Ridge regression

Statistical Learning
CLAMSES - University of Milano-Bicocca

Aldo Solari

References

- Hastie, T. (2020). Ridge regularization: an essential concept in data science. Technometrics, 62(4), 426-433.
- van Wieringen (2015). Lecture notes on ridge regression. arXiv preprint arXiv:1509.09169.

Condition number

- In the linear model, the estimate of β is obtained by solving the normal equations
$$X^\top X\beta = X^\top y$$
- The difficulty of solving this system of linear equations can be described by the *condition number*
$$\kappa(X^\top X) = \frac{d_{\max}}{d_{\min}}$$
the ratio between the largest and smallest singular values of $X^\top X$
- If the condition number is very large, then the matrix is said to be *ill-conditioned* (see Section 2.6 of CASL)

Toy linear model with $n = p = 2$. We set X and β as

$$X = \begin{bmatrix} 10^9 & -1 \\ -1 & 10^{-5} \end{bmatrix} \quad \beta = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

And if we define $y = X\beta$, this gives

$$y = \begin{bmatrix} 10^9 & -1 \\ -1 & 10^{-5} \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 10^9 - 1 \\ -0.99999 \end{bmatrix}$$

The reciprocal of condition number, i.e. $1/\kappa(X^\top X) = 9.998e - 29$, is smaller than (my) machine precision, i.e. $2.220446e - 16$

```
X <- matrix(c(10^9, -1, -1, 10^(-5)), 2, 2)
beta <- c(1,1)
y <- X %*% beta

solve( crossprod(X), crossprod(X, y) )

Error in solve.default(crossprod(X)) :
system is computationally singular:
reciprocal condition number = 9.998e-29

.Machine$double.eps
2.220446e-16
```

Ridge regression solution

- Ridge provides a remedy for an *ill-conditioned* X^tX matrix
- If our $n \times p$ design matrix X has column rank less than p (or nearly so in terms of its condition number), then the usual least-squares regression equation is in trouble:

$$\hat{\beta} = (X^tX)^{-1}X^ty$$

- What we do is add a *ridge* on the diagonal - $X^tX + \lambda I_p$ with $\lambda > 0$ - which takes the problem away:

$$\hat{\beta}_\lambda = (X^tX + \lambda I_p)^{-1}X^ty$$

- This is the ridge regression solution proposed by Hoerl and Kennard (1970)

- Ridge regression modifies the normal equations to

$$(X^\mathsf{T} X + \lambda I_p)\beta = X^\mathsf{T} y$$

  and the condition number of $(X^\mathsf{T} X + \lambda I_p)$ is

$$\kappa(X^\mathsf{T} X + \lambda I_p) = \frac{d_{\max} + \lambda}{d_{\min} + \lambda}$$

- Notice that even if $d_{\min} = 0$, the condition number will be finite if $\lambda > 0$
- This technique is known as Tikhonov regularization, after the Russian mathematician Andrey Tikhonov

Penalized (Lagrange) form

- The optimization problem that ridge is solving

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

  where $\|\cdot\|$ is the ℓ_2 Euclidean norm
- The ridge remedy comes with consequences. The ridge estimate is biased toward zero. It also has smaller variance than the OLS estimate.
- Selecting λ amounts to a bias-variance trade-off

Cement data

$n = 13, p = 4$

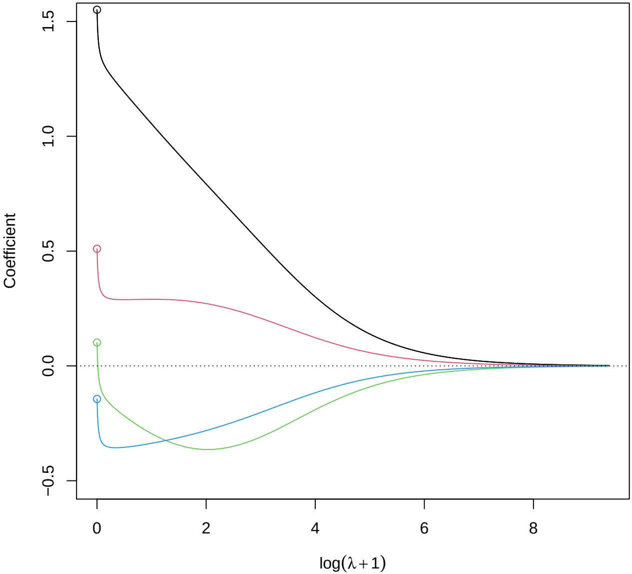
$$R = \begin{bmatrix} 1 & 0.23 & -0.82 & -0.25 \\ 0.23 & 1 & -0.14 & -0.97 \\ -0.82 & -0.14 & 1 & 0.03 \\ -0.25 & -0.97 & 0.03 & 1 \end{bmatrix}$$

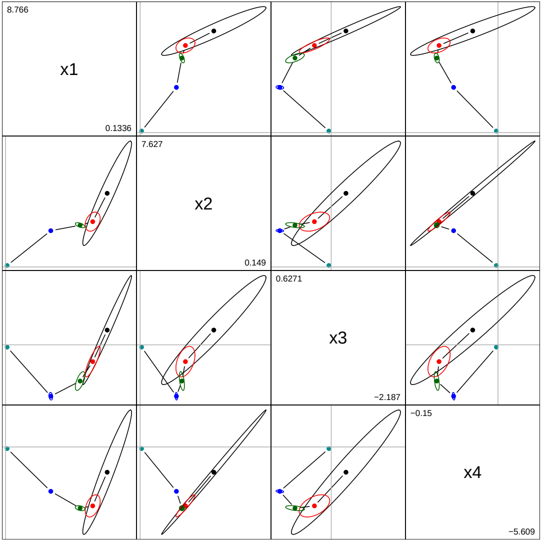
	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	62.41	70.07	0.89	0.40
x1	1.55	0.74	2.08	0.07
x2	0.51	0.72	0.70	0.50
x3	0.10	0.75	0.14	0.90
x4	-0.14	0.71	-0.20	0.84

R-squared: 0.9824

	x1	x2	x3	x4
VIF	38.50	254.42	46.87	282.51

Piepel, Redgate (1998) A Mixture Experiment Analysis of the Hald Cement Data, The American Statistician, 52:23–30


Coefficient
1.5
1.0
0.5
0.0
−0.5
0
2
4
6
8
log(λ + 1)



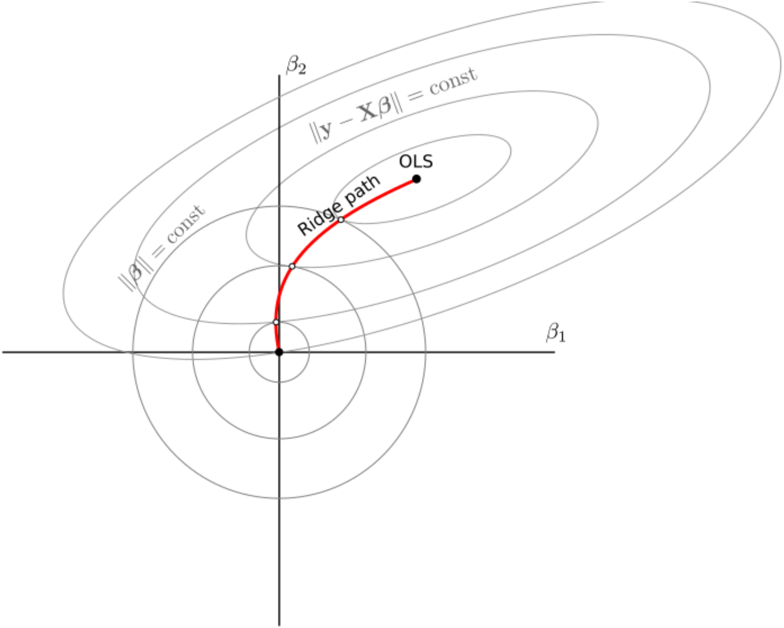
$$\lambda = 0, 0.1, 1, 10, 1000$$

Constrained form

- We can also express the ridge problem as

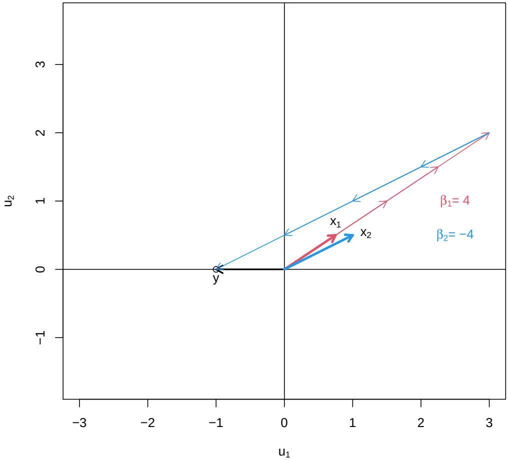
$$\min_{\beta} \|y - X\beta\|^2 \quad \text{subject to} \ \|\beta\| \leq c \tag{2}$$

- The two problems are of course equivalent: every solution $\hat{\beta}_\lambda$ in (1) is a solution to (2) with $c = \|\hat{\beta}_\lambda\|$



β_2
$\|\mathbf{y}-\mathbf{X}\beta\| = \text{const}$
OLS
Ridge path
$\|\beta\| = \text{const}$
β_1

Overfitting



Large estimates of β are often an indication of overfitting

Bayesian view

- Assume

$$y_i|\beta, X = x_i \sim x_i^t\beta + \epsilon_i$$

with ϵ_i i.i.d. $N(0, \sigma_\epsilon^2)$. Here we think of β as random as well, and having a prior distribution

$$\beta \sim N(0, \sigma_\beta^2 I_p)$$

- Then the negative log posterior distribution is proportional to (1), with

$$\lambda = \frac{\sigma_\epsilon^2}{\sigma_\beta^2}$$

and the posterior mean is the ridge estimator

- The smaller the prior variance parameter σ_β^2, the more the posterior mean is shrunk toward zero, the prior mean for β

Important details

- When including an intercept term, we usually leave this coefficient unpenalized, solving

$$\min_{\alpha,\beta} \|y - 1\alpha - X\beta\|^2 + \lambda\|\beta\|^2$$

- Ridge regression is not invariant under scale transformations of the variables, so it is standard practice to centre each column of X (hence making them orthogonal to the intercept term) and then scale them to have Euclidean norm $\sqrt{n}$
- It is straightforward to show that after this standardisation of X, $\hat{\alpha} = \bar{y}$, so we can also centre y and then remove α from our objective function
- Different `R` packages have different defaults, e.g. `glmnet` also standardizes y

- Let $\tilde{y} = (y - 1\bar{y})$ and $\tilde{X} = (X - 1\bar{x}^t)\text{diag}(1/s)$ be the centered y and standardized X, respectively, with
  - $\bar{y} = (1/n)\sum_{i=1}^n y_i$,
  - $\bar{x} = (1/n)X'1$,
  - $s = (s_1, \ldots, s_p)^t$ and $s_j^2 = (1/n)\sum_{i=1}^n (x_{ij} - \bar{x}_j)^2$
- Compute the scaled coefficients

$$\tilde{\beta}_\lambda = (\tilde{X}^t\tilde{X} + \lambda I_p)^{-1}\tilde{X}^t\tilde{y}$$

- Transform back to unscaled coefficients

$$\hat{\beta}_\lambda = \text{diag}(1/s)\tilde{\beta}_\lambda \quad \hat{\alpha} = \bar{y} - \bar{x}^t\hat{\beta}_\lambda$$

Ridge computations and the SVD

Tuning parameter

- In many wide-data and other ridge applications, we need to treat λ as a tuning parameter, and select a good value for the problem at hand.
- For this task we have a number of approaches available for selecting λ from a series of candidate values:
  - With a validation dataset separate from the training data, we can evaluate the prediction performance at each value of λ
  - Cross-validation does this effciently using just the training data, and leave-one-out (LOO) CV is especially efficient

SVD

- Whatever the approach, they all require computing a number of solutions $\hat{\beta}_\lambda$ at different values of λ: the *ridge regularization path*
- We can achieve great efficiency via the (full form) Singular Value Decomposition (SVD)

$$X = UDV^t$$

where U $n \times n$ orthogonal, V $p \times p$ orthogonal and D $n \times p$ diagonal, with diagonal entries $d_1 \geq \ldots \geq d_m \geq 0$, where $m = \min(n, p)$

- From the SVD we get

$$\begin{aligned}
\hat{\beta}_\lambda &= (VD^tU^tUDV^t + \lambda VV^t)^{-1}VD^tU^ty \\
&= V(D^tD + \lambda I_p)^{-1}D^tU^ty \\
&= \sum_{d_j>0} v_j \frac{d_j}{d_j^2 + \lambda} \langle u_j, y \rangle
\end{aligned} \tag{3}$$

  where v_j (u_j) is the jth column of V (U), and $\langle a, b \rangle = a^t b$

- Once we have the SVD of X, we have the ridge solution for all values of λ
- When $n > p$ the ridge solution with $\lambda = 0$ is simply the OLS solution for β
- When $p > n$, there are infinitely many least squares solutions for β, all leading to a zero-residual solution. From (3) with $\lambda = 0$ we get a unique solution, the one with minimum Euclidean norm

– Fitted values

$$
\begin{aligned}
\hat{y}_\lambda &= U\mathrm{diag}\left(\frac{d_1^2}{d_1^2+\lambda}, \ldots, \frac{d_p^2}{d_p^2+\lambda}\right)U^t y \\
&= \sum_{d_j>0} u_j \frac{d_j^2}{d_j^2+\lambda}\langle u_j, y\rangle
\end{aligned}
$$

Principal components regression

- Ridge
$$\hat{\beta}_\lambda = V\mathrm{diag}\left(\frac{d_1}{d_1^2+\lambda}, \ldots, \frac{d_p}{d_p^2+\lambda}\right)U^t y$$
- Principal components regression with q components
$$\hat{\beta}_q = V\mathrm{diag}\left(\frac{1}{d_1}, \ldots, \frac{1}{d_q}, 0, \ldots, 0\right)U^t y$$
- Both operate on the singular values, but where principal component regression thresholds the singular values, ridge regression shrinks them

Ridge and the bias-variance trade-off

Bias

- Assume that the data arise from a linear model $y \sim N(X\beta, \sigma^2 I_n)$, then $\hat{\beta}_\lambda$ will be a biased estimate of β. Throughout this section X is assumed fixed, $n > p$ and X has full column rank
- The ridge estimator can be expressed as

$$\hat{\beta}_\lambda = (X^t X + \lambda I_p)^{-1} X^t X \hat{\beta}$$

- We can get an explicit expression for the bias

$$\begin{aligned}
\text{Bias}(\hat{\beta}_\lambda) &= \mathbb{E}(\hat{\beta}_\lambda) - \beta \\
&= V \text{diag}\left(\frac{\lambda}{d_1^2 + \lambda}, \ldots, \frac{\lambda}{d_p^2 + \lambda}\right) V^t \beta \\
&= \sum_{j=1}^{p} v_j \frac{\lambda}{d_j^2 + \lambda} \langle v_j, \beta \rangle
\end{aligned}$$

Variance

- Similarly there is a nice expression for the covariance matrix

$$\begin{aligned}\mathrm{Var}(\hat{\beta}_\lambda) &= \sigma^2 V \mathrm{diag}\left(\frac{d_1^2}{(d_1^2+\lambda)^2}, \ldots, \frac{d_p^2}{(d_p^2+\lambda)^2}\right) V^t \\ &= \sigma^2 \sum_{j=1}^{p} \frac{d_j^2}{(d_j^2+\lambda)^2} v_j v_j^t\end{aligned}$$

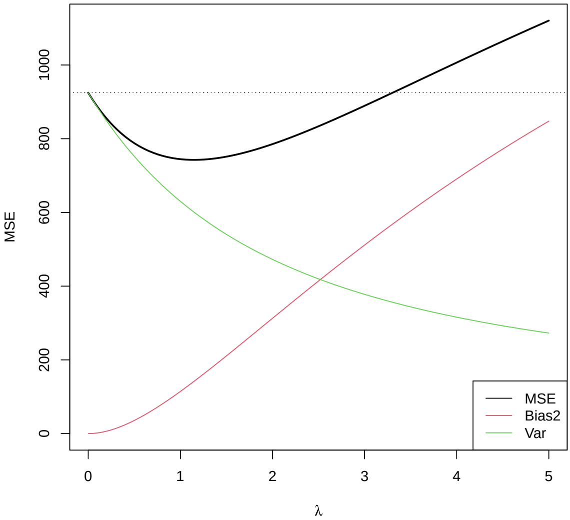
- With $\lambda = 0$, this is $\mathrm{Var}(\hat{\beta}) = \sigma^2 (X^t X)^{-1} \succeq \mathrm{Var}(\hat{\beta}_\lambda)$ for $\lambda > 0$

Mean Squared Error

- MSE of the ridge regression estimator

$$\begin{aligned} \mathrm{MSE}(\hat{\beta}_\lambda) &= \mathbb{E}[(\hat{\beta}_\lambda - \beta)^t(\hat{\beta}_\lambda - \beta)] \\ &= \mathrm{tr}[\mathrm{Var}(\hat{\beta}_\lambda)] + \mathrm{Bias}(\hat{\beta}_\lambda)^t \mathrm{Bias}(\hat{\beta}_\lambda) \end{aligned}$$

- *Theorem (Theobald, 1974)*
  There exists $\lambda > 0$ such that $\mathrm{MSE}(\hat{\beta}_\lambda) < \mathrm{MSE}(\hat{\beta})$.


MSE
Bias2
Var
0
200
400
600
800
1000
0
1
2
3
4
5
λ
MSE

Expected prediction error

- When we make predictions $\hat{y}_i = x_i^t \hat{\beta}_\lambda$ at x_i

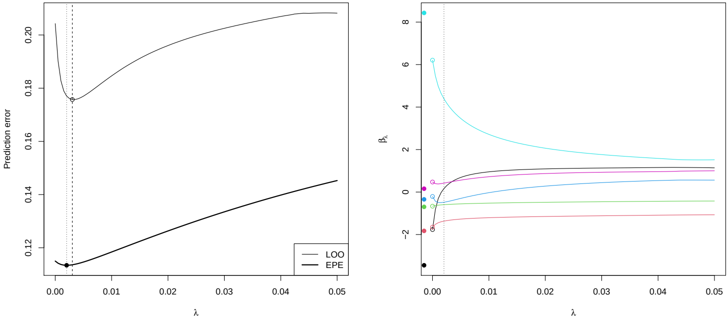
$$\begin{aligned} \mathrm{MSE}(\hat{y}_i) &= \mathbb{E}[(x_i^t \hat{\beta}_\lambda - x_i^t \beta)^2] \\ &= x_i^t \mathrm{Var}(\hat{\beta}_\lambda) x_i + [x_i^t \mathrm{Bias}(\hat{\beta}_\lambda)]^2 \end{aligned}$$

- Expected prediction error

$$\mathbb{E}\Big[\frac{1}{n}\sum_{i=1}^n (\hat{y}_i - y_i^{\mathrm{new}})^2\Big] = \frac{1}{n}\sum_{i=1}^n \mathrm{MSE}(\hat{y}_i) + \sigma^2$$

Longley data

$n = 16, p = 6$



Orthonormal design matrix

- Consider an orthonormal design matrix X, i.e. $X^t X = I_p = (X^t X)^{-1}$, e.g.

$$X = \frac{1}{2}\begin{bmatrix} -1 & -1 \\ -1 & 1 \\ 1 & -1 \\ 1 & 1 \end{bmatrix}$$

- $\hat{\beta}_\lambda = \frac{1}{(1+\lambda)}\hat{\beta}$
- $\mathrm{Var}(\hat{\beta}_\lambda) = \frac{\sigma^2}{(1+\lambda)^2} I_p$
- $\mathrm{MSE}(\hat{\beta}_\lambda) = \frac{p\sigma^2}{(1+\lambda)^2} + \frac{\lambda^2\|\beta\|^2}{(1+\lambda)^2}$ with minimum at $\lambda = \frac{p\sigma^2}{\|\beta\|^2}$

Ridge and leave-one-out cross validation

LOO

- For n-fold (LOO) CV, we have another beautiful result for ridge and other linear operators

$$\mathrm{LOO}_\lambda = \frac{1}{n}\sum_{i=1}^{n}(y_i - x_i^t\hat{\beta}_\lambda^{(-i)})^2 = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - x_i^t\hat{\beta}_\lambda}{1 - R_{ii}^\lambda}\right)^2$$

where $\hat{\beta}_\lambda^{(-i)}$ is the ridge estimate computed using the $(n-1)$ observations with the pair (x_i, y_i) and

$$R^\lambda = X(X^tX + \lambda I)^{-1}X^t$$

- The equation says we can compute all the LOO residuals for ridge from the original residuals, each scaled up by $1/(q - R_{ii}^\lambda)$
- We can obtain R^λ efficiently for all λ via

$$R^\lambda = U\mathrm{diag}\left(\frac{d_1^2}{d_1^2 + \lambda}, \ldots, \frac{d_p^2}{d_p^2 + \lambda}\right)U^t$$

- For each pair (x_i, y_i) left out, we solve

$$\min_{\beta} \sum_{l \neq i} (y_l - x_l^t \beta) + \lambda \|\beta\|^2$$

  with solution $\hat{\beta}_{\lambda}^{(-i)}$.
- Let $y_i^* = x_i^t \hat{\beta}_{\lambda}^{(-i)}$. If we insert the pair (x_i, y_i^*) back into the size $n-1$ dataset, it will not change the solution
- Back at a full n dataset, and using the linearity of the ridge operator, we have

$$y_i^* = \sum_{l \neq i} R_{il}^{\lambda} y_l + R_{ii}^{\lambda} y_i^* = \sum_{l=1}^{n} R_{il}^{\lambda} y_l - R_{ii}^{\lambda} y_i + R_{ii}^{\lambda} y_i^* = \hat{y}_i - R_{ii}^{\lambda} y_i + R_{ii}^{\lambda} y_i^*$$

  from which we see that $(y_i - y_i^*) = (y_i - \hat{y}_i)/(1 - R_{ii}^{\lambda})$

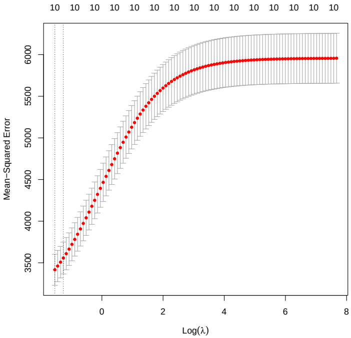
GCV

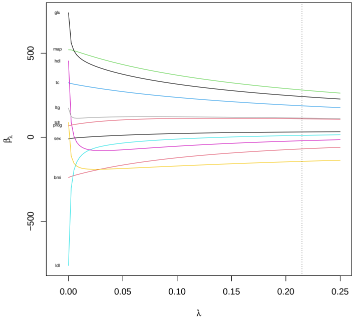
- The identity $\mathrm{tr}(R^\lambda) = \sum_{i=1}^n R_{ii}^\lambda$ suggests $R_{ii}^\lambda \approx \frac{1}{n}\mathrm{tr}(R^\lambda)$
- Generalized cross validation

$$\mathrm{GCV}_\lambda = \frac{1}{n}\sum_{i=1}^n \frac{(y_i - x_i^t\hat{\beta}_\lambda)^2}{(1-\frac{1}{n}\mathrm{tr}(R^\lambda))^2}$$

Diabetes data

$$n = 442, p = 10$$





Ridge and the kernel trick

- The fitted values from ridge regression are

$$\hat{y}_\lambda = X(X^tX + \lambda I_p)^{-1}X^ty \tag{4}$$

- An alternative way of writing this is suggested by the following

$$\begin{aligned} X^t(XX^t + \lambda I_n) &= (X^tX + \lambda I_p)X^t \\ (X^tX + \lambda I_p)^{-1}X^t &= X^t(XX^t + \lambda I_n)^{-1} \\ X(X^tX + \lambda I_p)^{-1}X^ty &= XX^t(XX^t + \lambda I_n)^{-1}y \end{aligned}$$

giving

$$\hat{y}_\lambda = K(K + \lambda I_n)^{-1}y \tag{5}$$

where $K = XX^t = \{x_i^t x_j\}_{ij}$ is the $n \times n$ *gram* matrix of pairwise inner products, where x_i^t and x_j^t are the ith and jth row of X

- Complexity can be expressed in terms of floating point operations (flops) required to find the solution. (4) requires $O(np^2 + p^3)$ operations, (5) $O(pn^2 + n^3)$ operations

- Suppose we want to add all pairwise interactions

$$\begin{aligned} x_{i1}, x_{i2}, \ldots, x_{ip} \\ x_{i1}x_{i1}, x_{i1}x_{i2}, \ldots, x_{i1}x_{ip} \\ \vdots \\ x_{ip}x_{i1}, x_{ip}x_{i2}, \ldots, x_{ip}x_{ip} \end{aligned}$$

giving $O(p^2)$ columns in the design matrix. Since (5) now requires $O(p^2 n^2 + n^3)$ operations, for large p it can be computationally prohibitive
- However, K can be computed directly with

$$K_{ij} = (\frac{1}{2} + x_i^t x_j)^2 - \frac{1}{4} = \sum_k x_{ik}x_{jk} + \sum_{k,l} x_{ik}x_{il}x_{jk}x_{jl}$$

this amounts to an inner product between vectors of the form

$$(x_{i1}, \ldots, x_{ip}, x_{i1}x_{i1}, \ldots, x_{i1}x_{ip}, x_{i2}x_{i1}, \ldots, x_{i2}x_{ip}, \ldots, x_{ip}x_{ip})$$

and it requires $O(pn^2)$ operations