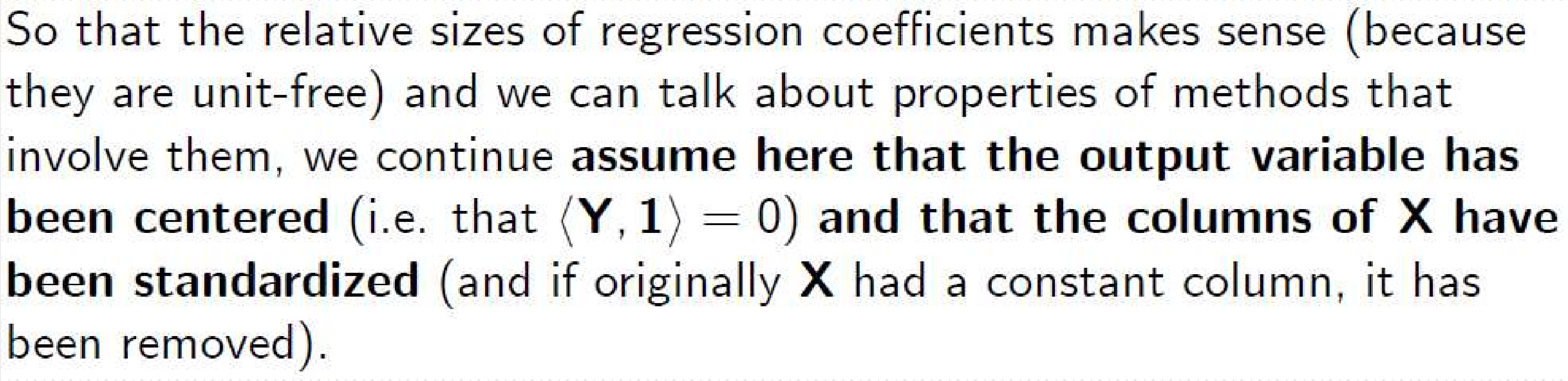
Non-OLS Linear SEL Prediction: LASSO etc.

Stephen Vardeman

Analytics Iowa LLC

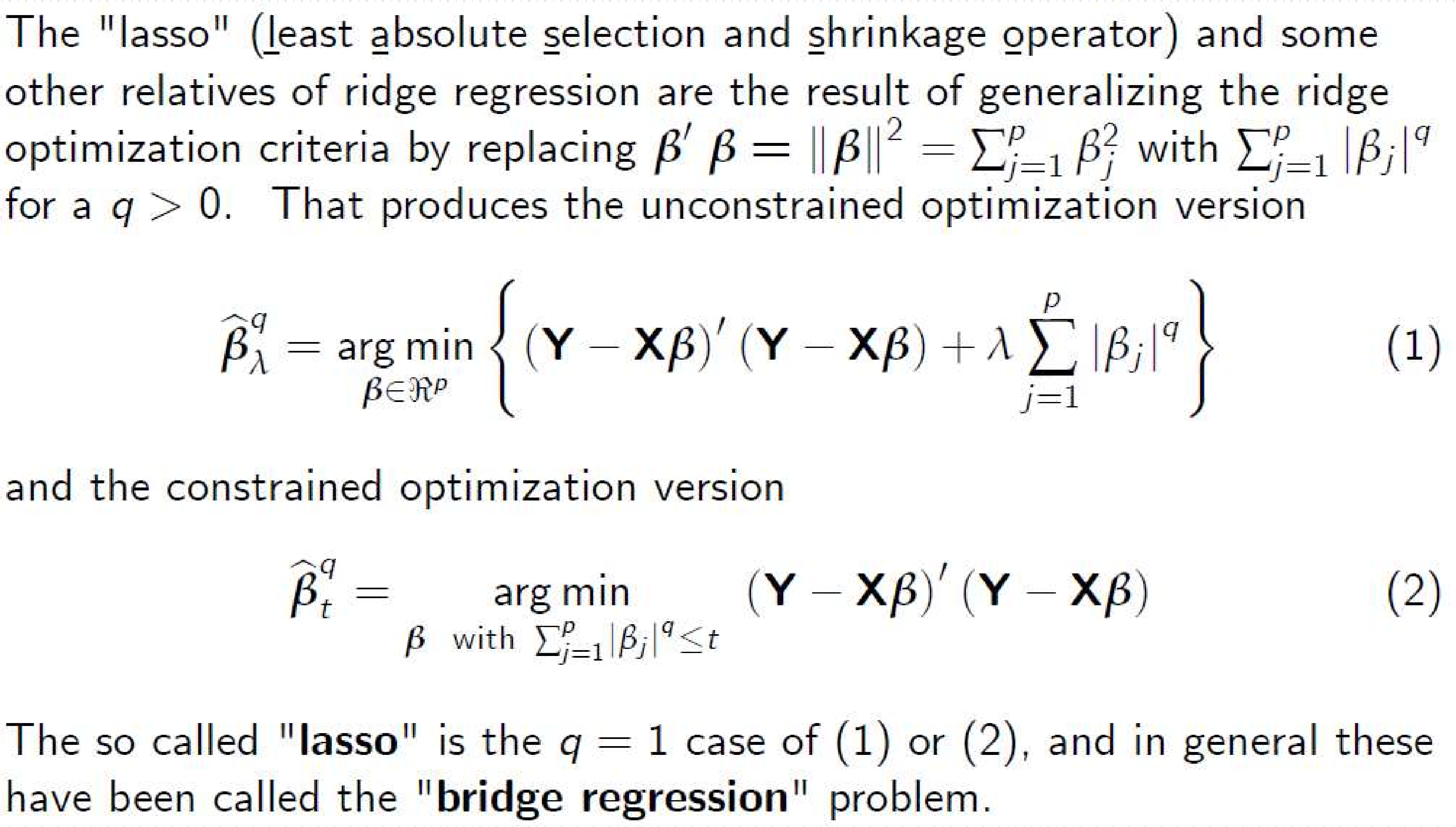
ISU Statistics and IMSE

Reiteration of the context



So that the relative sizes of regression coefficients makes sense (because they are unit-free) and we can talk about properties of methods that involve them, we continue **assume here that the output variable has been centered** (i.e. that $\langle \mathbf{Y}, \mathbf{1} \rangle = 0$) **and that the columns of X have been standardized** (and if originally $\mathbf{X}$ had a constant column, it has been removed).

Lasso formulations



The "lasso" (least absolute selection and shrinkage operator) and some other relatives of ridge regression are the result of generalizing the ridge optimization criteria by replacing $\boldsymbol{\beta}' \boldsymbol{\beta} = \|\boldsymbol{\beta}\|^2 = \sum_{j=1}^{p} \beta_j^2$ with $\sum_{j=1}^{p} |\beta_j|^q$ for a $q > 0$. That produces the unconstrained optimization version

$$\widehat{\boldsymbol{\beta}}_{\lambda}^{q} = \underset{\boldsymbol{\beta} \in \Re^p}{\arg\min} \left\{ (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})' (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}) + \lambda \sum_{j=1}^{p} |\beta_j|^q \right\} \tag{1}$$

and the constrained optimization version

$$\widehat{\boldsymbol{\beta}}_{t}^{q} = \underset{\boldsymbol{\beta} \text{ with } \sum_{j=1}^{p} |\beta_j|^q \leq t}{\arg\min} (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})' (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}) \tag{2}$$

The so called "**lasso**" is the $q = 1$ case of (1) or (2), and in general these have been called the "**bridge regression**" problem.

Shrinking and variable selection

That is, for $t > 0$

$$\widehat{\boldsymbol{\beta}}_t^{\text{lasso}} = \underset{\beta \ \text{ with } \sum_{j=1}^{p} |\beta_j| \le t}{\arg\min} \left(\mathbf{Y} - \mathbf{X}\beta\right)' \left(\mathbf{Y} - \mathbf{X}\beta\right) \tag{3}$$

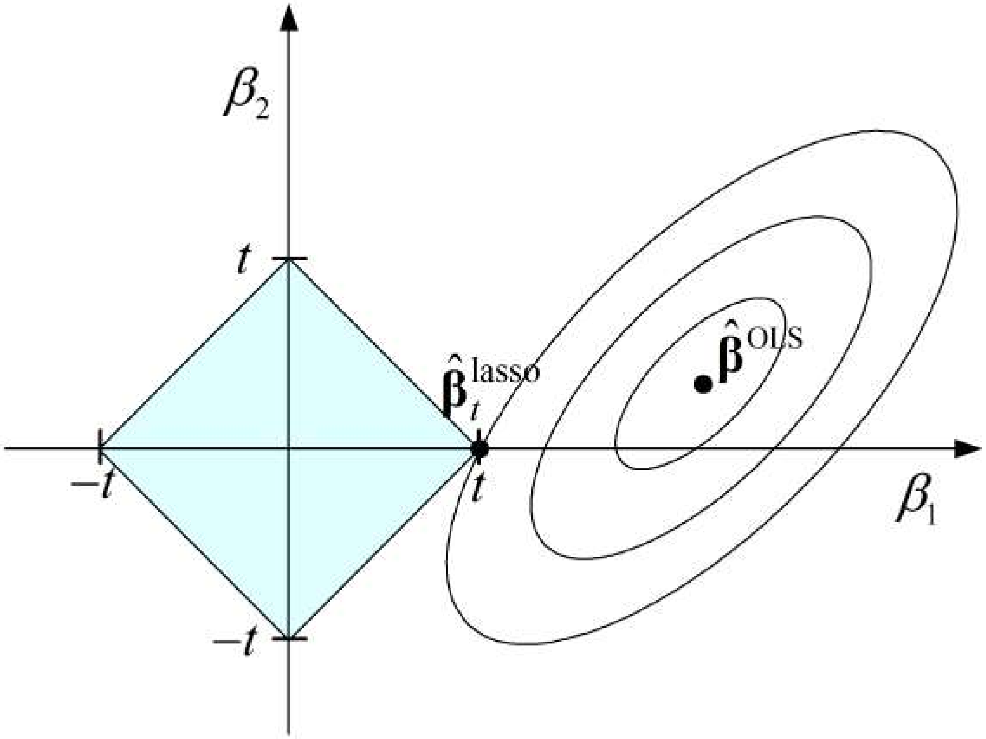
Because of the shape of the constraint region

$$\left\{ \boldsymbol{\beta} \in \Re^p \middle| \ \sum_{j=1}^{p} |\beta_j| \le t \right\}$$

(in particular its sharp corners at coordinate axes) some coordinates of $\widehat{\boldsymbol{\beta}}_t^{\text{lasso}}$ are often 0, and the lasso automatically provides simultaneous shrinking of $\widehat{\boldsymbol{\beta}}^{\text{OLS}}$ toward $\mathbf{0}$ and rational subset selection. (The same is true of cases of (2) with $q < 1$.)

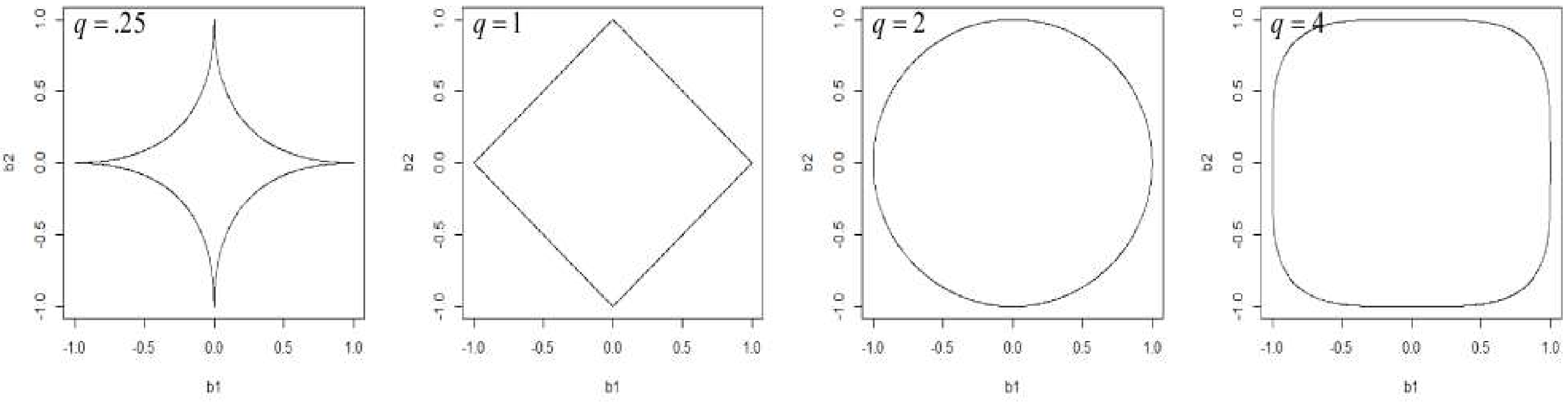
Geometry of lasso optimization

Here is a representation of the constrained optimization problem solved by the lasso coefficient vector, $\widehat{\beta}_t^{\text{lasso}}$ for $p = 2$.

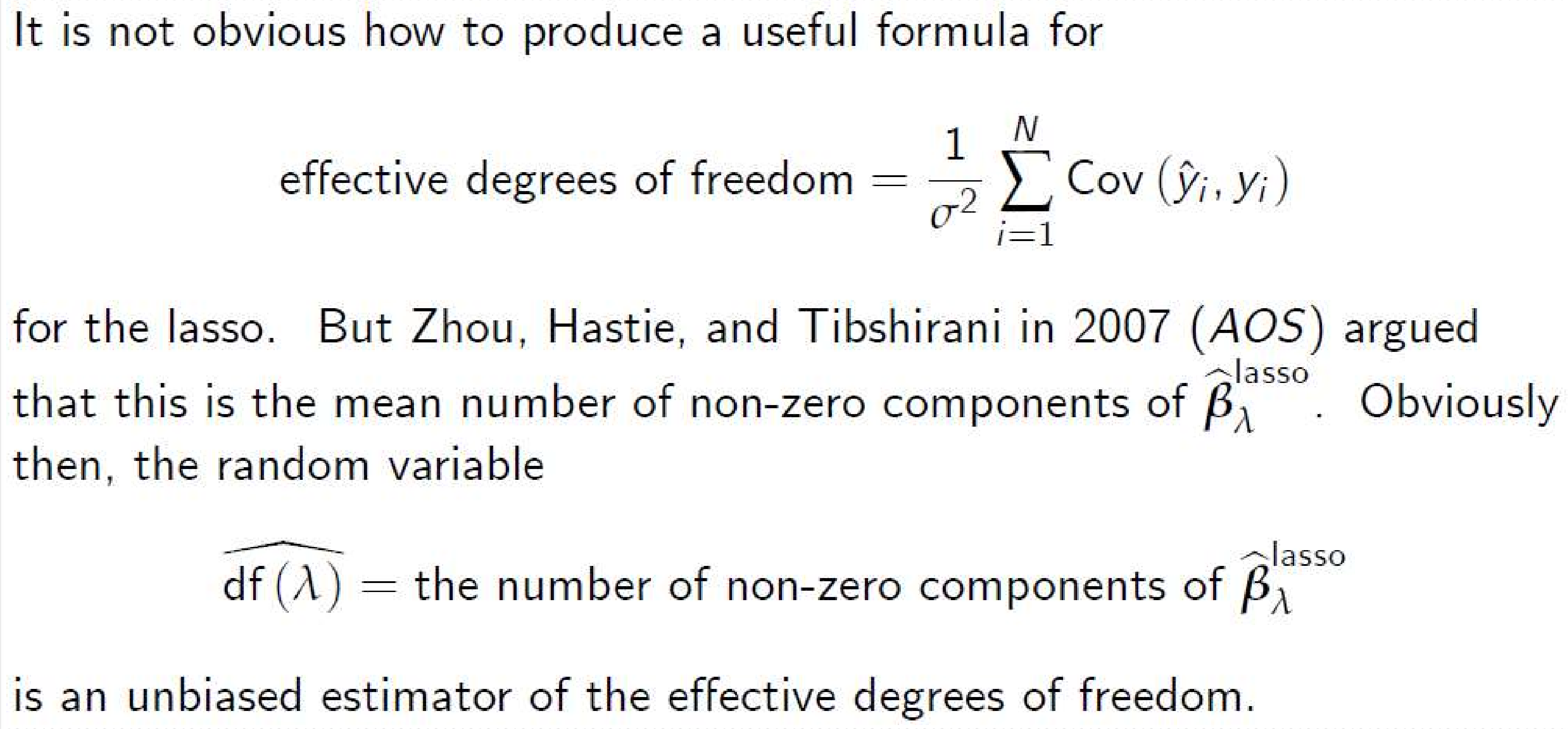

β_2
t
$\hat{\boldsymbol{\beta}}^{\text{OLS}}$
$\hat{\boldsymbol{\beta}}_t^{\text{lasso}}$
$-t$
t
β_1
$-t$

Bridge regression constraint regions

For comparison purposes, here are representations of $p = 2$ bridge regression constraint regions for $t = 1$. For $q < 1$ the regions not only have "corners," but are not convex.



Lasso effective degrees of freedom



It is not obvious how to produce a useful formula for

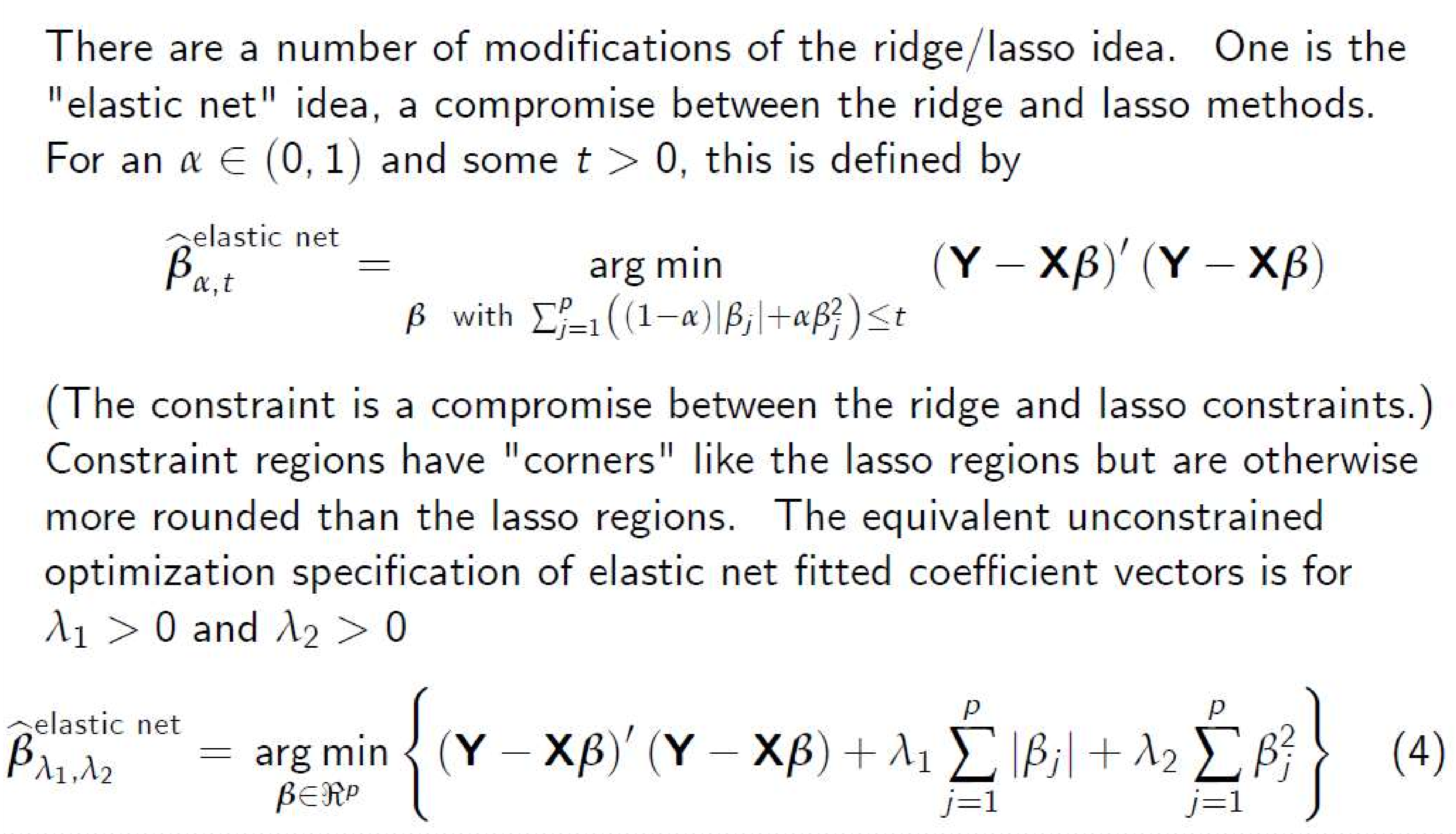
$$\text{effective degrees of freedom} = \frac{1}{\sigma^2} \sum_{i=1}^{N} \text{Cov}\left(\hat{y}_i, y_i\right)$$

for the lasso. But Zhou, Hastie, and Tibshirani in 2007 (AOS) argued that this is the mean number of non-zero components of $\widehat{\boldsymbol{\beta}}_\lambda^{\text{lasso}}$. Obviously then, the random variable

$$\widehat{\text{df}\left(\lambda\right)} = \text{the number of non-zero components of } \widehat{\boldsymbol{\beta}}_\lambda^{\text{lasso}}$$

is an unbiased estimator of the effective degrees of freedom.

Elastic net formulations



There are a number of modifications of the ridge/lasso idea. One is the "elastic net" idea, a compromise between the ridge and lasso methods. For an $\alpha \in (0,1)$ and some $t > 0$, this is defined by

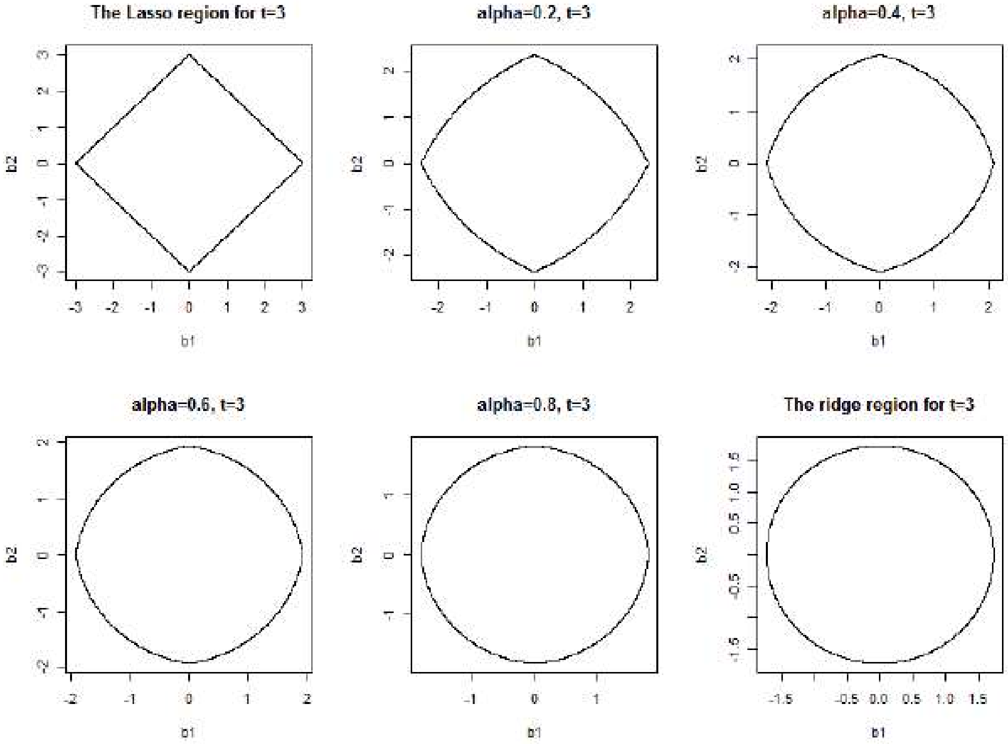
$$\widehat{\beta}_{\alpha,t}^{\text{elastic net}} = \underset{\beta \ \text{ with } \sum_{j=1}^{p}\left((1-\alpha)|\beta_j| + \alpha\beta_j^2\right) \leq t}{\arg\min} (\mathbf{Y} - \mathbf{X}\beta)'(\mathbf{Y} - \mathbf{X}\beta)$$

(The constraint is a compromise between the ridge and lasso constraints.) Constraint regions have "corners" like the lasso regions but are otherwise more rounded than the lasso regions. The equivalent unconstrained optimization specification of elastic net fitted coefficient vectors is for $\lambda_1 > 0$ and $\lambda_2 > 0$

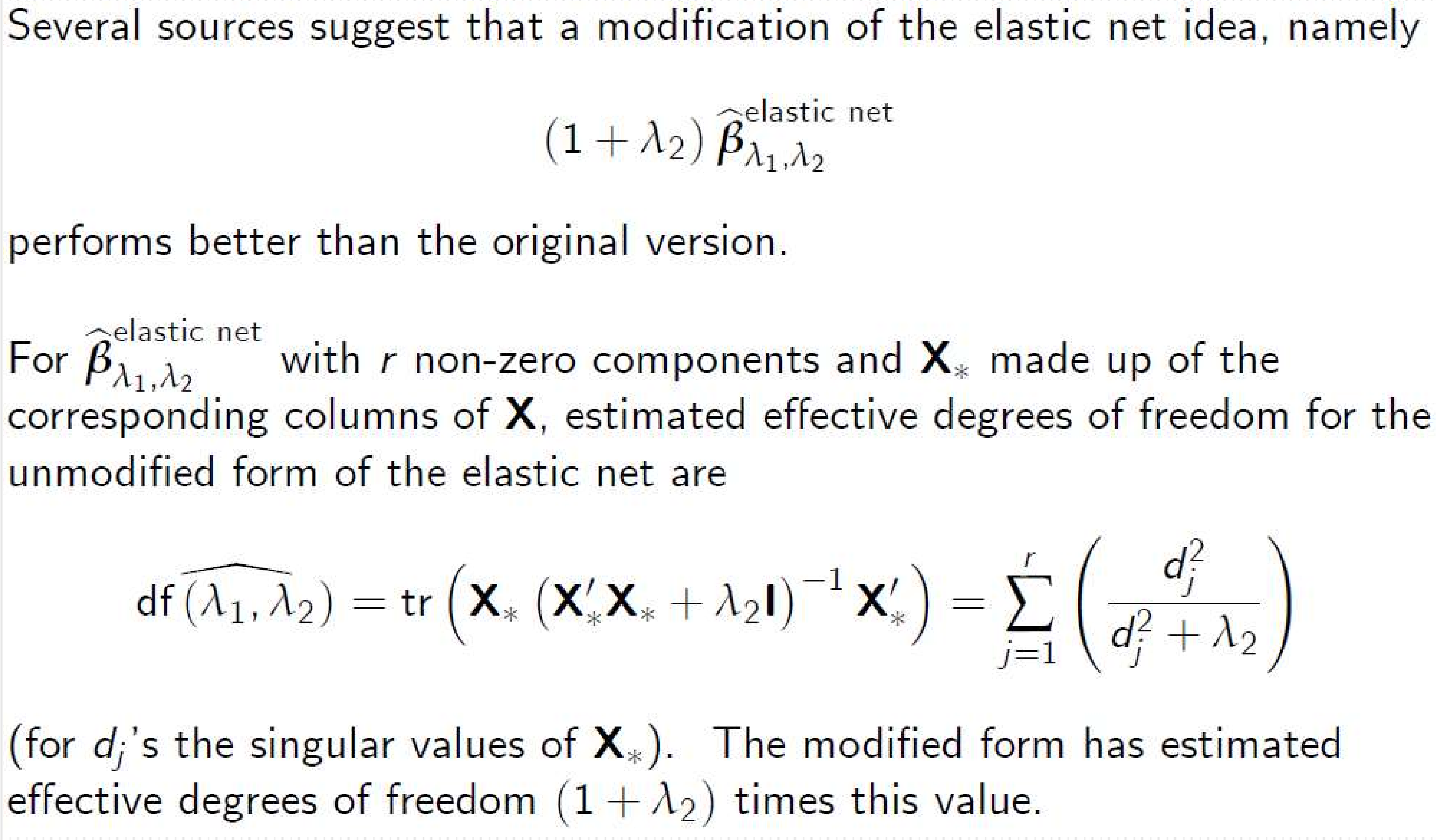
$$\widehat{\beta}_{\lambda_1,\lambda_2}^{\text{elastic net}} = \underset{\beta \in \Re^p}{\arg\min} \left\{ (\mathbf{Y} - \mathbf{X}\beta)'(\mathbf{Y} - \mathbf{X}\beta) + \lambda_1 \sum_{j=1}^{p} |\beta_j| + \lambda_2 \sum_{j=1}^{p} \beta_j^2 \right\} \quad (4)$$

Elastic net constraint regions

Below are some representations of $p = 2$ elastic net constraint regions for $t = 3$ (made using some code of Prof. Huaiqing Wu) that clearly show the compromise nature of the elastic net.



Elastic net estimated df



Several sources suggest that a modification of the elastic net idea, namely

$$(1+\lambda_2)\,\widehat{\boldsymbol{\beta}}^{\text{elastic net}}_{\lambda_1,\lambda_2}$$

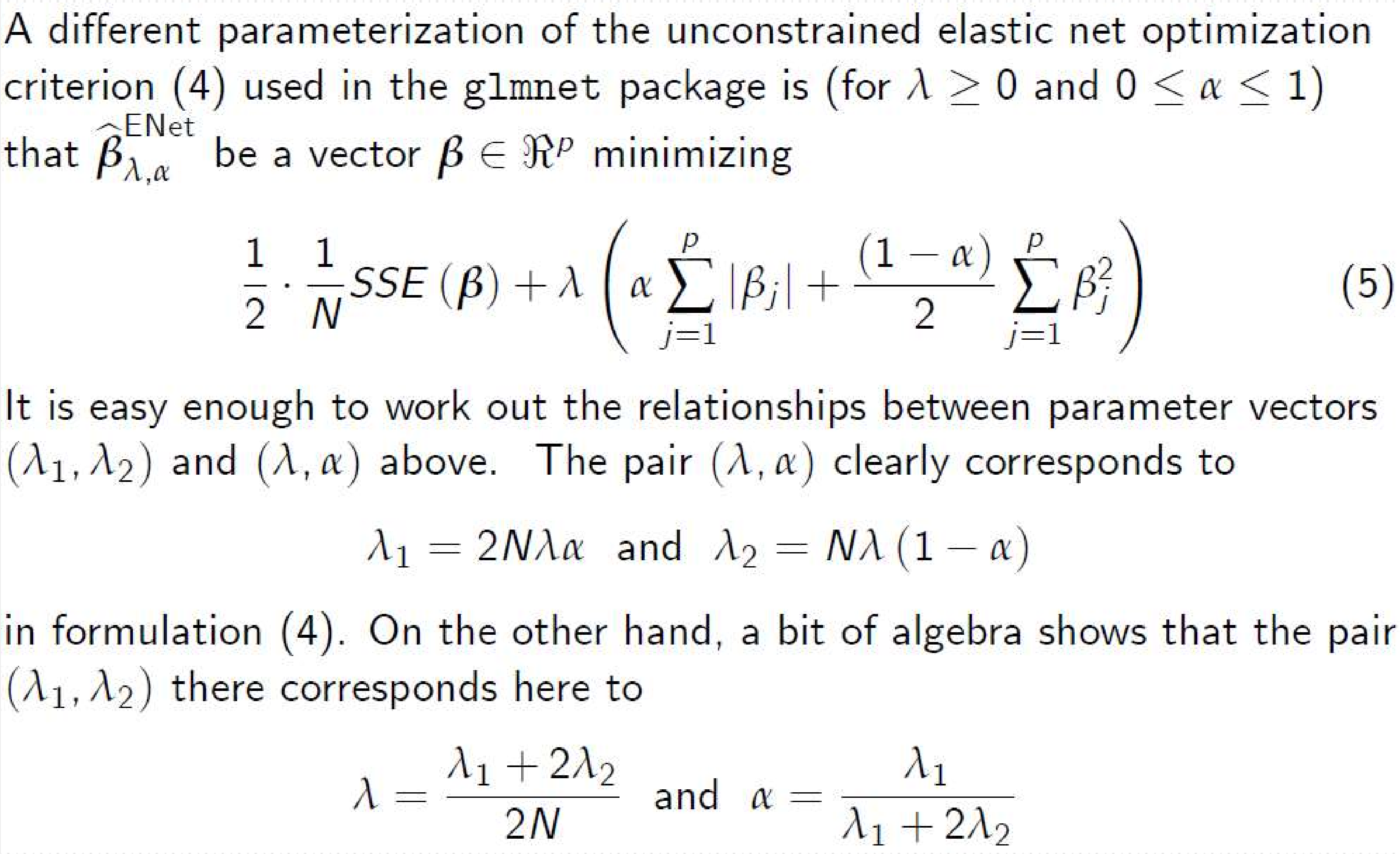
performs better than the original version.

For $\widehat{\boldsymbol{\beta}}^{\text{elastic net}}_{\lambda_1,\lambda_2}$ with r non-zero components and $\mathbf{X}_*$ made up of the corresponding columns of $\mathbf{X}$, estimated effective degrees of freedom for the unmodified form of the elastic net are

$$\text{df}\,\widehat{(\lambda_1,\lambda_2)} = \text{tr}\left(\mathbf{X}_*\left(\mathbf{X}_*'\mathbf{X}_* + \lambda_2\mathbf{I}\right)^{-1}\mathbf{X}_*'\right) = \sum_{j=1}^{r}\left(\frac{d_j^2}{d_j^2+\lambda_2}\right)$$

(for d_j's the singular values of $\mathbf{X}_*$). The modified form has estimated effective degrees of freedom $(1+\lambda_2)$ times this value.

`glmnet` elastic net parameterization



A different parameterization of the unconstrained elastic net optimization criterion (4) used in the `glmnet` package is (for $\lambda \geq 0$ and $0 \leq \alpha \leq 1$) that $\widehat{\boldsymbol{\beta}}_{\lambda,\alpha}^{\mathrm{ENet}}$ be a vector $\boldsymbol{\beta} \in \Re^p$ minimizing

$$\frac{1}{2} \cdot \frac{1}{N} SSE\left(\boldsymbol{\beta}\right) + \lambda \left(\alpha \sum_{j=1}^{p} |\beta_j| + \frac{(1-\alpha)}{2} \sum_{j=1}^{p} \beta_j^2 \right) \qquad (5)$$

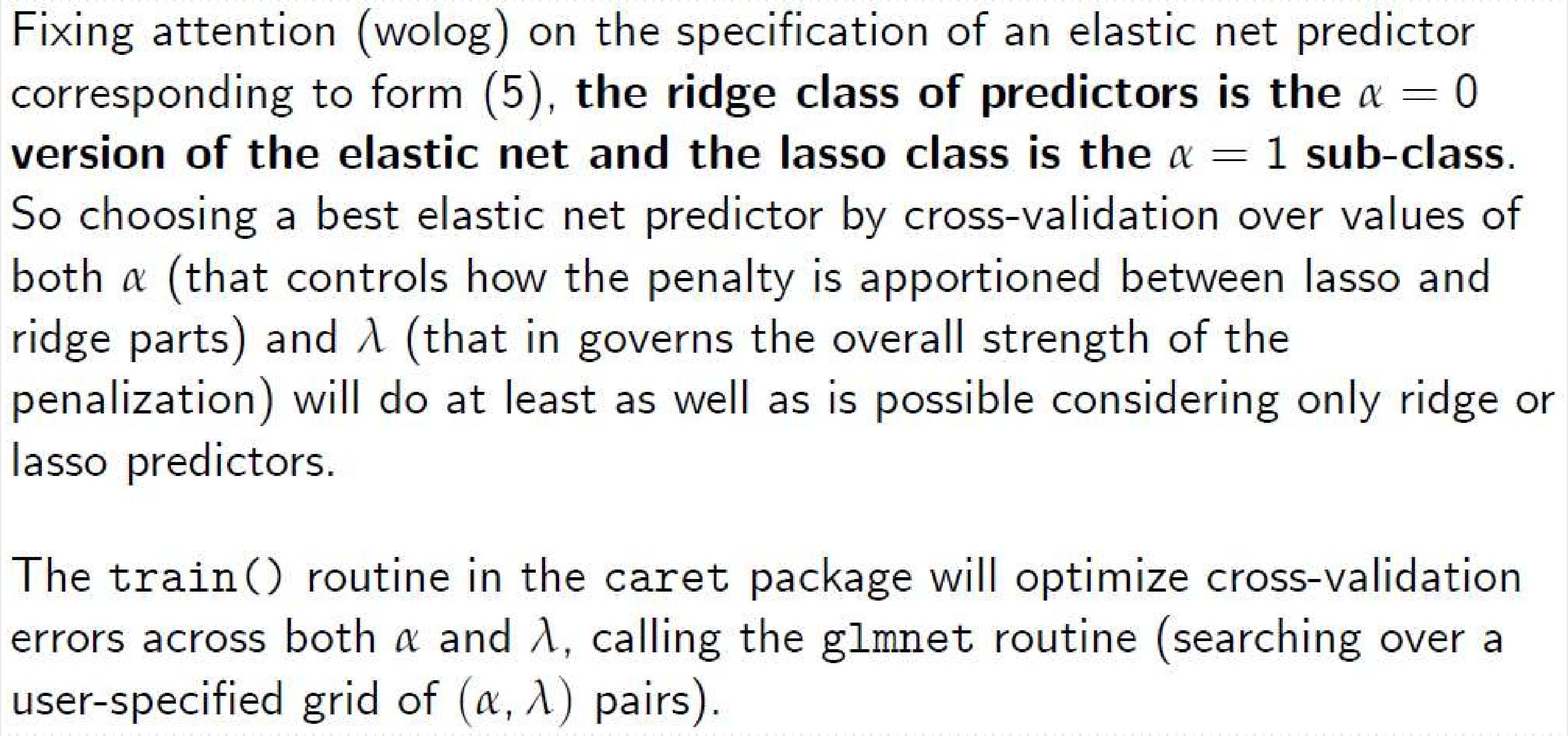
It is easy enough to work out the relationships between parameter vectors (λ_1, λ_2) and (λ, α) above. The pair (λ, α) clearly corresponds to

$$\lambda_1 = 2N\lambda\alpha \quad \text{and} \quad \lambda_2 = N\lambda\left(1-\alpha\right)$$

in formulation (4). On the other hand, a bit of algebra shows that the pair (λ_1, λ_2) there corresponds here to

$$\lambda = \frac{\lambda_1 + 2\lambda_2}{2N} \quad \text{and} \quad \alpha = \frac{\lambda_1}{\lambda_1 + 2\lambda_2}$$

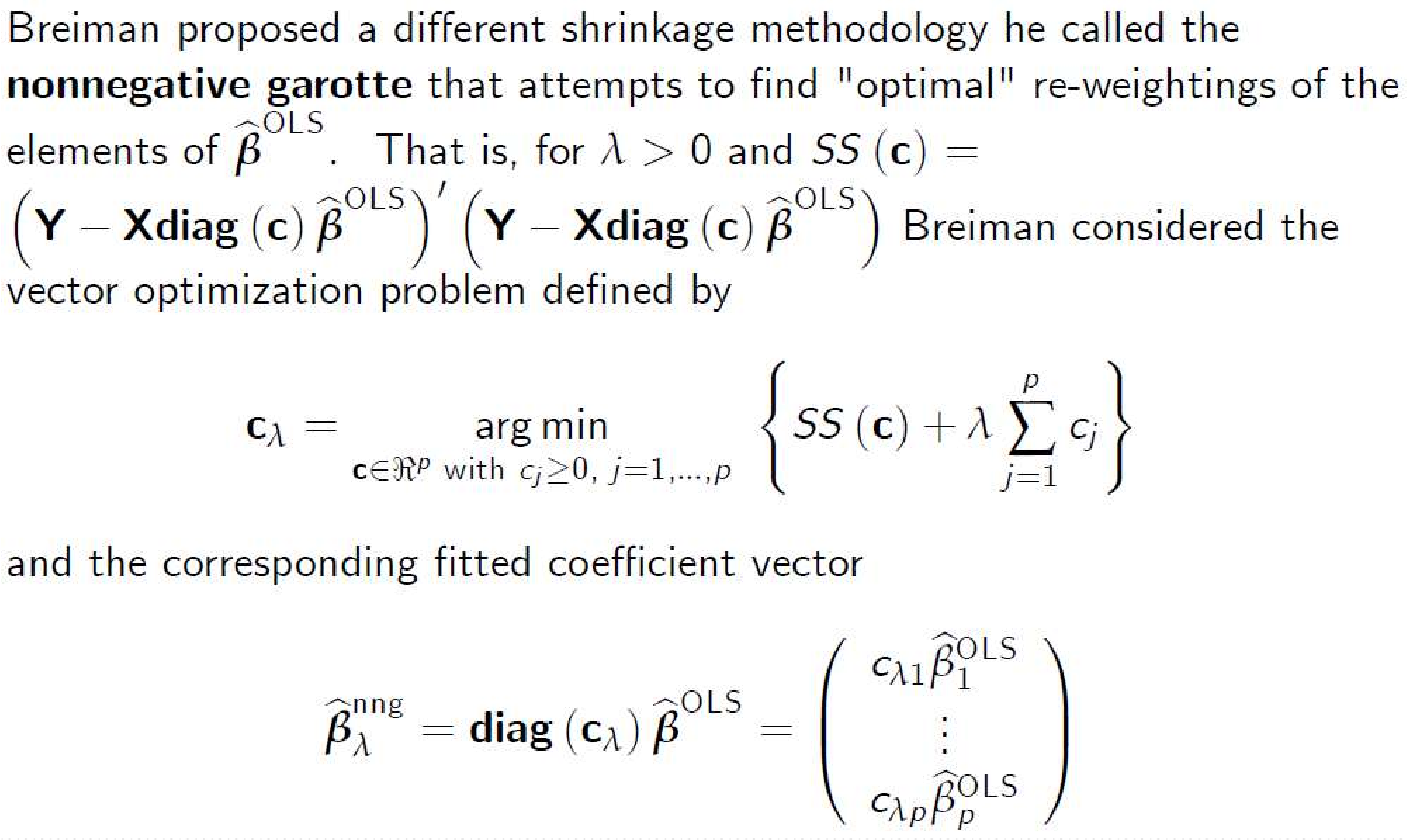
Cross-validation choice of e-net parameters



Fixing attention (wolog) on the specification of an elastic net predictor corresponding to form (5), **the ridge class of predictors is the $\alpha = 0$ version of the elastic net and the lasso class is the $\alpha = 1$ sub-class**. So choosing a best elastic net predictor by cross-validation over values of both α (that controls how the penalty is apportioned between lasso and ridge parts) and λ (that in governs the overall strength of the penalization) will do at least as well as is possible considering only ridge or lasso predictors.

The `train()` routine in the `caret` package will optimize cross-validation errors across both α and λ, calling the `glmnet` routine (searching over a user-specified grid of (α, λ) pairs).

Nonnegative garrote



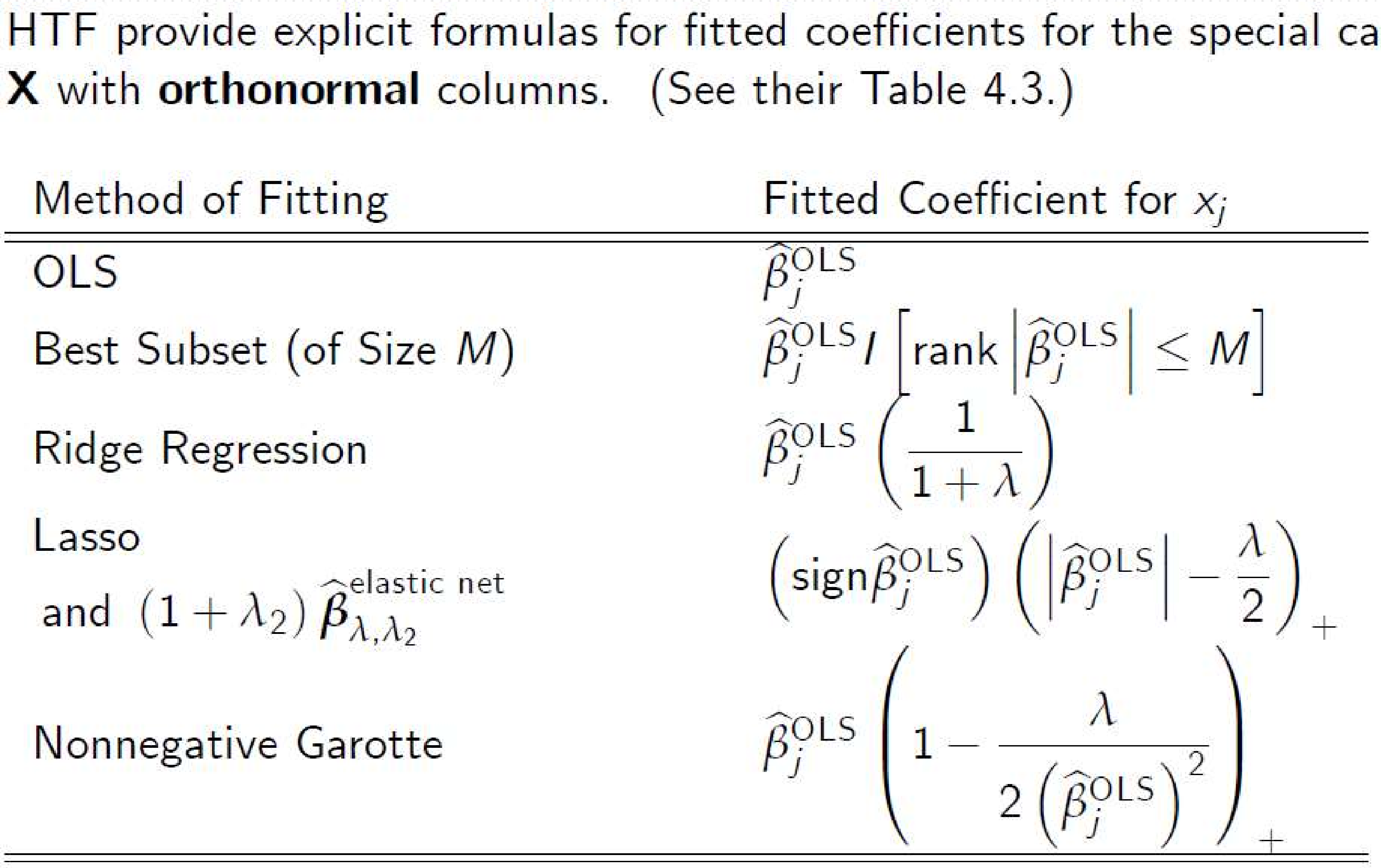
Breiman proposed a different shrinkage methodology he called the **nonnegative garotte** that attempts to find "optimal" re-weightings of the elements of $\widehat{\beta}^{\text{OLS}}$. That is, for $\lambda > 0$ and $SS(\mathbf{c}) = \left(\mathbf{Y} - \mathbf{X}\mathbf{diag}(\mathbf{c})\,\widehat{\beta}^{\text{OLS}}\right)'\left(\mathbf{Y} - \mathbf{X}\mathbf{diag}(\mathbf{c})\,\widehat{\beta}^{\text{OLS}}\right)$ Breiman considered the vector optimization problem defined by

$$\mathbf{c}_\lambda = \underset{\mathbf{c}\in\Re^p \text{ with } c_j\geq 0,\ j=1,\ldots,p}{\arg\min} \left\{ SS(\mathbf{c}) + \lambda \sum_{j=1}^{p} c_j \right\}$$

and the corresponding fitted coefficient vector

$$\widehat{\beta}_\lambda^{\text{nng}} = \mathbf{diag}(\mathbf{c}_\lambda)\,\widehat{\beta}^{\text{OLS}} = \begin{pmatrix} c_{\lambda 1}\widehat{\beta}_1^{\text{OLS}} \\ \vdots \\ c_{\lambda p}\widehat{\beta}_p^{\text{OLS}} \end{pmatrix}$$

Comparison of shrinkage method coefficients

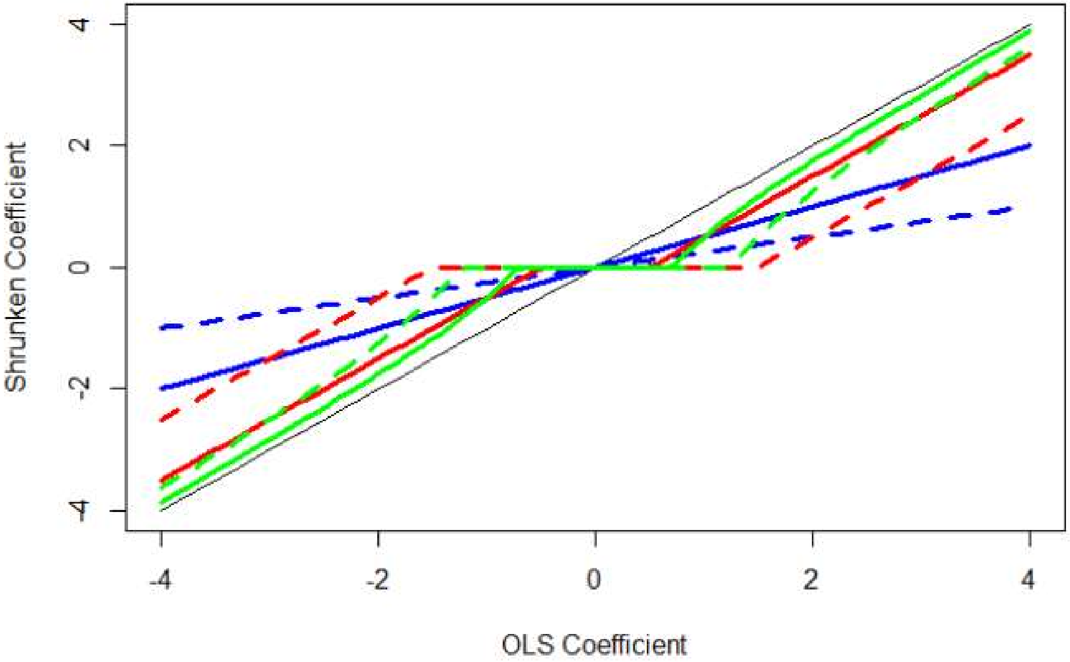


HTF provide explicit formulas for fitted coefficients for the special case of **X** with **orthonormal** columns. (See their Table 4.3.)

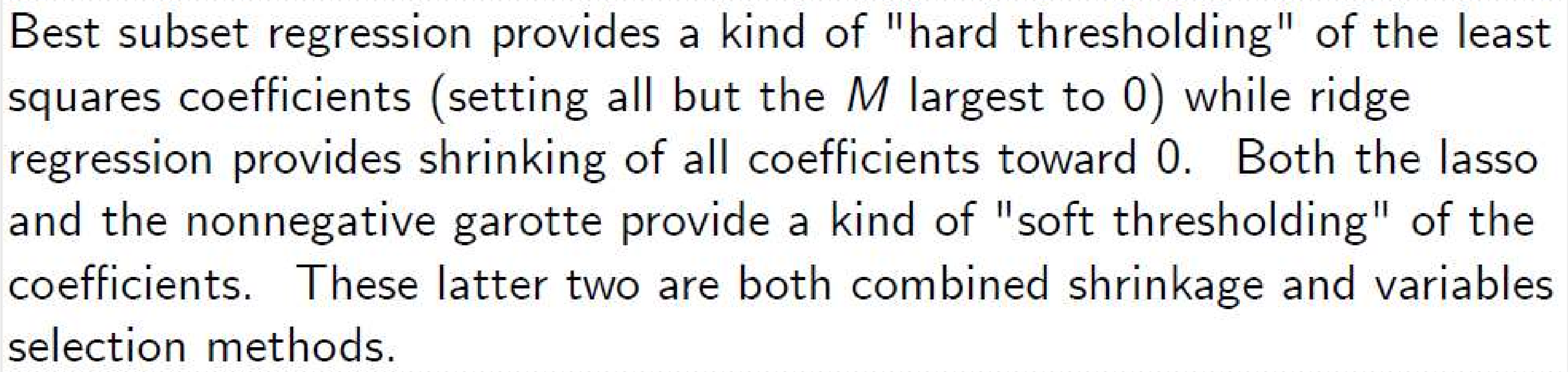
Method of Fitting	Fitted Coefficient for x_j
OLS	$\widehat{\beta}_j^{\text{OLS}}$
Best Subset (of Size M)	$\widehat{\beta}_j^{\text{OLS}} I\left[\text{rank}\left\lvert\widehat{\beta}_j^{\text{OLS}}\right\rvert \leq M\right]$
Ridge Regression	$\widehat{\beta}_j^{\text{OLS}}\left(\frac{1}{1+\lambda}\right)$
Lasso and $(1+\lambda_2)\,\widehat{\boldsymbol{\beta}}_{\lambda,\lambda_2}^{\text{elastic net}}$	$\left(\text{sign}\widehat{\beta}_j^{\text{OLS}}\right)\left(\left\lvert\widehat{\beta}_j^{\text{OLS}}\right\rvert - \frac{\lambda}{2}\right)_+$
Nonnegative Garotte	$\widehat{\beta}_j^{\text{OLS}}\left(1 - \frac{\lambda}{2\left(\widehat{\beta}_j^{\text{OLS}}\right)^2}\right)_+$

Comparison of shrinkage methods

The figure below provides plots of the functions (in the previous table) of OLS coefficients giving ridge (blue), lasso (red), and nonnegative garotte (green) coefficients for the "orthonormal predictors" case. (Solid lines are $\lambda = 1$ plots and dotted ones are for $\lambda = 3$.)



Interpretation of comparisons



Best subset regression provides a kind of "hard thresholding" of the least squares coefficients (setting all but the M largest to 0) while ridge regression provides shrinking of all coefficients toward 0. Both the lasso and the nonnegative garotte provide a kind of "soft thresholding" of the coefficients. These latter two are both combined shrinkage and variables selection methods.

Extensions/generalizations of lasso

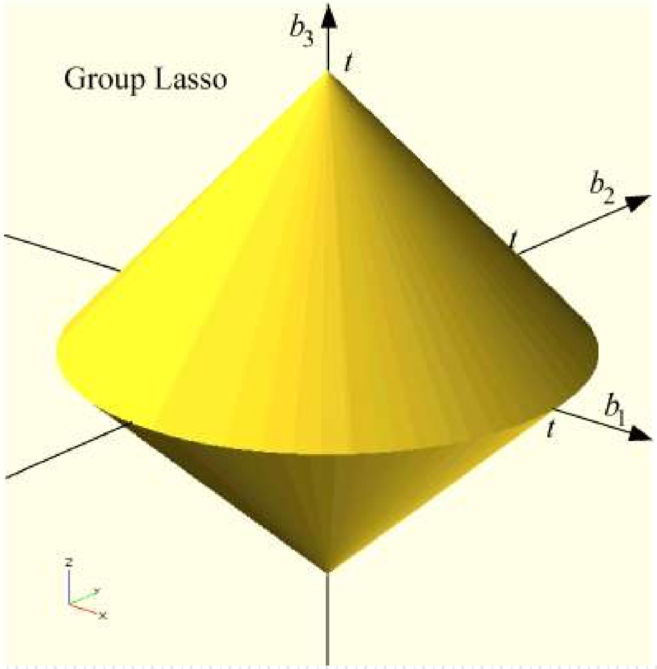
Many lasso-like penalized least squares methods have been suggested, tailored to various special circumstances. Notable are "group lasso," "sparse group lasso," and "fused lasso" methods. To indicate what has been proposed, we'll illustrate the (2-) group lasso. If for some reason the coordinates of $\mathbf{x} \in \Re^p$ break naturally into 2 groups (say the first l and last $p-l$ coordinates of $\mathbf{x}$), for a $\lambda > 0$, a "group lasso" coefficient vector is

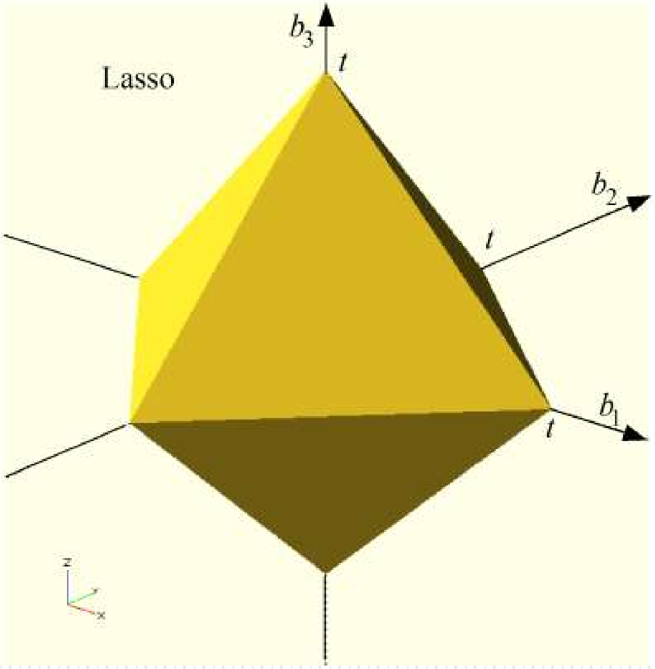
$$\widehat{\beta}_\lambda^{\text{group lasso}} = \underset{\beta \in \Re^p}{\arg\min} \left\{ (\mathbf{Y} - \mathbf{X}\beta)' (\mathbf{Y} - \mathbf{X}\beta) + \lambda \left(\sqrt{\sum_{j=1}^{l} \beta_j^2} + \sqrt{\sum_{j=l+1}^{p} \beta_j^2} \right) \right\}$$

Of course, there can be more than 2 groups, and when each group is of size 1 this reduces to the simple Lasso. The geometry of constraint regions associated with this methodology suggests why it tends to "zero-out" coefficients in groups associated with the penalty.

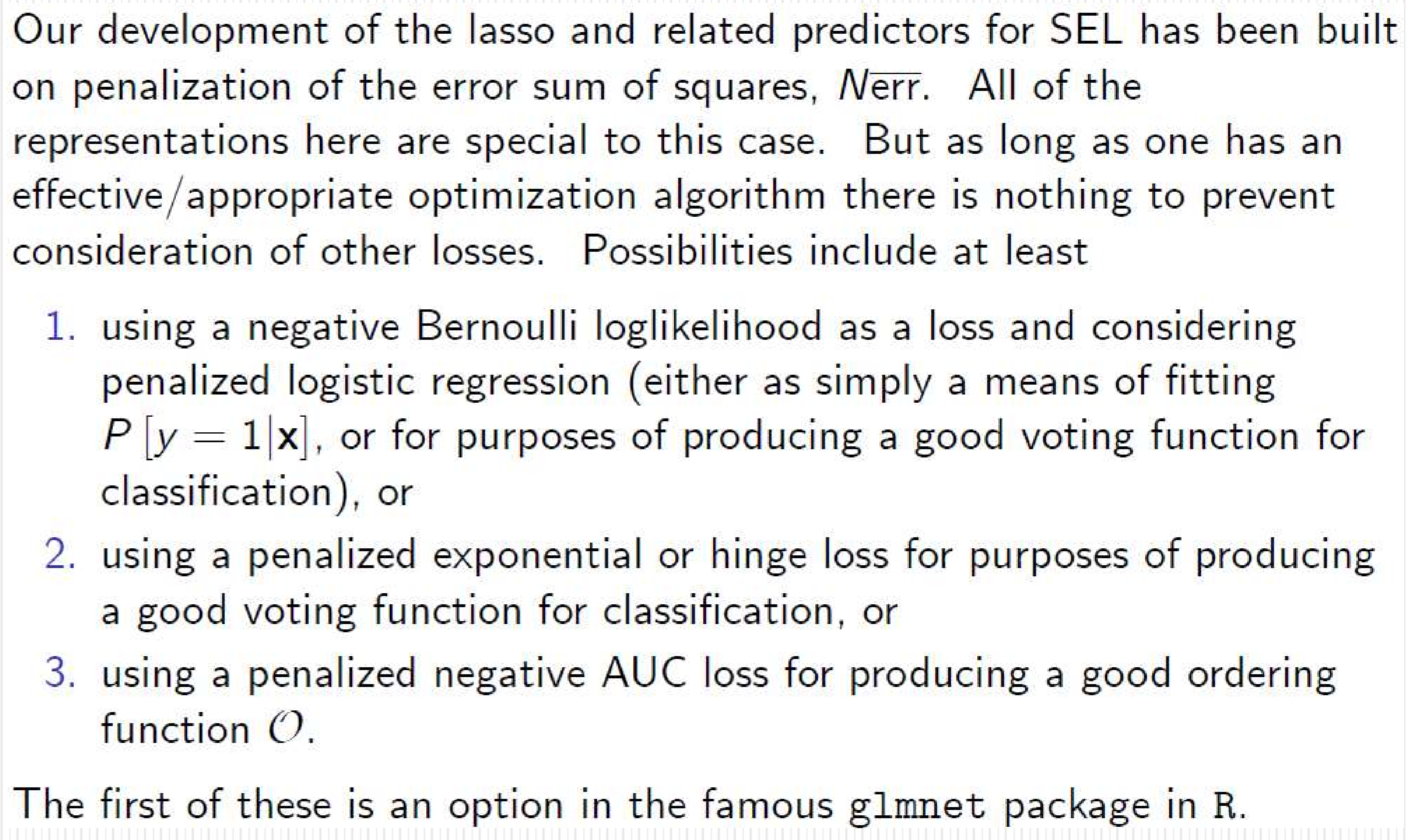
Geometry of group lasso constraint region

Here is a representation of a $p = 3$ constraint region associated with a grouped lasso where coordinates 1 and 2 of $\mathbf{x}$ are grouped separate from coordinate 3. The corresponding lasso region is shown for comparison purposes.





Other problems/losses (beyond SEL)



Our development of the lasso and related predictors for SEL has been built on penalization of the error sum of squares, $N\overline{\text{err}}$. All of the representations here are special to this case. But as long as one has an effective/appropriate optimization algorithm there is nothing to prevent consideration of other losses. Possibilities include at least

1. using a negative Bernoulli loglikelihood as a loss and considering penalized logistic regression (either as simply a means of fitting $P\left[y=1|\mathbf{x}\right]$, or for purposes of producing a good voting function for classification), or
2. using a penalized exponential or hinge loss for purposes of producing a good voting function for classification, or
3. using a penalized negative AUC loss for producing a good ordering function $\mathcal{O}$.

The first of these is an option in the famous `glmnet` package in `R`.