample variance-covariance matrix of $\mathbf{x}$.

Note that the *deflation* step for $\mathbf{X}$ could have been done for $\mathbf{y}$ as well so that $\mathbf{y} := P_{C^\perp(\mathbf{z})}\mathbf{y}$. But this is unnecessary because $P_{C^\perp(\mathbf{z})}$ is a projection matrix (hence idempotent). It can be checked that either $\mathbf{y}$ or $P_{C^\perp(\mathbf{z})}\mathbf{y}$ gives the same $\mathbf{a}$ and $\mathbf{z}$. As a direct result from the *deflation* step on $\mathbf{X}$, it can be easily checked that the PLS components are orthogonal to each other as formally given by the following proposition.

Proposition 2. *With PLS regression,* $\mathbf{z}_j^T\mathbf{z}_{j'} = 0$ *for any* $j \neq j' = 1, \ldots, k$

Proof. WLOG, let's show $\mathbf{z}_1^T \mathbf{z}_2 = 0$. Note $\mathbf{z}_2 = \mathbf{P}_{C^\perp(\mathbf{z}_1)} \mathbf{X}\mathbf{a}_2$. It follows that

$$\mathbf{z}_1^T \mathbf{z}_2 = \mathbf{z}_1^T \mathbf{P}_{C^\perp(\mathbf{z}_1)} \mathbf{X}\mathbf{a}_2 = 0,$$

since $\mathbf{z}_1^T \mathbf{P}_{C^\perp(\mathbf{z}_1)} = 0$. □

PLS seeks directions that have high variance and have high correlation with the response, in contrast to principal components regression which keys only on high variance. Nevertheless, the variance part often dominates the objective function, making PLS and PCR similar. From the insights often by SVD of $\mathbf{X}$, both are also similar to ridge regression. With that being said, PLS has some advantages to PCR since PLS takes the correlation with response into consideration and often entails a smaller number of components than PCR.

5 Total Least Squares

The *Total Least Squares* seeks the line or surface plane ℓ such that the overall (perpendicular) distance between observed points $\mathbf{z}_i := (y_i, \mathbf{x}_i) \in \mathbb{R}^{p+1}$ for $i = 1, \ldots, n$ and ℓ is minimized. Namely,

$$\hat{\ell} = \arg\min_{\ell} Q(\ell) = \arg\min_{\ell} \sum_{i=1}^{n} \text{dist}\left\{(y_i, \mathbf{x}_i); \ell\right\}^2. \tag{30}$$

Figure 3 provides an illustration of the different focuses in ordinary least squares (OLS) and total least squares (TLS). TLS minimizes the perpendicular distance from each point to the straight line while OLS minimizes the vertical distance between observed and predicted values. Note that TLS is not scale invariant. TLS is one standard solution to the errors-in-variables or measurement error modeling.

To express the perpendicular distance from a point to a plane (or line), it is more convenient to use a common representation of plane or line in analytic geometry. In this representation, a plane ℓ in $\mathbb{R}^{p+1}$ can be uniquely determined by two elements: a point $\mathbf{w}$ on ℓ and the (unit) normal direction $\mathbf{r}$ that is perpendicular to ℓ, as shown in Figure 4

Given $\mathbf{w}$ and $\mathbf{r}$, ℓ can be defined as

$$\ell := \left\{\mathbf{z} \in \mathbb{R}^{p+1} : \mathbf{r}^T(\mathbf{z} - \mathbf{w}) = 0, \text{ where } \| \mathbf{r} \| = 1.\right\} \tag{31}$$

With this above definition, ℓ is often expressed as $\ell = \mathbf{w} + \mathbf{r}^\perp$.

It follows that the distance from any point $\mathbf{z} \in \mathbb{R}^{p+1}$ to ℓ in (31) is given by

$$\text{dist}(\mathbf{z}, \ell) := | \mathbf{r}^T(\mathbf{z} - \mathbf{w}) | . \tag{32}$$

Thus the criterion $Q(\ell)$ in (30) can be written $Q(\ell) = Q(\mathbf{r}, \mathbf{w})$. The TLS problem becomes: seeking $(\mathbf{r}, \mathbf{w})$ that minimizes

$$Q(\mathbf{r}, \mathbf{w}) = \sum_{i=1}^{n} \left\{\mathbf{r}^T(\mathbf{z}_i - \mathbf{w})\right\}^2 = \sum_{i=1}^{n} \left\{ \sum_{j=1}^{p} r_j(x_{ij} - w_j) + r_{p+1}(y_i - w_{p+1}) \right\}^2. \tag{33}$$

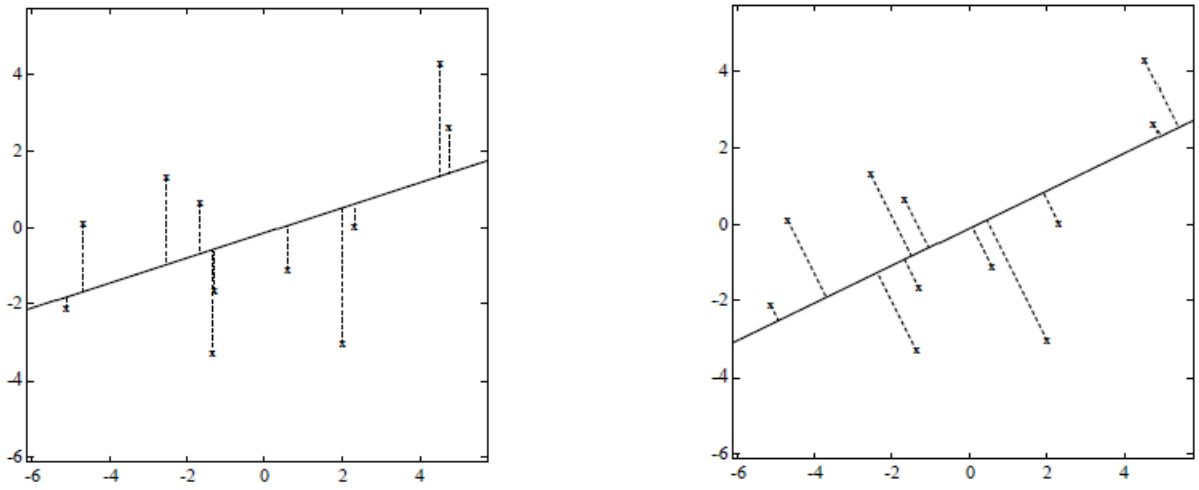

Figure 3: A 2-D Illustration of Ordinary Least Squares (OLS; left panel) and Total Least Squares (TLS; right panel). In this case, the slope of the line is given by the first PC direction for data $(\mathbf{x}, \mathbf{y})$, (and the normal direction specified by their second PC).

Now it can be shown that

$$\begin{aligned} Q(\mathbf{r}, \mathbf{w}) &= \sum_{i=1}^{n} \left\{ \mathbf{r}^T (\mathbf{z}_i - \mathbf{w}) \right\}^2 \\ &= \sum_{i=1}^{n} \left\{ \mathbf{r}^T (\mathbf{z}_i - \bar{\mathbf{z}} + \bar{\mathbf{z}} + \mathbf{w}) \right\}^2 \\ &= \sum_{i=1}^{n} \left\{ \mathbf{r}^T (\mathbf{z}_i - \bar{\mathbf{z}}) \right\}^2 + n \cdot \left\{ \mathbf{r}^T (\bar{\mathbf{z}} - \mathbf{w}) \right\}^2 \\ &\geq \sum_{i=1}^{n} \left\{ \mathbf{r}^T (\mathbf{z}_i - \bar{\mathbf{z}}) \right\}^2 \end{aligned} \tag{34}$$

with "=" held iff $\mathbf{w} = \bar{\mathbf{z}}$. This implies the optimization problem in TLS can be reduced to what follows: with $\mathbf{w} = \bar{\mathbf{z}}$ fixed, seeking $\mathbf{r}$ that minimizes

$$Q(\mathbf{r}) = \sum_{i=1}^{n} \left\{ \mathbf{r}^T (\mathbf{z}_i - \bar{\mathbf{z}}) \right\}^2 = \mathbf{r}^T \mathbf{Z}_0^T \mathbf{Z}_0 \mathbf{r}, . \tag{35}$$

where

$$\mathbf{Z}_0 := (\mathbf{x} - \bar{\mathbf{x}} \mid \mathbf{y} - \bar{y}\mathbf{j}) = \begin{pmatrix} x_{11} - \bar{x}_1 & \cdots & x_{1p} - \bar{x}_p & y_1 - \bar{y} \\ x_{21} - \bar{x}_1 & \cdots & x_{2p} - \bar{x}_p & y_2 - \bar{y} \\ \vdots & & \vdots & \vdots \\ x_{n1} - \bar{x}_1 & \cdots & x_{np} - \bar{x}_p & y_n - \bar{y} \end{pmatrix} = \mathbf{P}_n \mathbf{Z}. \tag{36}$$

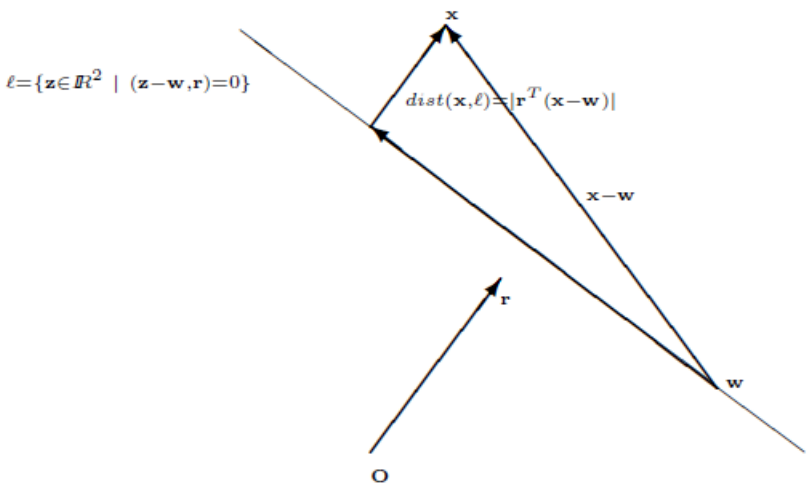

Figure 4: Expression of the line ℓ in the plane through a point $\mathbf{w}$ on ℓ and the unit normal direction $\mathbf{r}$ orthogonal to ℓ. The figure is taken from de Groen (1996).

with $\mathbf{P}_n = \mathbf{I}_n - \mathbf{J}_n/n$ and $\mathbf{Z} = (\mathbf{X},\ \mathbf{y})$ is the column-centered matrix.

The solution $\hat{\mathbf{z}}$ is given as the right singular vector of $\mathbf{Z}_0$ corresponding to its smallest singular value. That is, let $\mathbf{Z}_0 = \mathbf{UDV^T}$ be the singular value decomposition of $\mathbf{Z}_0$. Let $\mathbf{v}_{p+1}$ denote the last column of $\mathbf{V}$. Then $\hat{\mathbf{r}} = \mathbf{v}_{p+1}$. Therefore, the TLS plane $\hat{\ell}$ is

$$\hat{\ell} := \left\{\mathbf{z} \in \mathbb{R}^{p+1} : \mathbf{v}_{p+1}^T(\mathbf{z} - \bar{\mathbf{z}}) = 0\right\}.$$

We often re-express $\hat{\ell}$ in the form of $y = a + \mathbf{b}^T\mathbf{x}$ so that we may predict y via TLS.

References

Breheny, P. and Huang, J. (2011). Coordinate descent algorithms for nonconvex penalized regression, with applications to biological feature selection. *The Annals of Applied Statistics*, **5**: 232–253.

de Groen, P. (1996). An Introduction to Total Least Squares. *Nieuw Archief voor Wiskunde, Vierde serie*, **14**: 237–253.

Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. (2004). Least Angle Regression (with discussion). *Annals of Statistics*, **32**: 407–499.

Fan, J. and Li, R. (2001), Variable Selection via Nonconcave Penalized Likelihood and Its Oracle Properties. *Journal of the American Statistical Association*, **96**: 1348–1360.

Hastie, T., Tibshirani, R., and Friedman, J. H. (2008). *Elements of Statistical Learning*, 2nd Edition. Chapman & Hall.

Hastie, T., Tibshirani, R., and Wainwright, M. (2015). *Statistical Learning with Sparsity: The Lasso and Generalizations*. CRC Press.

Osborne, M. R., Presnell, B., and Turlach, B. A. (2000). On the LASSO and Its Dual. *Journal of Computational and Graphical Statistics*, **9**(2): 319–337.

Shibata, R. (1981). An optimal selection of regression variables. *Biometrika*, **68**: 45–54.

Su, X., Wonkye, Y., Wang, P., and Yin, X. (2017+). Weighted Orthogonal Components Regression Analysis. arXiv preprint, arXiv:1709.04135. URL https://arxiv.org/pdf/1709.04135.pdf

Tibshirani, R. (2013). The Lasso problem and uniqueness. *Electronic Journal of Statistics*, **7**: 1456–1490.