Bicoordinate Descent for the LASSO*

In Song Kim† John Londregan‡ Marc Ratkovic§

Version 0.5 – July 28, 2015

Abstract

We propose an estimator for the LASSO that iteratively optimizes the coefficients in pairs. In addition to improving efficiency by coordinating the updates of the the paired variables, our algorithm affords insights into the nature of the LASSO problem. Our method outperforms the popular `glmnet` algorithm in all but high-K low-N settings, executing increasingly better as N increases.

Key Words: variable selection, LASSO

1 Introduction

We offer an improvement to the coordinate wise descent method for estimating the LASSO pioneered by Tibshirani (1996), Fu (1998), and by Friedman, Hastie and Tibshirani (2010a). Our bicoordinate descent method generalizes the one parameter at a time soft thresholding embodied in Fu's "shooting algorithm" by updating the parameter values in pairs. When the regressors are orthogonal our algorithm coincides with one coordinate at a time soft thresholding, but when the explanators are correlated our algorithm coordinates the simultaneous adjustment of the coefficients to update the coefficient estimates more efficiently in pairs. The results is a substantial reduction in the number of passes through the data that the algorithm takes on its path to convergence. The time required by our method for each pass through the data increases relative to unicoordinate descent by much less than the number of passes falls, resulting in an overall improvement in the time to convergence.

*We thank Princeton University and the Universidad de Desarollo for financial support. The proposed methods can be implemented via the open-source statistical software, `bcd: Bicoordinate Descent for the LASSO`, available through the Comprehensive R Archive Network (http://cran.r-project.org/package=bcd).

†Assistant Professor, Department of Political Science, Massachusetts Institute of Technology, Cambridge, MA, 02139. Email: insong@mit.EDU, URL: http://web.mit.edu/insong/www/

‡Professor of Politics and International Affairs, Woodrow Wilson School, Princeton University, Princeton NJ 08544. Phone: 609-258-4854, Email: jbl@princeton.edu

§Assistant Professor, Department of Politics, Princeton University, Princeton NJ 08544. Phone: 608-658-9665, Email: ratkovic@princeton.edu, URL: http://www.princeton.edu/~ratkovic

Our exposition proceeds as follows. The next section introduces our bicoordinate descent algorithm for LASSOed regression, and provides graphical intuition about its workings. Proofs are provided in Appendix A. The subsequent section discusses some algorithmic adaptations that accelerate computation. In the subsequent section we we provide some results comparing the computational speed of our algorithm with that of the standard `glmnet` software. In the following section we extend the model, to encompass the probit, using the EM algorithm to create an interface between the bicoordinate descent solution for the least squares problem and the nonlinear nature of the probit model[1]. A final section concludes and discusses ongoing directions of research. The open source software, `bcd: Bicoordinate Descent for the LASSO`, for fitting the proposed method is available through the Comprehensive R Archive Network (http://cran.r-project.org/package=bcd).

2 Estimating the LASSO

Tibshirani (1996) promulgated the LASSO model as a practical sparse estimator. He noted that in the special case of orthogonal regressors a remarkably straightforward solution can be found by individually soft thresholding each of the estimated coefficients. Fu (1998) developed a "shooting" algorithm that generalizes this approach to any set of regressors–at each pass through the data the algorithm successively updates the parameters one at a time using soft thresholding. Convergence of Fu's algorithm is quick, and Friedman, Hastie and Tibshirani (2010a), hereafter "FHT", make the algorithm even faster by arraying solutions to a sequence of LASSO problems in a trellis, which they refer to as a "regularization path", in which they use each solution as a starting value for the next problem. Their unicoordinate descent algorithm, which they supplement with a brace of best programming practices, has defined the computational frontier for the LASSO model. Our primary departure from the FHT algorithm is to update the parameters in pairs, exploiting correlations among the explanatory variables to generate a more direct pathway to the solution.

2.1 Formalizing the Algorithm

Starting with data of the form $\left\{Y_i, \{X_{ij}\}_{j=1}^k\right\}_{i=1}^n$ we first center the observations, and normalize the l^2 norm of each of the explanators to equal one, leaving us with: $\left\{y_i, \{x_{ij}\}_{j=1}^k\right\}_{i=1}^n$ satisfying $\sum_{i=1}^n y_i = 0$, and for each $j \in \{1, ..., k\}$ we also have $\sum_{i=1}^n x_{ij} = 0$, and $\sum_{i=1}^n x_{ij}^2 = 1$. If any pairs of explanators are perfectly correlated we arbitrarily remove one element of the perfectly correlated pair, until no

[1] We defer presenting some analytical results for weighted least squares to an appendix.

perfectly correlated pairs of explanators remain[2].

The LASSO estimator introduced by Tibshirani (1996) is the solution to a problem of the form:

$$P1: \min_{\{\beta_j\}_{j=1}^k} RSS\left(\{\beta_j\}_{j=1}^k | \left\{\{y_i, \{x_{ij}\}_{j=1}^k\right\}_{i=1}^n\right) \text{ subject to } \sum_{j=1}^k |\beta_j| \leq t \tag{1}$$

where:

$$RSS\left(\{\beta_j\}_{j=1}^k | \left\{y_i, \{x_{ij}\}_{j=1}^k\right\}_{i=1}^n\right) = \sum_{i=1}^n \left(y_i - \sum_{j=1}^k \beta_j x_{ij}\right)^2 \tag{2}$$

Next, recalling that we have culled all of the perfectly correlated observations, let's arrange our data into $C = \left\lfloor \frac{k}{2} \right\rfloor$ pairs, indexed by $c \in \{1, ..., C\}$, with at most one singleton observation which remains when k is odd.

Now suppose that we take successive passes through the data. At iteration s we turn to each pair of coefficients in turn, taking the others as given at their current values. We seek to minimize the constrained residual sum of squares with respect to $\{\beta_{2c-1}, \beta_{2c}\}$ only, while of course continuing to satisfy the constraint. We can formalize this problem as:

$$P2_c^s: \min_{\beta_{2c-1}, \beta_{2c}} RSS_c\left(\beta_{2c-1}, \beta_{2c} | \left\{v_{ic}^s, x_{i,2c-1}, x_{i,2c}\right\}_{i=1}^n\right) \text{ subject to } |\beta_{2c-1}| + |\beta_{2c}| \leq \theta_c^s$$

where:

$$RSS_c\left(\beta_{2c-1}, \beta_{2c} | \left\{v_{ic}^s, x_{i,2c-1}, x_{i,2c}\right\}_{i=1}^n\right) = \sum_{i=1}^n (v_{ic}^s - \beta_{2c-1} x_{i,2c-1} - \beta_{2c} x_{i,2c})^2$$

and:

$$v_{ic}^s = \left(y_i - \sum_{j<2c-1} \beta_j^{s,lasso} x_{ij} - \sum_{2c<j} \beta_j^{s-1,lasso} x_{ij}\right) \quad \text{while} \quad \theta_c^s = t - \sum_{j<2c-1} |\beta_j^{s,lasso}| - \sum_{2c<j} |\beta_j^{s-1,lasso}|$$

We denote the solutions to $P2_c^s$ by $(\beta_{2c-1}^{s,lasso}, \beta_{2c}^{s,lasso})$.

Finally, if k is odd, there remains a singleton observation that is not encompassed by any of the pairs. Define:

$$P3^s: \min_{\beta_k} \sum_{i=1}^n \left(v_{ik}^s - \beta_k x_{i,k}\right)^2 \text{ subject to } |\beta_k| \leq \theta_p^s$$

[2]Of course for each perfectly correlated pair for which at least one element is selected by the LASSO there will in general be a continuum of equivalent solutions to our problem.

where:

$$v_{ik}^s = \left(y_i - \sum_{j<k} \beta_j^{s,lasso} x_{ij}\right) \text{ and } \theta_k^s = t - \sum_{j<k} |\beta_j^{s,lasso}|$$

and we denote the solutions to $P3^s$ by $\beta_k^{s,lasso}$.

2.2 The Bicoordinate Descent Algorithm

Our algorithm for calculating $(\beta_{2c-1}^{s,lasso}, \beta_{2c}^{s,lasso})$ proceeds as follows. Let $(\beta_{2c-1}^{s,ols}, \beta_{2c}^{s,ols})$ solve:

$$POLS_c^s: \quad \min_{\beta_{2c-1}^*, \beta_{2c}^*} RSS_c\left(\beta_{2c-1}^*, \beta_{2c}^* | \left\{v_{ic}^s, x_{i,2c-1}, x_{i,2c}\right\}_{i=1}^n\right)$$

so:

$$\begin{pmatrix} \beta_{2c-1}^{s,ols} \\ \beta_{2c}^{s,ols} \end{pmatrix} = \frac{1}{1 - R_c^2} \begin{pmatrix} \sum_{i=1}^n v_{ic}^s (x_{i,2c-1} - R_c x_{i,2c}) \\ \sum_{i=1}^n v_{ic}^s (x_{i,2c} - R_c x_{i,2c-1}) \end{pmatrix}$$

where:

$$R_c = \sum_{i=1}^n x_{i,2c-1} x_{i,2c} \tag{3}$$

Next define:

$$R_c^{s*} = sign(\beta_{2c-1}^{s,ols}) \times sign(\beta_{2c}^{s,ols}) \times R_c \tag{4}$$

We let λ denote the Lagrange multiplier associated with the constraint in $P1$. We will treat this as a "tuning parameter" shared by all of the $P2_c^s$.

Couched in terms of λ, when:

$$\frac{\lambda}{2(1 + R_c^{s*})} < \min\{|\beta_{2c-1}^{s,ols}|, |\beta_{2c}^{s,ols}|\} \tag{5}$$

our estimates are calculated as:

$$\begin{pmatrix} \beta_{2c-1}^{s,lasso} \\ \beta_{2c}^{s,lasso} \end{pmatrix} = \begin{pmatrix} sign(\beta_{2c-1}^{s,ols})\left(|\beta_{2c-1}^{s,ols}| - \frac{\lambda}{2(1+R_c^{s*})}\right) \\ sign(\beta_{2c}^{s,ols})\left(|\beta_{2c}^{s,ols}| - \frac{\lambda}{2(1+R_c^{s*})}\right) \end{pmatrix} \tag{6}$$

When condition (5) fails, but

$$|\beta_{2c-1}^{s,ols}| > |\beta_{2c}^{s,ols}| \tag{7}$$

then the update step for $(\beta_{2c-1}^{s,lasso}, \beta_{2c}^{s,lasso})$ is:

$$\left(\beta_{2c-1}^{s,lasso}, \beta_{2c}^{s,lasso}\right) = sign(\beta_{2c-1}^{s,ols}) max\left\{|\beta_{2c-1}^{s,ols}| + R_{2c}^{*s}|\beta_{2c}^{s,ols}| - \frac{\lambda}{2}, 0\right\} \tag{8}$$

whereas if (5) fails but the inequality in condition (7) is reversed, then:

$$\left(\beta_{2c-1}^{s,lasso}, \beta_{2c}^{s,lasso}\right) = sign(\beta_{2c}^{s,ols}) max\left\{0, |\beta_{2c}^{s,ols}| + R_{c}^{*s}|\beta_{2c-1}^{s,ols}| - \frac{\lambda}{2}\right\} \tag{9}$$

Notice that when $R_c^{s*} = 0$ the bicoordinate descent algorithm coincides with the soft thresholding embodied in the "shooting" algorithm of Fu (1998).

Of course, the solution to $P3^s$ is simply given by the soft thresholding result returned by Fu's algorithm:

$$\beta_p^{s,lasso} = sign(\beta_p^{s,ols}) max\left\{|\beta_p^{s,ols}| - \frac{\lambda}{2}, 0\right\} \tag{10}$$

where:

$$\beta_p^{s,ols} = \sum_{i=1}^{n} v_{ic}^s x_{ik}$$

2.3 Why it Works

Let's take a closer look at the objective function for $P2_c^s$. First it's useful to define a few terms. For comparison let's start with the unconstrained sum of squared errors:

$$sse_0^{c,s} = \sum_{i=1}^{n}\left(v_{ic}^s - \beta_{2c-1}^{s,ols} x_{i,2c-1} - \beta_{2c}^{s,ols} x_{i,2c}\right)^2 \tag{11}$$

Next consider the following quadratic function $Q(\beta_{2c-1} - \beta_{2c-1}^{s,ols}, \beta_{2c} - \beta_{2c}^{s,ols}, R_c^{s*})$:

$$Q(\beta_{2c-1} - \beta_{2c-1}^{s,ols}, \beta_{2c} - \beta_{2c}^{s,ols}, R_c^{s*}) = (\beta_{2c-1} - \beta_{2c-1}^{s,ols}, \beta_{2c} - \beta_{2c}^{s,ols}) \begin{pmatrix} 1 & R_c^{s*} \\ R_c^{s*} & 1 \end{pmatrix} \begin{pmatrix} \beta_{2c-1} - \beta_{2c-1}^{s,ols} \\ \beta_{2c} - \beta_{2c}^{s,ols} \end{pmatrix} \tag{12}$$

It turns out that we can reconceive the objective function for $P2_c^s$ in terms of Q. We state this formally as:

Lemma 1: $RSS_c\left(\beta_{2c-1}, \beta_{2c} | \left\{v_{ic}^s, x_{i,2c-1}, x_{i,2c}\right\}_{i=1}^n\right) = sse_0^{c,s} + Q(\beta_{2c-1} - \beta_{2c-1}^{s,ols}, \beta_{2c} - \beta_{2c}^{s,ols}, R_c^{s*})$

Proof of Lemma 1: See the appendix.

Notice that $sse_0^{c,s}$ is constant with respect to β_{2c-1} and β_{2c}, so Lemma 1 allows us to reformulate $P2_c^s$ as a quadratic programming problem:

$$P2_c^{s\prime}: \min_{\beta_{2c-1}, \beta_{2c}} Q(\beta_{2c-1} - \beta_{2c-1}^{s,ols}, \beta_{2c} - \beta_{2c}^{s,ols}, R_c^{s*}) \quad \text{subject to} \quad |\beta_{2c-1}| + |\beta_{2c}| \leq \theta_c^s$$

The constraint is in the form of a diamond, while the level sets of the objective function for $P2_c^{s\prime}$ are ellipses. This is illustrated in the lefthand panel of figure 1, where the hollow dot corresponds to $(\beta_{2c-1}^{s,ols}, \beta_{2c}^{s,ols})$ while $(\beta_{2c-1}^{s,lasso}, \beta_{2c}^{s,lasso})$ is represented by the solid dot.

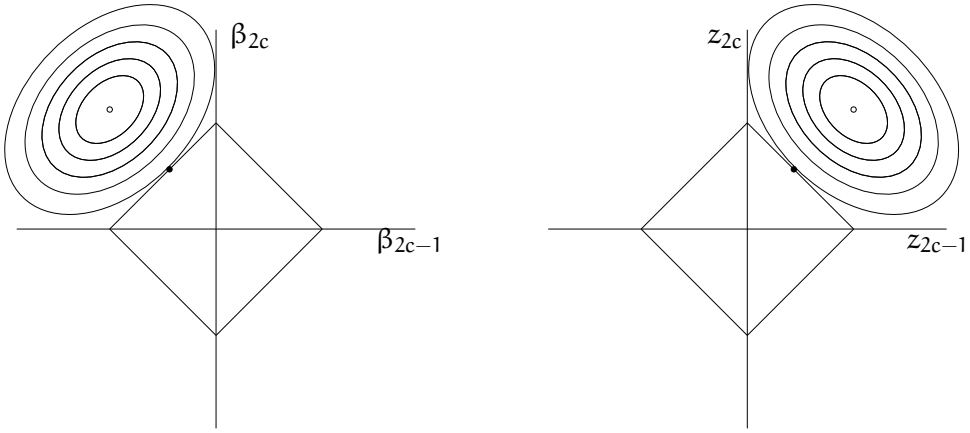


Figure 1: Optimization $P2_c^{s\prime}$ *vs* PZ

There is an isomorphic relationship amongst solutions in distinct quadrants. To see this, define $\delta_j^s \equiv sign(\beta_j^{s,ols})$ and let $z_j^s = \delta_j^s \beta_j$, and then rewrite problem $P2_c^{s\prime}$ as:

$$\min_{z_{2c-1}, z_{2c}} Q\left(\delta_{2c-1}^s z_{2c-1} - \delta_{2c-1}^s |\beta_{2c-1}^{s,ols}|, \delta_{2c}^s z_{2c} - \delta_{2c}^s |\beta_{2c}^{s,ols}|, R_c^{s*}\right) \quad \text{subject to} \quad |z_{2c-1}| + |z_{2c}| \leq \theta_c^s$$

recalling our definition of R_c^{s*} from expression (4) this can be reexpressed as:

$$PZ: \min_{z_{2c-1}, z_{2c}} Q\left(z_{2c-1} - |\beta_{2c-1}^{s,ols}|, z_{2c} - |\beta_{2c}^{s,ols}|, R_c^{s*}\right) \text{ subject to } |z_{2c-1}| + |z_{2c}| \leq \theta_c^s$$

If $(\hat{z}_{2c-1}^s, \hat{z}_{2c}^s)$ solves PZ then

$$(\beta^{s,lasso}_{2c-1}, \beta^{s,lasso}_{2c}) = (\delta^{s}_{2c-1}\hat{z}^{s}_{2c-1}, \delta^{s}_{2c}\hat{z}^{s}_{2c}) \tag{13}$$

is a solution to $P2^{s\prime}_c$. The righthand panel of figure 1 depicts the reformulation of $P2^{s\prime}_c$ as PZ. The orientation of the ellipse shifts with the translation to the first quadrant, this corresponds to the change from R^s_c to R^{s*}_c. The hollow dot in the right panel corresponds to $(|\beta^{s,ols}_{2c-1}|, |\beta^{s,ols}_{2c}|)$ whereas the solid dot indicates the solution values (z^s_{2c-1}, z^s_{2c}) for PZ.

As it transpires the solution to PZ is non-negative. In fact, if the constraint binds the solution is to be found along the first quadrant simplex $\Delta(\theta^s_c)$:

$$\Delta(\theta^s_c) = \left\{(z^s_{2c-1}, z^s_{2c}) | z^s_{2c-1} + z^s_{2c} = \theta^s_c \text{ and } z_{2c-1} \geq 0 \text{ and } z_{2c} \geq 0\right\} \tag{14}$$

We state this important result as:

Lemma 2: The solutions to PZ satisfy $\hat{z}_{2c-1} \geq 0$ and $\hat{z}_{2c} \geq 0$, while for $\theta^s_c \leq |\beta^{OLS}_{2c-1}| + |\beta^{OLS}_{2c}|$, $(\hat{z}^s_{2c-1}, \hat{z}^s_{2c}) \in S$.

Proof: See the appendix.

Now let's take a graphical approach to the solution. Consider the objective function:

$$Q\left(z^s_{2c-1} - |\beta^{s,ols}_{2c-1}|, z^s_{2c} - |\beta^{s,ols}_{2c}|, R^{s*}_c\right) \tag{15}$$

for PZ. The lefthand panel of figure 2 shows the level curves for Q. At an interior solution for (z^s_{2c-1}, z^s_{2c}) the highest level curve that makes contact with the constraint will be tangent to $\Delta(\theta^s_c)$, and so it will have the same slope, -1, as the simplex, several points at which the slope of a level curve matches -1 are depicted in the lefthand panel of figure 2. The level curve slopes are given by:

$$\frac{dz^s_{2c}}{dz^s_{2c-1}} = -\frac{\frac{\partial Q}{\partial z_{2c-1}}}{\frac{\partial Q}{\partial z_{2c}}} \tag{16}$$

Setting this slope to -1 and solving we recover the locus of points at which the level curves of Q share the same slope as $\Delta(\theta^s_c)$, see the central panel of figure 2:

$$z^s_{2c} = |\beta^{s,ols}_{2c}| - |\beta^{s,ols}_{2c-1}| + z^s_{2c-1} \tag{17}$$

Putting this formally, we have:

Lemma 3: The locus of points at which the level curves of Q share the same slope as $\Delta(\theta^s_c)$ is given by (17).

Proof: See the appendix.

If the line (17) intersects $\Delta(\theta_c^s)$ we have a tangency solution for PZ, such a solution is depicted in the righthand panel of figure 2, where it corresponds to the solid dot whose coordinates are given by:

$$(z_{2c-1}^s, z_{2c}^s) = \left(\frac{\theta_c^s + |\beta_{2c-1}^{s,ols}| - |\beta_{2c}^{s,ols}|}{2}, \frac{\theta_c^s + |\beta_{2c}^{s,ols}| - |\beta_{2c-1}^{s,ols}|}{2} \right) \tag{18}$$

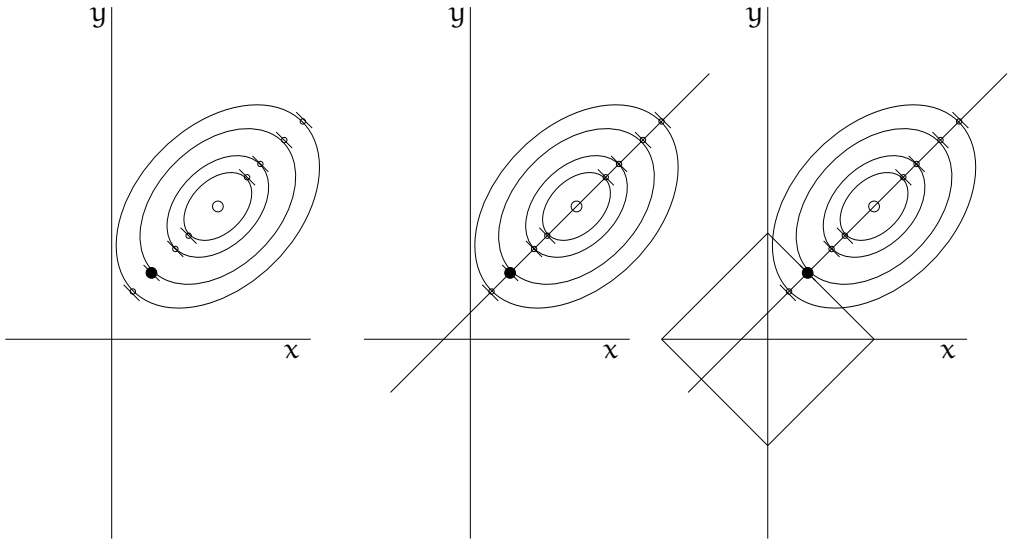


Figure 2: Left: Tangencies Center: Locus of Tangencies Right: Interior Solution

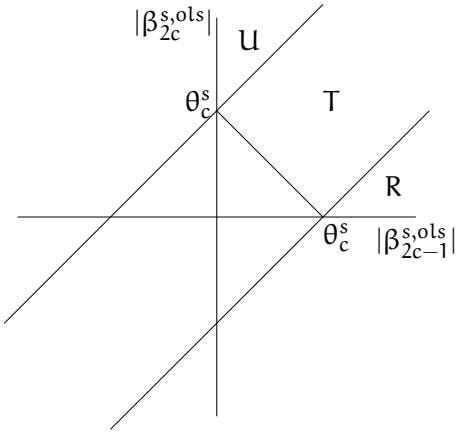


Figure 3: Solutions relative to θ_c^s

A tangency solution will only exist if the locus of tangencies intersects $\Delta(\theta_c^s)$, and expression (17) tells us that this set of tangencies always corresponds to a line with slope 1 passing through

$\left(|\beta_{2c-1}|, |\beta_{2c}|\right)$. This tells us that that whenever $(\beta_{2c-1}, \beta_{2c})$ lie in the region of figure 3 that is marked T, southeast of the line through $(|\beta_{2c-1}|, |\beta_{2c}|) = (0, \theta_c^s)$ with slope equal to one:

$$z_{2c} = \theta_c^s + z_{2c-1} \tag{19}$$

northwest of the line with unit slope that passes through $(|\beta_{2c-1}^{s,ols}|, |\beta_{2c}^{s,ols}|) = (\theta_c^s, 0)$:

$$z_{2c} = -\theta_c^s + z_{2c-1} \tag{20}$$

and northeast of the boundary $\theta_c^s < |\beta_{2c-1}^{s,ols}| + |\beta_{2c}^{s,ols}|$:

$$z_{2c} = \theta_c^s - z_{2c-1} \tag{21}$$

we will have a tangency solution. We can express the conditions, that establish whether the least squares estimates are in region T more compactly as:

$$\left| \, |\beta_{2c}^{s,ols}| - |\beta_{2c-1}^{s,ols}| \, \right| \leq \theta_c^s \tag{22}$$

So whenever the LASSO constraint is binding and we satisfy condition (22) we will have a tangency solution given by (18).

In contrast, if $|\beta_{2c-1}^{s,ols}|$, and $|\beta_{2c}^{s,ols}|$ lie outside region T in figure 3, and so fail to satisfy condition (22), then Lemma 3 implies we cannot have a tangency solution .

For the remaining solutions it is useful to refer to the following lemma:

Lemma 4: whenever $\theta_c^s > 0$

$$sign\left(Q(\theta_c^s - |\beta_{2c-1}^{s,ols}|, -|\beta_{2c}^{s,ols}|, R_c^*) - Q(-|\beta_{2c-1}^{s,ols}|, \theta_c^s - |\beta_{2c}^{s,ols}|, R_c^*)\right) = sign\left(|\beta_{2c-1}^{s,ols}| - |\beta_{2c}^{s,ols}|\right)$$

Proof: See the appendix[3].

Thus we have:

Corollary A: $(|\beta_{2c-1}^{s,ols}|, |\beta_{2c}^{s,ols}|) \in U$ implies

$$Q(-|\beta_{2c-1}^{s,ols}|, \theta_c^s - |\beta_{2c}^{s,ols}|, R_c^*) > Q(\theta_c^s - |\beta_{2c-1}^{s,ols}|, -|\beta_{2c}^{s,ols}|, R_c^*)$$

[3]Notice that the case in which $\theta_c^s = 0$ is trivial, as the only possible solution is $(\hat{z}_{2c-1}, \hat{z}_{2c}) = (0, 0)$ in which case distinctions among tangencies and various corner solutions are vacuous.

Proof: By the definition of U, $(|\beta^{s,ols}_{2c-1}|, |\beta^{s,ols}_{2c}|) \in U$ implies $|\beta^{s,ols}_{2c}| > |\beta^{s,ols}_{2c-1}| + \theta > |\beta^{s,ols}_{2c-1}|$, and so by Lemma 4 we have $Q(-|\beta^{s,ols}_{2c-1}|, \theta^s_c - |\beta^{s,ols}_{2c}|, R^*_c) > Q(\theta^s_c - |\beta^{s,ols}_{2c-1}|, -|\beta^{s,ols}_{2c}|, R^*_c)$. □

Corollary B: $(|\beta^{s,ols}_{2c-1}|, |\beta^{s,ols}_{2c}|) \in R$ implies

$$Q(\theta^s_c - |\beta^{s,ols}_{2c-1}|, -|\beta^{s,ols}_{2c}|, R^*_c) > Q(-|\beta^{s,ols}_{2c-1}|, \theta^s_c - |\beta^{s,ols}_{2c}|, R^*_c)$$

The proof of Corollary B is completely analogous.

Now let's consider what happens when $\left(|\beta^{s,ols}_{2c-1}|, |\beta^{s,ols}_{2c}|\right)$ lies outside of region T, so that we do not have a tangency solution. If we have a $(\beta^{s,ols}_{2c-1}, \beta^{s,ols}_{2c})$ pair above the line (19), in the region marked U in figure 3, so that:

$$|\beta^{s,ols}_{2c}| - |\beta^{s,ols}_{2c-1}| \geq \theta^s_c \tag{23}$$

then by Lemma 2 we must have a solution in $\Delta(\theta^s_c)$, but we have just established that $\left(|\beta^{s,ols}_{2c-1}|, |\beta^{s,ols}_{2c}|\right) \in T$ is a necessary condition for a first quadrant tangency, so the only remaining alternatives are a solution at the upper corner of the constraint set $(0, \theta^s_c)$ and a solution at the righthand corner, $(\theta^s_c, 0)$. Corollary A to Lemma 4 implies that the upper corner of the constraint set:

$$(\beta^{s,ols}_{2c-1}, \beta^{s,ols}_{2c}) = (0, \theta^s_c) \tag{24}$$

provides a better solution. Likewise, if $(\beta^{s,ols}_{2c-1}, \beta^{s,ols}_{2c})$ lie below (20), in the region of figure 3 marked R, so that:

$$|\beta^{s,ols}_{2c}| - |\beta^{s,ols}_{2c-1}| \geq \theta^s_c \tag{25}$$

then Lemma 4 Corollary B betokens a solution at the right corner:

$$(z_{2c-1}, z_{2c}) = (\theta^s_c, 0) \tag{26}$$

It remains for us to solve for θ^s_c. Given that the constraint is binding, which it will be when $\lambda > 0$, we will have $(z_{2c-1}, z_{2c}) \in \Delta(\theta^s_c)$, and we can reposit PZ as:

$$PZ': \quad \min_{z_{2c-1},z_{2c}} \quad Q\left(z_{2c-1} - |\beta_{2c-1}^{s,ols}|, z_{2c} - |\beta_{2c}^{s,ols}|, R_c^{s*}\right)$$

$$\text{subject to} \quad z_{2c-1} + z_{2c} = \theta_c^s$$
$$z_{2c-1} \geq 0$$
$$z_{2c} \geq 0$$

Formulating the Lagrangian we have:

$$\min_{z_{2c-1},z_{2c}} L = Q\big(z_{2c-1} - |\beta_{2c-1}^{s,ols}|, z_{2c} - |\beta_{2c}^{s,ols}|, R_c^{s*}\big) + \lambda\left(z_{2c-1} + z_{2c} - \theta_c^s\right) - \mu_{2c-1}z_{2c-1} - \mu_{2c}z_{2c} \quad (27)$$

Now let's consider the possible solutions.

Lemma 5: At an interior solution to (27) with with both $z_{2c-1} > 0$ and $z_{2c} > 0$ we have:

$$\theta_c^s = |\beta_{2c-1}^{s,ols}| + |\beta_{2c}^{s,ols}| - \frac{\lambda}{1 + R_c^{s*}} \quad (28)$$

Proof: See the Appendix.

Substituting from (28) into (22) and rearranging terms we have our conditions for a tangency solution in terms of $|\beta_{2c-1}^{s,ols}|$, $|\beta_{2c}^{s,ols}|$, and λ:

$$\frac{\lambda}{2(1 + R_c^{s*})} \leq \min\left\{|\beta_{2c-1}^{s,ols}|, |\beta_{2c}^{s,ols}|\right\} \quad (29)$$

when (29) is satisfied we can substitute from (28) into (18) to obtain our tangency solution:

$$(z_{2c-1}, z_{2c}) = \left(|\beta_{2c-1}^{s,ols}| - \frac{\lambda}{2(1 + R_c^{s*})}, |\beta_{2c}^{s,ols}| - \frac{\lambda}{2(1 + R_c^{s*})}\right) \quad (30)$$

The lefthand panel of figure 4 depicts the solutions when $R_c^{s*} > 0$, while the right hand panel shows the case of $R_c^{s*} < 0$. In each figure, the region marked T, for "tangency", corresponds to condition (29). Notice that for a given value of λ this area is more extensive when $R_c^{s*} > 0$, as shown in the left panel, than it is for negatively correlated pairs of regressors, as depicted in the righthand panel.

Now suppose we have a corner solution with $z_{2c-1} > 0$ but $z_{2c} = 0$.

Lemma 6: At a corner solution to (27) with with $z_{2c-1} > 0$ but $z_{2c} = 0$ we have:

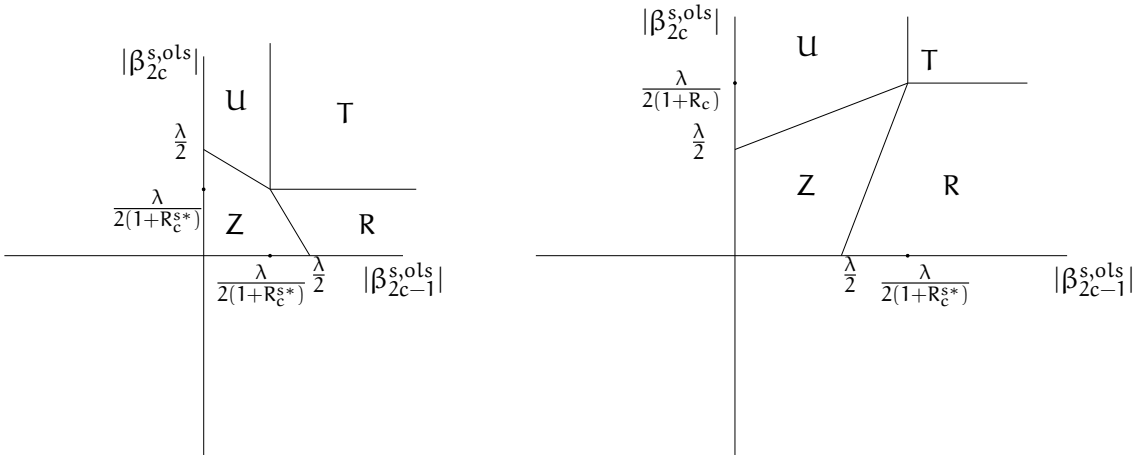


Figure 4: Left: Solutions with $R_c^{s*} > 0$, Right: Solutions with $R_c^{s*} < 0$

$$\theta_c^s = |\beta_{2c-1}^{s,ols}| + R_c^{s*}|\beta_{2c}^{s,ols}| - \frac{\lambda}{2} \tag{31}$$

Proof: See the Appendix.

Of course, this only works provided $\theta_c^s \geq 0$, that is, if:

$$\frac{\lambda}{2} \leq |\beta_{2c-1}^{s,ols}| + R_c^{s*}|\beta_{2c}^{s,ols}| \tag{32}$$

Substituting θ_c^s from (31) into $\frac{\partial L}{\partial z_{2c}} \geq 0$ we have:

$$\begin{aligned}
\frac{\partial L}{\partial z_{2c}} &= 2R_c^{s*}(z_{2c} - |\beta_{2c-1}^{s,ols}|) + 2(0 - |\beta_{2c}^{s,ols}|) + \lambda \\
&= 2R_c^{s*}(|\beta_{2c-1}^{s,ols}| + R_c^{s*}|\beta_{2c}^{s,ols}| - \frac{\lambda}{2} - |\beta_{2c-1}^{s,ols}|) + 2(0 - |\beta_{2c}^{s,ols}|) + \lambda \\
&= -2|\beta_{2c}^{s,ols}| + 2R_c^{s*2}|\beta_{2c}^{s,ols}| + \lambda(1 - R_c^{s*}) \geq 0
\end{aligned}$$

that is, we need:

$$\frac{\lambda}{2} \geq (1 + R_c^*)|\beta_{2c}^{s,ols}| \tag{33}$$

Combining conditions (32) and (33), we have:

$$(1 + R_c^*)|\beta_{2c}^{s,ols}| \leq \frac{\lambda}{2} \leq |\beta_{2c-1}^{s,ols}| + R_c^{s*}|\beta_{2c}^{s,ols}|$$

The set of $(|\beta_{2c-1}^{s,ols}|, |\beta_{2c}^{s,ols}|)$ pairs satisfying this condition corresponds to the region labeled R in figure 4. This region is larger when $R_c^{s*} < 0$, as shown in the right hand panel, than it is when

the regressors are positively correlated–the bicoordinate descent LASSO update is more likely to eliminate one of the coefficients at the update step when the correlation between the regressors is negative.

Substituting from (31) into (26) we have:

$$(z_{2c-1}, z_{2c}) = \left(|\beta_{2c}^{s,ols}| + R_c^{s*} |\beta_{2c-1}^{s,ols}| - \frac{\lambda}{2}, 0 \right) \tag{34}$$

Likewise, we have a solution at the top corner, with:

$$(z_{2c-1}, z_{2c}) = \left(0, |\beta_{2c}^{s,ols}| + R_c^{s*} |\beta_{2c-1}^{s,ols}| - \frac{\lambda}{2} \right) \tag{35}$$

provided:

$$(1 + R_c^*)|\beta_{2c-1}^{s,ols}| \leq \frac{\lambda}{2} \leq |\beta_{2c}^{s,ols}| + R_c^{s*} |\beta_{2c-1}^{s,ols}|$$

Notice that when $R_c^{s*} < 0$ a wider range of parameter estimates results in one parameter, as in regions R and U, or both coefficients, corresponding to region Z, being updated to zero, see the righthand panel of figure 4, than in the case shown in the left panel, corresponding to $R_c^{s*} > 0$. In either case, with $R_c^{s*} \neq 0$ at each pass through the data the bicoordinate descent algorithm allocates slack across the variables more efficiently than does unicoordinate descent, while in the "knife's edge" situation of $R_c^{s*} = 0$ unicoordinate and bicoordinate descent update identically conditional on the remaining parameter estimates.

2.4 Comparison with Unicoordinate Descent

To illustrate the advantages bicoordinate descent affords, consider a typical update step using each of unicoordinate descent, the strategy used by Tibshirani (1996), and Friedman, Hastie and Tibshirani (2010a), and bicoordinate descent. While each algorithm will take it's own pathway to a global solution to the LASSO problem, let's consider a hypothetical update step for coefficient pair c: $\{\beta_{2c-1}, \beta_{2c}\}$ holding constant $\{\beta_j\}_{j \notin \{2c-1, 2c\}}$. Let the LASSO constraint correspond to[4]:

$$\sum_{j=1}^{k} |\beta_j| \leq t$$

[4]Both algorithms parameterize the constraint by setting the first order conditions equal to the Lagrange multiplier λ, but there is an isomorphism between λ and t. Here we make the constraint explicit to clarify the advantages of bicoordinate descent.

where k is the number of potential explanators in our model. Now define:

$$\nu_{ic}^{0} = y_{ic} - \sum_{j \notin \{2c-1, 2c\}} \beta_j x_{ij}$$

and let:

$$\theta^{0} = t - \sum_{j \notin \{2c-1, 2c\}} |\beta_j|$$

while $\{\beta_{2c-1}^{0}, \beta_{2c}^{0}\}$ denote our starting values.

A unicoordinate algorithm will update as follows:

firstly choose $\beta_{2c-1}^{unicoord}$ to solve:

$$min_{\beta_{2c-1}} \sum_{i=1}^{N} \left(\nu_{ic}^{0} - \beta_{2c-1} x_{2c-1,i} - \beta_{2c}^{0} x_{2c,i}\right)^2 \text{ subject to: } |\beta_{2c-1}| \leq \theta^{0} - |\beta_{2c}^{0}|$$

next, update the second coefficient $\beta_{2c}^{unicoord}$ as the solution to:

$$P1_{2c} : min_{\beta_{2c}} \sum_{i=1}^{N} \left(\nu_{ic}^{0} - \beta_{2c-1}^{unicoord} x_{2c-1,i} - \beta_{2c} x_{2c,i}\right)^2$$

subject to:

$$|\beta_{2c}| \leq \theta^{0} - |\beta_{2c-1}^{unicoord}| \tag{36}$$

The resulting sum of squares is:

$$RSS^{0}(\beta_{2c-1}^{unicoord}, \beta_{2c}^{unicoord}) = \sum_{i=1}^{N} \left(\nu_{ic}^{0} - \beta_{2c-1}^{unicoord} x_{2c-1,i} - \beta_{2c}^{unicoord} x_{2c,i}\right)^2$$

While any solution to $P1_{2c}$ must satisfy the constraint (36). Substituting $\beta_{2c}^{unicoord}$ for β_{2c} we are left with:

$$|\beta_{2c-1}^{unicoord}| + |\beta_{2c}^{unicoord}| \leq \theta^{0} \tag{37}$$

In contrast, our bicoordinate descent algorithm will simultaneously update the coefficients to $(\beta_{2c-1}^{bicoord}, \beta_{2c}^{bicoord})$ that solve:

$$P_2^{e} : min_{(\beta_{2c-1}, \beta_{2c})} \sum_{i=1}^{N} \left(\nu_{ic}^{0} - \beta_{2c-1} x_{2c-1,i} - \beta_{2c} x_{2c,i}\right)^2 \tag{38}$$

subject to:

$$|\beta_{2c-1}| + |\beta_{2c}| \leq \theta^0 \tag{39}$$

resulting in a residual sum of squares of:

$$RSS^0(\beta_{2c-1}^{bicoord}, \beta_{2c}^{bicoord}) = \sum_{i=1}^{N} \left(\nu_{ic}^0 - \beta_{2c-1}^{bicoord} x_{2c-1,i} - \beta_{2c}^{bicoord} x_{2c,i}\right)^2$$

In particular, any other coefficient pair $(\beta'_{2c-1}, \beta'_{2c})$ that satisfy condition (39) must give rise to at least as high a sum of squares as the solution to P_2^e, $(\beta_{2c-1}^{bicoord}, \beta_{2c}^{bicoord})$:

$$RSS^0(\beta_{2c-1}^{bicoord}, \beta_{2c}^{bicoord}) \leq RSS^0(\beta'_{2c-1}, \beta'_{2c})$$

But from inequality (37) we know that $(\beta_{2c-1}^{unicoord}, \beta_{2c}^{unicoord})$ satisfy (39), and hence:

$$RSS^0(\beta_{2c-1}^{bicoord}, \beta_{2c}^{bicoord}) \leq RSS^0(\beta_{2c-1}^{unicoord}, \beta_{2c}^{unicoord})$$

For equal starting values, the sum of squares achieved by bicoordinate descent weakly dominates the unicoordinate solution. This is the payoff for choosing the pairings for the explanators, and for the trivial extra calculation involved in computing the bicoordinate updates. In Section 4, we show that in practice bicoordinate descent can substantially reduce the number of updates required for each coefficient pair.

3 Computational Mechanics

The payoff to our algorithm is the speed with which it computes the LASSO estimates. While bicoordinate descent provides savings in the number of passes to be taken through the data, we need also to be abstemious in the computations required at each iteration. We highlight several areas in which we have enhanced the efficiency of the algorithm.

3.1 Warm Starts

Firstly, the `glmnet` algorithm used by (Friedman, Hastie and Tibshirani, 2010b) takes advantage of "warm starts." Their algorithm begins by identifying the smallest value for λ that will still set all of the coefficients equal to zero. Their algorithm descends from this value of λ in a sequence of steps, each of which takes its predecessor as the source of a starting value.

We emulate their approach. Let $r_{y,j}$ be given by:

$$r_{y,j} = \sum_{i=1}^{n} y_i x_{j,i}$$

Now define:

$$\lambda^{max} \equiv 2max\{r_{y,j}\}_{j=1}^{k}$$

Next we choose a multiple ϵ of λ_{max} to define the smallest λ value we will consider, $\lambda_{min} = \epsilon\lambda_{max}$. Next we choose a number of "cross pieces", M, for the trellis. Finally, we construct a "shrinkage step":

$$\sigma = \frac{(1-\epsilon)\lambda_{max}}{M}$$

such that $\lambda_{min} = \lambda_{max} - M\sigma$. At each iteration we shrink λ from it's previous value: $\lambda_m = \lambda_{m-1} - \sigma$. We then start our calculations with lagged values for $\vec{\beta}^{lasso}$ of $\vec{\beta}_{0,lasso} = \vec{\beta}_{-1,lasso} = \vec{0}$. Our starting value for round $m \in \{1, ..., M\}$ of our descent to the next cross piece of the trellis is:

$$\vec{\beta}_m^{m,start} = 2\vec{\beta}^{m-1,lasso} - \sigma\vec{\beta}^{m-2,lasso}$$

At each iteration we then update the first and second lags of $\vec{\beta}$. We find that these interpolated "warm starts" provide more advantageous initial values than do the unalloyed elements of $\vec{\beta}^{m-1,lasso}$.

3.2 Sufficient Statistics

Our algorithm calls for us to calculate $(\beta_{2c-1}^{s,ols}, \beta_{2c}^{s,ols})$ at each iteration step. While these calculations depend on the *status quo* values for the coefficients, they also rely on various cross products from the data. We eschew recalculation of the latter.

Let $r_{j,j'}$ be defined analogously with $r_{y,j}$:

$$r_{j,j'} = \sum_{i=1}^{n} x_{j,i} x_{j',i}$$

Notice that in this notation $R_c \equiv r_{2c-1,2c}$.

To be comprehensive, let's suppose there are $k = 2k^* + 1$ explanators. The case of an even number is yet easier. Now formulate the $k \times (k+1)$ matrix S, which we'll use to keep track of the moments in the data. For $c \leq k^*$, we'll denote row $2c-1$ of S, as $\vec{s}'_{2c-1}$. It's elements are:

$$S_{2c-1,j} = \begin{cases} \frac{-r_{j,2c-1} + r_{2c,2c-1} r_{j,2c}}{1 - r_{c}^{2c,2c-1}} & j \notin \{2c-1, 2c, k+1\} \\ 0 & j \in \{2c-1, 2c\} \\ \frac{r_{y,2c-1} - r_{2c,2c-1} r_{y,2c}}{1 - r_{2c,2c-1}^2} & j = k+1 \end{cases} \tag{40}$$

Likewise, the elements of row $2c$ of S, $\vec{s}'_{2c}$, are:

$$S_{2c,j} = \begin{cases} \frac{-r_{j,2c} + r_{2c,2c-1} r_{j,2c-1}}{1 - r_{2c,2c-1}^2} & j \notin \{2c-1, 2c, k+1\} \\ 0 & j \in \{2c-1, 2c\} \\ \frac{r_{y,2c} - r_{2c,2c-1} r_{y,2c-1}}{1 - r_{2c,2c-1}^2} & j = k+1 \end{cases} \tag{41}$$

the k^{th} and final row of S, $\vec{s}'_{k}$, is:

$$S_{k,j} = \begin{cases} -r_{j,k} & 1 \leq j < k \\ 0 & j = k \\ r_{y,k} & j = k+1 \end{cases} \tag{42}$$

Starting from the initial $(k+1) \times 1$ vector $\vec{\alpha}^{s,c,ols}$, where:

$$\alpha_j^{s,c,ols} = \begin{cases} \beta_j^{s,ols} & j \leq 2c-2 \\ \beta_j^{s-1,ols} & 2c+1 \leq j \leq k \\ 1 & j = k+1 \end{cases} \tag{43}$$

while:

$$\alpha_j^{s,c,ols} = \begin{cases} \beta_j^{s,ols} & j < k \\ \beta_k^{s-1,ols} & j = k \\ 1 & j = k+1 \end{cases} \tag{44}$$

we update $(\beta_{2c-1}^{s,ols}, \beta_{2c}^{s,ols})$:

$$\beta_{2c-1}^{s,ols} = \vec{s}'_{2c-1} \vec{\alpha}^{s,c,ols} \text{ and } \beta_{2c}^{s,ols} = \vec{s}'_{2c} \vec{\alpha}^{s,c,ols} \tag{45}$$

While:

$$\beta_k^{s,ols} = \vec{s}'_k \vec{\alpha}^{s,k,ols} \tag{46}$$

We note that this algorithm yields the same results as reiterated solution of P_2^c, a claim we formalize as:

Lemma 7: Given $\left\{\beta_j^{s,ols}\right\}_{j\leq 2c-2}, \left\{\beta_j^{s-1,ols}\right\}_{2c-1<j\leq k}\Big\}$, and $\left\{y_i, \{x_{ij}\}_{j=1}^k\right\}_{i=1}^n$, the left hand side values of (45) and (46) correspond to solutions for $P2_c^s$ and $P3_c^s$, respectively.

Proof: See the appendix.

Notice that as we move toward a solution the $\vec{\alpha}^{s,c,ols}$ change, but S remains the same. For large values of n this can represent a substantial computational saving. Hastie[5] describes using a similar procedure, which he calls "covariance updating." Indeed, we suspect that, adjusting for differences in notation, the row of S that deals with the odd singleton variable in our framework coincides exactly with the algorithmic artfulness described by Hastie.

3.3 Managing the Active Set

Another important source of computational speed is the management of the "active set" used in the estimation. The idea is to restrict our attention to only variables that have a chance of surviving the LASSO process, and for this we have a straightforward screening procedure.

Firstly we identify the explanator x_{max} for which $r_{max,y} \geq r_{j,y} \forall j$, this is our starting value λ_{max} for λ. Our "active set" of variables consists solely of x_{max}. At $\lambda = r_{max,y}$ the LASSO with x_{max} as our sole potential explanator will produce a coefficient of zero, just barely censoring x_{max}. We then reduce λ by successive increments.

Now suppose that corresponding to the current value of λ we have an active set A_λ of variables $\{x_k\}_{k\in A_\lambda}$, we have estimated the LASSO coefficients corresponding to λ, and we are about to move on to the next lower value, $\lambda' = \lambda - \sigma$, in our sequence. Before we move on, for each variable x_j that is excluded from the active set we calculate:

$$b_j^{\lambda^{shadow}} = r_{y,j} - \sum_{k\in A_\lambda} r_{j,k}\hat{\beta}_k^{LASSO}(\lambda)$$

Notice that $b_j^{\lambda^{shadow}}$ is equivalent to the j^{th} element of $X'\vec{e}^{\lambda}$ where $\vec{e}^{\lambda}$ is the vector of residual values associated with λ.

Next we check whether there are any variables in the complement of A_λ for which:

$$b_j^{\lambda^{shadow}} > \lambda \tag{47}$$

If there are any such variables, we add them to A_λ, and then repeat the calculations for the λ iteration. If there are no such variables, we then check whether for any variables outside of A_λ:

[5] See `http://web.stanford.edu/~hastie/TALKS/glmnet.pdf`

$$b_j^{\lambda^{shadow}} > \lambda' \tag{48}$$

We then generate the active list for λ', $A_{\lambda'}$, as the union of A_λ and the set of new variables that satisfy condition (48).

Next, we add any variable x_{j^*} for which $b_j^{\lambda^{shadow}} > \lambda'$ to the active list $A_{\lambda'}$ corresponding to λ'.

These measures reduce the number of actual calculations needed to update the coefficients at each iteration. In "wide" datasets with large numbers of explanatory variables this can represent a substantial reduction in the computational burden. This approach to managing the active set corresponds with the "strong rule" analyzed by Tibshirani et al. (2012).

3.4 Bounding the Correlations

Even the calculations required to evaluate (48) can become burdensome as the set of potential explanators expands. We can delay, and in some cases entirely avoid, calculating some of the correlations amongst the dormant variables by recycling the correlations we computed in the process of establishing λ_{max}, which required us to calibrate the crossproducts of each column of X with $\vec{y}$. Once we have the current round of coefficient estimates in hand, it is likewise straightforward to calculate the crossproduct of the error vector $\vec{e}^{\lambda}$ with $\vec{y}$. As it happens, these quantities convey some information about the magnitudes of the correlations between $\vec{e}^{\lambda}$ the columns of X, which is to say, about the $b_j^{\lambda^{shadow}}$.

Consider the following matrix of crossproducts involving $\vec{y}$, $\vec{e}^{\lambda}$ and $\vec{x}$, and arbitrary column of X:

$$M = \begin{pmatrix} s_y^2 & s_y s_e r_{ey} & s_y r_{xy} \\ s_y s_e r_{ey} & s_e^2 & s_e r_{ex} \\ s_y r_{xy} & s_e r_{ex} & 1 \end{pmatrix} \tag{49}$$

We know that M is positive definite; all of our variables are centered, and the elements of X have been normalized, so M is in fact a sample covariance matrix.

In particular, M will be positive definite if and only if the following matrix is also positive definite[6]:

[6] Just pre and post multiply M by a diagonal matrix that contains $\frac{1}{s_y}$ as the first diagonal entry, $\frac{1}{s_e}$ as the second, and one as the third element of the diagonal.

$$C = \begin{pmatrix} 1 & r_{ey} & r_{xy} \\ r_{ey} & 1 & r_{ex} \\ r_{xy} & r_{ex} & 1 \end{pmatrix} \tag{50}$$

In particular, from positive definiteness of C we know that $r_{ex}^2 < 1$, while $\det(C) > 0$, that is:

$$-r_{ex}^2 + 2r_{ey}r_{ex}r_{xy} + 1 - r_{ey}^2 - r_{xy}^2 > 0 \tag{51}$$

Notice that $r_{ex} = 0$ satisfies (51) only if:

$$r_{ey}^2 + r_{xy}^2 < 1 \tag{52}$$

If the correlations of $\vec{e}^{\lambda}$ and $\vec{x}$ with $\vec{y}$ violate condition (52), then their correlation with each other cannot equal zero.

The roots of:

$$\theta^2 - (2r_{xy}r_{ey})\theta - (1 - r_{xy}^2 - r_{ey}^2) = 0 \tag{53}$$

are:

$$(\theta_1, \theta_2) = (r_{ey}r_{xy} - \sqrt{1 - r_{ey}^2}\sqrt{1 - r_{xy}^2}, r_{ey}r_{xy} + \sqrt{1 - r_{ey}^2}\sqrt{1 - r_{xy}^2}) \tag{54}$$

Notice that these roots straddle the origin when we satisfy inequality 52, whereas the are both of the same sign when the inequality is reversed.

In any case, positive definiteness of C tells us that r_{ex} must take on a value between $\min\{\theta_1, \theta_2\}$ and $\max\{\theta_1, \theta_2\}$. Moreover, the roots to (53) will always lie on $[-1, 1]$. In fact, when $r_{ey} = r_{xy}$ the roots are $\{-1, 1\}$, while otherwise our roots, and r_{ex} are guaranteed to lie on the interior of $(-1, 1)$.

So, in particular, for the gradient check at step k we compare $s_e r_{ex}$ with λ_{k-1}, adding $\vec{x}$ to the active set if:

$$2s_e|r_{ex}| < \lambda_{k-1} \tag{55}$$

whereas we otherwise omit it from the current round of calculations.

Of course, evaluating (55) entails calculating r_{ex}. But we can obviate the calculation of this correlation when we satisfy:

$$2s_e\left(|r_{ey}r_{xy}| + \sqrt{1-r_{ey}^2}\sqrt{1-r_{xy}^2}\right) < \lambda_{k-1} \tag{56}$$

When condition (56) is met we can exploit the fact that r_{ex} is bounded by the roots, so that:

$$|r_{ex}| < \left(|r_{ey}r_{xy}| + \sqrt{1-r_{ey}^2}\sqrt{1-r_{xy}^2}\right)$$

and hence we know r_{ex} will satisfy condition (55) without having to calculate it.

At each iteration of the LASSO our algorithm tallies the vector of residuals $\vec{e}^{\lambda}$, and then calculates a fresh value r_{ey}, but this allows us to eschew the computation of any r_{ex} that satisfies (56).

3.5 Analytical inflection points and the Active Set

All of the preceding computational procedures are easily adapted to cases in which some of the variables are exempted from the LASSO, as might arise when one knows that a certain list of variables from a "reference model" need to be included in the specification. However, when the entire complement of variables are subject to the LASSO, we have one more computational arrow in our quiver–we can solve for the first two inflection points after λ_{max} at very low computation cost, bringing analytical formulas to bear. This enables us to jump quickly through the initial portion of the LASSO trellis, providing another substantial boost to the speed of our algorithm. These first two steps are tantamount to the initial updates used by the LARS algorithm of Efron et al. (2004). The LARS algorithm entails inverting a cascade of increasingly large matrices, but our first two steps involve no matrices larger than 2×2.

3.5.1 The First Jump

On the interval between λ^{max} and the smallest λ value, $\lambda_{sidekick}$, that leaves but one nonzero LASSO coefficient we know that the coefficient for the nonzero LASSO coefficient is a linear function of λ:

$$\beta_{max} = sign(r_{max,y})\left(|r_{max,y}| - \tfrac{\lambda}{2}\right) = a_{max} + c_{max}\lambda$$

where $a_{max} = r_{max,y}$ and $c_{max} = -\frac{1}{2}sign(r_{max,y})$.

Over the same interval, the remaining OLS coefficients, conditional on β_{max}, are themselves linear in β_{max}:

$$\beta_j = r_{j,y} - r_{max,j}\beta_{max}$$

and hence they are also linear in λ:

$$\beta_j = (r_{j,y} - r_{max,j}a_{max}) + r_{max,j}c_{max}\lambda = a_j + c_j\lambda$$

where $a_j = r_{j,y} - r_{max,j}r_{max,y}$ and $c_j = \frac{1}{2}sign(r_{max,y})r_{max,j}$.

Every variable except x_{max} will satisfy the following condition for $\lambda \in (\lambda_{sidekick}, \lambda_{max})$:

$$-\lambda < a_j + c_j\lambda < \lambda$$

Now let:

$$\lambda_j^+ = \frac{a_j}{1 - c_j} \text{ and } \lambda_j^- = \frac{-a_j}{1 + c_j}$$

while:

$$\hat{\lambda}_j = \begin{cases} \lambda_j^+ & \text{if } c_j > 0 \\ \lambda_j^- & \text{if } c_j < 0 \end{cases}$$

and $\lambda_j^* = max\{\hat{\lambda}_j, 0\}$.

It follows that:

$$\lambda_{sidekick} = max_j\{\lambda_j^*\}_{j \neq max}$$

If we let $x_{sidekick}$ denote the variable associated with this maximum value we see that at $\lambda_{sidekick}$ we have:

$$\beta_{max}^{LASSO} = a_{max} + c_{max}\lambda_{sidekick}$$

while all the other beta values are equal to zero. We'll denote the active set of coefficients as $A^{\lambda_{sidekick}}$. After the first jump $A^{\lambda_{sidekick}}$ consists of $\{sidekick, max\}$.

3.5.2 The Second Jump

Now let's consider what happens for $\lambda \in (\lambda_{next}, \lambda_{sidekick})$, where λ_{next} corresponds to the next inflection point after $\lambda_{sidekick}$. Let $(\hat{\alpha}_{max}, \hat{\alpha}_{sidekick}\}$ denote the coefficients from an OLS regression of y on x_{max} and $x_{sidekick}$. Along this interval we will have an interior solution for the LASSO coefficients corresponding to x_{max} and $x_{sidekick}$, which will thus be linear functions of λ:

$$\alpha_{max}^{LASSO} = \hat{\alpha}_{max} - \frac{sign(\hat{\alpha}_{max})}{2(1 + r_{max,sidekick})}\lambda \text{ and } \alpha_{sidekick}^{LASSO} = \hat{\alpha}_{sidekick} - \frac{sign(\hat{\alpha}_{sidekick})}{2(1 + r_{max,sidekick})}\lambda$$

while the conditional least squares estimator for each of the remaining coefficients is linear in α_{max}^{LASSO} and $\alpha_{sidekick}^{LASSO}$:

$$\hat{\beta}_j = r_{yj} - \alpha_{max}^{LASSO} r_{j,max} - \alpha_{sidekick}^{LASSO} r_{j,sidekick}$$

Substituting from our expressions for the two active coefficients this becomes:

$$\hat{\beta}_j = \tilde{a}_j + \tilde{c}_j\lambda$$

where:

$$\tilde{a}_j = r_{yj} - \hat{\alpha}_{max} r_{j,max} - \hat{\alpha}_{sidekick} r_{j,sidekick}$$

and:

$$\tilde{c}_j = \frac{sign(\hat{\alpha}_{max}) r_{j,max} + sign(\hat{\alpha}_{sidekick}) r_{j,sidekick}}{2(1 + r_{max,sidekick})}$$

We now proceed in parallel with the first update, every variable except x_{max} and $x_{sidekick}$ will satisfy the following condition for $\lambda \in (\lambda_{next}, \lambda_{sidekick})$:

$$-\lambda < \tilde{a}_j + \tilde{c}_j\lambda < \lambda$$

Now let:

$$\tilde{\lambda}_j^+ = \frac{\tilde{a}_j}{1 - \tilde{c}_j} \text{ and } \tilde{\lambda}_j^- = \frac{-\tilde{a}_j}{1 + \tilde{c}_j}$$

$$\bar{\lambda}_j = \begin{cases} \tilde{\lambda}_j^+ & \text{if } c_j > 0 \\ \tilde{\lambda}_j^- & \text{if } c_j < 0 \end{cases}$$

and $\tilde{\lambda}_j^* = max\{\bar{\lambda}_j, 0\}$.

It follows that:

$$\lambda_{next} = max_j\{\tilde{\lambda}_j^*\}_{j \notin \{max,sidekick\}}$$

If we let x_{next} denote the variable associated with this maximum value we see that at λ_{next} we have:

$$\begin{aligned} \alpha_{max}^{LASSO} &= \hat{\alpha}_{max} - sign(\hat{\alpha}_{max})\frac{\lambda_{next}}{2(1+r_{max,sidekick})} \\ \alpha_{sidekick}^{LASSO} &= \hat{\alpha}_{sidekick} - sign(\hat{\alpha}_{sidekick})\frac{\lambda_{next}}{2(1+r_{max,sidekick})} \end{aligned}$$

while all the other beta values are equal to zero[7]. Notice that $A^{\lambda_{next}} = \{next, sidekick, max\}$.

4 Comparative Timing

Our algorithm is still at the developmental stage, and in the hands of professional programmers we do not doubt that our procedure will execute even more rapidly than it does. However, we do want to provide the reader with an idea of the efficacy of our code so we here present some benchmarks relative to the unicoordinate descent `glmnet` algorithm, using a variety of datasets that vary in size, and in the severity of the collinearity observed among their component variables.

	Data Passes	Min.	First Quintile	Mean	Median	Third Quintile	Max.
DIABETES	Diabetes data from Efron et al. (2004)						
Bicoord.	215	412.044	427.4355	461.4442	452.783	478.276	1062.181
`glmnet`	1164	1839.346	1883.3175	2040.9046	1921.353	1975.819	7571.260
RED	Wine quality data (red varietals) from Cortez et al. (2009)						
Bicoord.	121	15.82811	17.01458	18.62527	17.43571	18.32009	45.96997
`glmnet`	342	60.78047	62.07158	65.92969	62.99484	66.20146	125.62298
SOIL	Soil Quality data from Bondell and Smith (2008)						
Bicoord.	245	43.91919	45.29539	46.01222	45.85828	46.47594	49.01433
`glmnet`	647	138.48964	143.93945	146.69634	144.66505	145.93290	173.56228
WHITE	Wine quality data (white varietals) from Cortez et al. (2009)						
Bicoord.	253	51.66146	52.67688	53.69331	53.02394	53.33742	82.76253
`glmnet`	520	100.26352	102.47824	104.02415	102.89359	103.92719	131.41892

Table 1: **Speed Comparisons between Bicoordinate Decent and glmnet**: Results are from 100 trials. Units are in milliseconds.

[7]In the very unlikely event that $\lambda_{next} < \lambda_{drop} = 2sign(\hat{\alpha}_{max})(1+r_{max,sidekick})\hat{\alpha}_{max}$ we instead stop at λ_{drop}, at which point the "max" variable goes dormant, and we repeat the second jump using sidekick in place of max, and λ_{drop} instead of $\lambda_{sidekick}$.

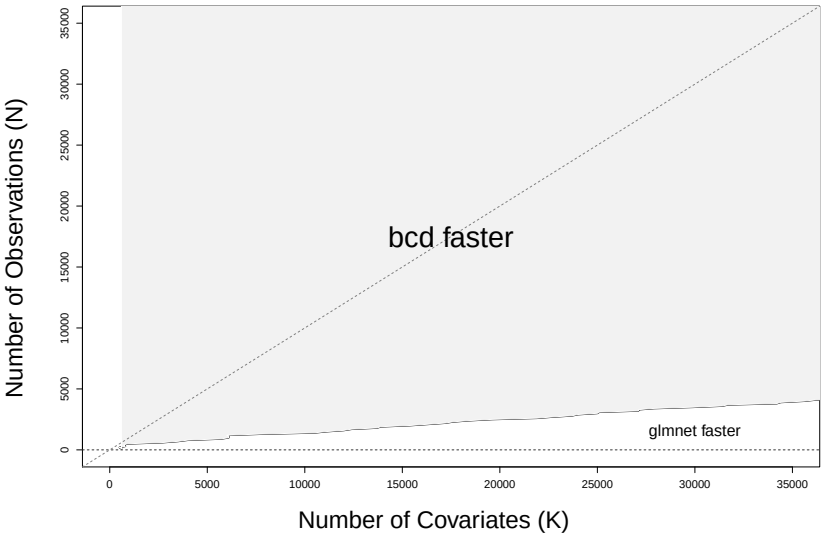


Figure 5: **Sample Characteristics and Relative Convergence Speed**: Simulation is based on a set of randomly generated datasets with varying number of observations (N) and covariates (K). The shaded region in gray color corresponds to the size of input matrix (N $\times$ K) where `bcd` is faster than `glmnet`.

Results of several time trials appear in Table 1. We used the `microbenchmark` package in R for the speed tests, with 100 trials. The first column counts complete "passes" through the data, with each pass corresponding to a full set of parameter updates. Notice that this count is not an accounting artifact of bicoordinate descent updating parameters two at a time. If we have twenty parameters in our model, one data pass by bicoordinate descent consists of updating each of the ten pairs of parameters, whereas one data pass for unicoordinate descent involves twenty single parameter updates, either way twenty parameters are updated, and either way we count but a single pass through the data. Managing the active set also leaves our accounting for iterations unaffected, if we have twenty parameters with six active and fourteen dormant, then one round of updates to the six active parameters counts as a full "data pass."

We observe a dramatic reduction in data passes moving from `glmnet` to bicoordinate descent, and a comparable reduction in the time required to conduct the calculations, with the bicoordinate algorithm working between two and three and a half times as fast. We note that this speed advantage comes despite the extra "overhead" costs of bicoordinate descent, which recalibrates the parameter matches every time a new parameter enters the active set.

In general, for a given number of explanators there is a critical sample size above which bicoordinate descent achieves results faster than `glmnet`.

5 Extensions

In this section we present some results for the probit model, while we defer analytical results to Appendix B. We note that while the probit model is of considerable interest in its own right as an alternative to the logit, probits are also widely used in dealing with the censoring and sample selection issues that can arise in a regression context.

5.1 Extension to the Probit

It is straightforward to extend the bicoordinate descent algorithm to encompass the probit model (Bliss, 1934), the nonlinear nature of that framework notwithstanding. To do so we exploit the EM algorithm of Dempster, Laird and Rubin (1977), which allows us to convert a curvilinear maximum likelihood problem into an syncopated sequence of "e-steps" and regression like "m-steps" each of which lends itself to an unvarnished application of our LASSO algorithm. The applicability of the EM algorithm to probit models was noticed by McCulloch (1999), who observes that the "working probits" method of Finney (1952), which applies an iterative weighted least squares algorithm, converges more rapidly. However, for our purposes the straightforward interface between the m-step and bicoordinate descent renders the EM process a natural artifice. We are not the first to advocate the use of an EM algorithm to sparsify a probit model. Figueredo (2003) shows how to apply the EM algorithm as part of a Maximum A Posteriori estimation of a Bayesian probit model with Laplacian priors that closely resembles the LASSO. We note that `glmnet` contains a very efficient logit feature, but does not include a probit module. Extension of bicoordinate descent to a logit setting is also possible, but it entails the use of a weighted least squares algorithm, and a consequent increment to the complexity of the algorithm described in the next section.

5.2 The Model

We now seek to solve:

$$\max_{\beta} \sum_{i=1}^{n} \ln\Phi\left((2\delta_i - 1)\vec{x}_i\vec{\beta}\right) \text{ subject to } \sum_{j=1}^{k} |\beta_j| \leq T \tag{57}$$

where $\delta_i \in \{0, 1\}$ is the dichotomous dependent variable for the probit.

We'll do this using the EM algorithm of Dempster, Laird and Rubin (1977). First, let's consider the E-step.

5.2.1 The E-step

First of all, let's define $\bar{p} = \frac{1}{n}\sum_{i=1}^{n}\delta_i$ as the sample mean of δ. We can start out estimating the latent dependent variable, z_i^*, as:

$$z_i^0 = \begin{cases} \Phi^{-1}(\bar{p}) & \text{if } \delta_i = 1 \\ \Phi^{-1}(1-\bar{p}) & \text{if } \delta_i = 0 \end{cases}$$

More generally, if we inherit a set of estimates $\{z_i^{s-1}\}$ from the preceding iteration, $s-1$, of the algorithm, along with estimates $\{\beta_j^{s-1,\text{LASSO}}\}_{j=1}^{k}$, for $\{\beta_j\}_{j=1}^{k}$, we can start from:

$$z_i = \vec{x}_i'\vec{\beta} - \epsilon_i$$

where $\epsilon_i \sim N(0,1)$. This framework tells us the probability $\delta_i = 1$ is equal to the probability that $\epsilon \leq \vec{x}_i'\vec{\beta}$, which is to say $\Phi(\vec{x}_i'\vec{\beta})$. If we take expectations we have:

$$E\{z_i|\delta_i\} = \vec{x}_i'\vec{\beta} - E\{\epsilon_i|\delta_i, \vec{x}_i'\vec{\beta}\} = \vec{x}_i'\vec{\beta} + (2\delta_i - 1)\frac{\phi(\vec{x}_i'\vec{\beta})}{\Phi\big((2\delta_i-1)\vec{x}_i'\vec{\beta}\big)} \tag{58}$$

To align with our LASSO framework we want to set the mean of our latent dependent variable z equal to zero at each iteration. This means that for a given $\{z_i^{s-1}\}_{i=1}^{n}$ we calculate the mean $\bar{z}^a$. We then generate an estimate for $\{z_i^s\}_{i=1}^{n}$:

$$\hat{z}_i(\bar{z}^a) = \bar{z}^a + \vec{x}_i'\vec{\beta}^{s,\text{LASSO}} + (2\delta_i - 1)\frac{\phi(\bar{z}^a + \vec{x}_i'\vec{\beta})}{\Phi\big((2\delta_i-1)[\bar{z}^a + \vec{x}_i'\vec{\beta}]\big)} \tag{59}$$

We now calculate the mean of the resulting values for $\hat{z}_i$, call this $\bar{z}^b$. We repeat the steps with $\bar{z}^b$ in place of $\bar{z}^a$ until the resulting sequence of means converges to $\bar{z}^s$. We then subtract this mean from the expression in (59) to obtain the latent dependent variable for the M-step $z_i^s = \hat{z}_i(\bar{z}^s) - \bar{z}^s$.

5.2.2 The M-step

We now apply the next stage of the bicoordinate descent algorithm using z_i^s as our dependent variable in place of y_i.

To speed the process we evaluate the inverse Mills ratios $(2\delta_i - 1)\frac{\phi(\bar{z}^a + \vec{x}_i'\vec{\beta})}{\Phi\big((2\delta_i-1)[\bar{z}^a + \vec{x}_i'\vec{\beta}]\big)}$ using lookup tables, and otherwise applying the computational expedients we employ with the regular bicoordinate descent algorithm[8]. One artifice from our least squares toolkit that does not translate

[8]We have a version of the LASSOed probit software adapted to encompass settings in which only some of the variables are to be LASSOed.

to the LASSOed probit is the application of two LARS like steps at the start of the algorithm.

6 Discussion

Given the advantages offered by exploitation of the correlations among the explanators, why should one stop at bicoordinate descent? Why not coordinate across even more variables at each step? Indeed, when the design matrix is of full rank the standard formula for calculating regression coefficients converges in but a single step. However, with a large number of explanators the constraint set of the LASSO becomes a high dimensional polytope with myriad corners, edges, and faces to check for possible solutions. Also, of course, the matrix inversion problem can be computationally intense when the design matrix is of full rank but large, while it becomes impossible when the matrix is nonsingular, as it is guaranteed to be for a sufficiently large number of explanators.

The huge appeal of one at a time coordinate wise descent is its robustness to the rank of the design matrix. Tibshirani's soft thresholding vastly streamlines the updating process, and it relies on the convenient result that the signs of the LASSO coefficient updates will never be opposite those of the signs of the unconstrained coordinate wise regression update steps.

The analogy to this "no sign reversal" condition in our formulation is that our pairwise LASSO updates are guaranteed to remain in the closure of the same quadrant as the pairwise conditional regression coefficient updates. The cost of moving to bicoordinate descent is that it will only work for pairs of explanatory variables that are not perfectly correlated. But this is a scant price to pay, as the analyst has a variety of options; our solution is simply to drop one element one each perfectly correlated pair of variables from the specification. Alternatively, one could simply rematch the perfectly correlated pairs with other variables, or one could apply ordinary coordinate wise descent to the offending pairs.

Could this approach be extended to encompass tricoordinate descent? Perhaps, but the very convenient result that the LASSO updates will always be found in the same quadrants as the unconstrained updates does not generalize. In his figure 3a, Tibshirani (1996) p.271 shows that with three variables the LASSO coefficients may constitute interior solutions in a different quadrant than the conditional regression coefficients. An interesting subject for ongoing research is to identify whether there are conditions on the correlations among triples of variables that guarantee that the LASSO updates will be contained in the same octant as the least squares coefficient update steps.

Adapting bicoordinate descent to a weighted least squares setting is a subject of our ongoing research. In a weighted least squares setting the LASSOed coefficients are no longer guaranteed

to inhabit the same quadrant as the conditional regression coefficients, however, we show that it is still possible to generate closed form solutions for the pairwise LASSO updates. As a practical matter, most of these do remain in the same quadrant as the conditional regression coefficients. The weighted least squares extension permits analysts to apply the bicoordinate descent algorithm to logit models, as well as to other nonlinear models, such as those involving duration data.

7 Conclusion

We develop a bicoordinate descent algorithm for the LASSO. When the explanatory variables of a regression model are correlated our algorithm takes a more efficient path toward the solution, while it entails only a trivial amount of extra calculation as compared with the standard unicoordinate descent approach. We compare the speediness of our algorithm with that of the state of the art `glmnet` algorithm of Friedman, Hastie and Tibshirani (2010a). We also adapt the bicoordinate descent approach to encompass an application of the LASSO to the probit. In addition, we show how to adapt bicoordinate descent for weighted least squares estimators.

References

Bliss, C. 1934. "The Method of Probits." Science 79:38–39.

Bondell, Howard D. and Brian J. Smith. 2008. "Simultaneous Regression Shrinkage, Variable Selection, and Supervised Clustering of Predictors with OSCAR." Biometrics 64:115–23.

Cortez, P., A. Cerdeira, F. Almeida, T. Matos and J. Reis. 2009. "Modeling wine preferences by data mining from physicochemical properties." Decision Support Systems 47:547–553.

Dempster, A.P., N.M. Laird and D.B. Rubin. 1977. "Maximum Likelihood from Incomplete Data *via* the EM Algorithm (With discussion)." Journal of the Royal Statistcal Society, Series B 39:1–38.

Efron, Bradley, Trevor Hastie, Iain Johnstone and Robert Tibshirani. 2004. "Least Angle Regression." The Annals of Statistics 32:407–451.

Figueredo, Mário A. T. 2003. "Adaptive Sparseness for Supervised Learning." IEEE Transactions on Pattern Analysis and Machine Intelligence 25:1150–9.

Finney, D.J. 1952. Probit Analysis. Cambridge: Cambridge University Press.

Friedman, Jerome H., Trevor Hastie and Rob Tibshirani. 2010a. "Regularization Paths for Generalized Linear Models via Coordinate Descent." Journal of Statistical Software 33:1–22.

Friedman, Jerome, Trevor Hastie and Robert Tibshirani. 2010b. "Regularization Paths for Generalized Linear Models via Coordinate Descent." Journal of Statistical Software 33:1–22.
URL: *http://www.jstatsoft.org/v33/i01/*

Fu, Wenjiang. 1998. "Penalized Regressions: The Bridge vs the Lasso." Journal of Computational and Graphical Statistics 7:397–416.

McCulloch, Charles E. 1999. "Generalized Linear Models." Working Paper.

Tibshirani, Robert. 1996. "Regression Shrinkage and Selection via the Lasso." Journal of the Royal Statistical Society, Series B. 58:267–88.

Tibshirani, Robert, Jacob Bien, Jerome Friedman, Trevo Hastie, Noah Simon, Jonathan Taylor and Ryan Tibshirani. 2012. "Strong rules for discarding predicators in lasso-type problems." Journal of the Royal Statistcal Society, Series B 74:245–66.

Appendix A

Proof of Lemma 1

Proof.

$$
\begin{aligned}
& RSS_c\left(\beta_{2c-1},\beta_{2c}|\left\{v_{ic}^s, x_{i,2c-1}, x_{i,2c}\right\}_{i=1}^n\right) \\
= & \sum_{i=1}^n\left(v_{ic}^s-\beta_{2c-1}x_{i,2c-1}-\beta_{2c}x_{i,2c}\right)^2 \\
= & \sum_{i=1}^n\left(v_{ic}^s-\beta_{2c-1}^{s,ols}x_{i,2c-1}-\beta_{2c}^{s,ols}x_{i,2c}-(\beta_{2c-1}-\beta_{2c-1}^{s,ols})x_{i,2c-1}-(\beta_{2c}-\beta_{2c}^{s,ols})x_{i,2c}\right)^2 \\
= & \sum_{i=1}^n\Bigg(\left(v_{ic}^s-\beta_{2c-1}^{s,ols}x_{i,2c-1}-\beta_{2c}^{s,ols}x_{i,2c}\right)^2 \\
& \quad +2\left(v_{ic}^s-\beta_{2c-1}^{s,ols}x_{i,2c-1}-\beta_{2c}^{s,ols}x_{i,2c}\right)\left((\beta_{2c-1}-\beta_{2c-1}^{s,ols})x_{i,2c-1}+(\beta_{2c}-\beta_{2c}^{s,ols})x_{i,2c}\right) \\
& \quad -\left((\beta_{2c-1}-\beta_{2c-1}^{s,ols})x_{i,2c-1}+(\beta_{2c}-\beta_{2c}^{s,ols})x_{i,2c}\right)^2\Bigg) \\
= & \sum_{i=1}^n\left(v_{ic}^s-\beta_{2c-1}^{s,ols}x_{i,2c-1}-\beta_{2c}^{s,ols}x_{i,2c}\right)^2 \\
& \quad +2(\beta_{2c-1}-\beta_{2c-1}^{s,ols})\left(\sum_{i=1}^n\left(v_{ic}^s-\beta_{2c-1}^{s,ols}x_{i,2c-1}-\beta_{2c}^{s,ols}x_{i,2c}\right)x_{i,2c-1}\right) \\
& \quad +2(\beta_{2c}-\beta_{2c}^{s,ols})\left(\sum_{i=1}^n\left(v_{ic}^s-\beta_{2c-1}^{s,ols}x_{i,2c-1}-\beta_{2c}^{s,ols}x_{i,2c}\right)x_{i,2c}\right) \\
& \quad +\sum_{i=1}^n\left((\beta_{2c-1}-\beta_{2c-1}^{s,ols})x_{i,2c-1}+(\beta_{2c}-\beta_{2c}^{s,ols})x_{i,2c}\right)^2
\end{aligned}
$$

but the least squares estimates are chosen to guarantee that:

$$
\sum_{i=1}^n\left(v_{ic}^s-\beta_{2c-1}^{s,ols}x_{i,2c-1}-\beta_{2c}^{s,ols}x_{i,2c}\right)x_{i,2c-1}=0 \qquad \text{and} \qquad \sum_{i=1}^n\left(v_{ic}^s-\beta_{2c-1}^{s,ols}x_{i,2c-1}-\beta_{2c}^{s,ols}x_{i,2c}\right)x_{i,2c}=0
$$

so our expression simplifies to:

$$
\begin{aligned}
& RSS_c\left(\beta_{2c-1},\beta_{2c}|\left\{v_{ic}^s, x_{i,2c-1}, x_{i,2c}\right\}_{i=1}^n\right) \\
= & \sum_{i=1}^n\left(v_{ic}^s-\beta_{2c-1}^{s,ols}x_{i,2c-1}-\beta_{2c}^{s,ols}x_{i,2c}\right)^2+\sum_{i=1}^n\left((\beta_{2c-1}-\beta_{2c-1}^{s,ols})x_{i,2c-1}+(\beta_{2c}-\beta_{2c}^{s,ols})x_{i,2c}\right)^2 \\
= & \ sse_0^{c,s}+Q(\beta_{2c-1}-\beta_{2c-1}^{s,ols},\beta_{2c}-\beta_{2c}^{s,ols},R_c)
\end{aligned}
$$

□

Proof of Lemma 2

Proof. When the constraint is not binding, the result is trivial and the OLS and LASSO estimates coincide, whereas if $\theta_c^s = 0$ then the result again holds trivially, as the LASSO estimates must both equal zero. Now consider what happens when $0 < \theta_c^s < |\beta_{2c-1}^{s,ols}| + |\beta_{2c}^{s,ols}|$. Any pair $\vec{z}_0 = (z_{2c-1}, z_{2c})$ such that $z_{2c-1} + z_{2c} = \alpha < \theta_c^s$ is dominated by $\vec{z}' = (z_{2c-1} + \frac{\theta_c^s - \alpha}{2}, z_{2c} + \frac{\theta_c^s - \alpha}{2})$:

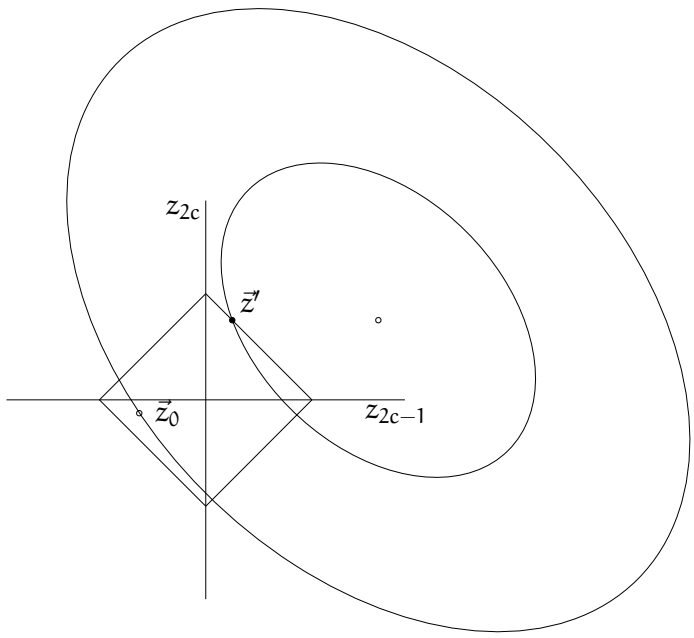


Figure 6: The unit simplex dominates the constraint set.

$$
\begin{aligned}
& Q\big(z_{2c-1} + \tfrac{\theta_c^s - \alpha}{2} - |\beta_{2c-1}^{s,ols}|, z_{2c} + \tfrac{\theta_c^s - \alpha}{2} - |\beta_{2c}^{s,ols}|, R_c^{s*}\big) - Q\big(z_{2c-1} - |\beta_{2c-1}^{s,ols}|, z_{2c} - |\beta_{2c}^{s,ols}|, R_c^{s*}\big) \\
= \; & \big(\tfrac{\theta_c^s - \alpha}{2}, \tfrac{\theta_c^s - \alpha}{2}\big) \begin{pmatrix} 1 & R_c^{s*} \\ R_c^{s*} & 1 \end{pmatrix} \begin{pmatrix} \frac{\theta_c^s - \alpha}{2} \\ \frac{\theta_c^s - \alpha}{2} \end{pmatrix} + 2\big(\tfrac{\theta_c^s - \alpha}{2}, \tfrac{\theta_c^s - \alpha}{2}\big) \begin{pmatrix} 1 & R_c^{s*} \\ R_c^{s*} & 1 \end{pmatrix} \begin{pmatrix} z_{2c-1} - |\beta_1^{s,ols}| \\ z_{2c} - |\beta_2^{s,ols}| \end{pmatrix} \\
= \; & 2(1 + R_c^{s*})\big(\tfrac{\theta_c^s - \alpha}{2}\big)^2 + 2(1 + R_c^{s*})\big(\tfrac{\theta_c^s - \alpha}{2}\big)\big\{z_{2c-1} + z_{2c} - |\beta_1^{s,ols}| - |\beta_2^{s,ols}|\big\} \\
< \; & (1 + R_c^{s*})(\theta_c^s - \alpha)\big\{\tfrac{\theta_c^s - \alpha}{2} + \alpha - \theta_c^s\big\} \\
= \; & -(1 + R_c^{s*})\big\{\frac{(\theta_c^s - \alpha)^2}{2}\big\} \\
< \; & 0
\end{aligned}
$$

Thus the only portion of the constraint that is not dominated according to this argument is the line segment $\Delta(\theta_c^s)$:

$$\Delta(\theta_c^s) = \{(z_1, z_2) | z_1 \geq 0, z_2 \geq 0, z_1 + z_2 = \theta_c^s\} \tag{60}$$

hence the solution to PZ: $(\hat{z}_{2c-1}, \hat{z}_{2c}) \in \Delta(\theta_c^s)$.

Finally we need to check that if $z_{2c-1} + z_{2c} = \alpha < \theta_c^s$ is inside the constraint set, then so is $(z_{2c-1} + \frac{\theta_c^s - \alpha}{2}, z_{2c} + \frac{\theta_c^s - \alpha}{2})$. The constraint $|z_{2c-1}| + |z_{2c}| \leq \theta_c^s$ can be rewritten as: $C1 : -\theta_c^s \leq z_{2c-1} + z_{2c} \leq \theta_c^s$ and $C2 : -\theta_c^s \leq z_{2c-1} - z_{2c} \leq \theta_c^s$. The pair $(z_{2c-1} + \frac{\theta_c^s - \alpha}{2}, z_{2c} + \frac{\theta_c^s - \alpha}{2})$ satisfy C1 by construction, while $z_{2c-1} + \frac{\theta_c^s - \alpha}{2} - (z_{2c} + \frac{\theta_c^s - \alpha}{2}) = z_{2c-1} - z_{2c}$ so that if $-\theta_c^s \leq z_{2c-1} - z_{2c} \leq \theta_c^s$ it follows that $\theta_c^s \leq z_{2c-1} + \frac{\theta_c^s - \alpha}{2} - (z_{2c} + \frac{\theta_c^s - \alpha}{2}) \leq \theta_c^s$

□

Proof of Lemma 3

Proof. Differentiating our expression for Q, (15), we have:

$$\frac{\partial Q}{\partial z_{2c-1}} = 2(z_{2c-1} - |\beta_{2c-1}|) + 2R_c^{s*}(z_{2c} - |\beta_{2c}|) \quad \text{and} \quad \frac{\partial Q}{\partial z_2} = 2(z_{2c} - |\beta_{2c}|) + 2R_c^{s*}(z_{2c-1} - |\beta_{2c-1}|)$$

substituting into (16) this yields:

$$\begin{aligned} \frac{dz_{2c}}{dz_{2c-1}} &= -\frac{2(z_{2c-1} - |\beta_{2c-1}|) + 2R_c^{s*}(z_{2c} - |\beta_{2c}|)}{2(z_{2c} - |\beta_{2c}|) + 2R_c^{s*}(z_{2c-1} - |\beta_{2c-1}|)} \\ &= -\frac{(z_{2c-1} - |\beta_{2c-1}|) + R_c^{s*}(z_{2c} - |\beta_{2c}|)}{(z_{2c} - |\beta_{2c}|) + R_c^{s*}(z_{2c-1} - |\beta_{2c-1}|)} \end{aligned}$$

□

Proof of Lemma 4

Proof. Substituting from (12) we have:

$$\begin{aligned} & Q(\theta_c^s - |\beta_{2c-1}^{s,ols}|, -|\beta_{2c}^{s,ols}|, R_c^*) \\ = & \left(\theta_c^s - |\beta_{2c-1}^{s,ols}|, -|\beta_{2c}^{s,ols}|\right) \begin{pmatrix} 1 & R_c^{s*} \\ R_c^{s*} & 1 \end{pmatrix} \begin{pmatrix} \theta_c^s - |\beta_{2c-1}^{s,ols}| \\ -|\beta_{2c}^{s,ols}| \end{pmatrix} \\ = & -2\theta_c^s(|\beta_{2c-1}^{s,ols}| + R_c^{s*}|\beta_{2c}^{s,ols}|) + \left(\theta_c^{s2} + \left(-|\beta_{2c-1}^{s,ols}|, -|\beta_{2c}^{s,ols}|\right) \begin{pmatrix} 1 & R_c^{s*} \\ R_c^{s*} & 1 \end{pmatrix} \begin{pmatrix} -|\beta_{2c-1}^{s,ols}| \\ -|\beta_{2c}^{s,ols}| \end{pmatrix}\right) \end{aligned} \tag{61}$$

likewise:

$$
\begin{aligned}
& Q(-|\beta^{s,ols}_{2c-1}|, \theta^s_c - |\beta^{s,ols}_{2c}|, R^*_c)) \\
= \; & \left(-|\beta^{s,ols}_{2c-1}|, \theta^s_c - |\beta^{s,ols}_{2c}|\right) \begin{pmatrix} 1 & R^{s*}_c \\ R^{s*}_c & 1 \end{pmatrix} \begin{pmatrix} -|\beta^{s,ols}_{2c-1}| \\ \theta^s_c - |\beta^{s,ols}_{2c}| \end{pmatrix} \\
= \; & -2\theta^s_c(R^{s*}_c|\beta^{s,ols}_{2c-1}| + |\beta^{s,ols}_{2c}|) + \left(\theta^{s2}_c + \left(-|\beta^{s,ols}_{2c-1}|, -|\beta^{s,ols}_{2c}|\right) \begin{pmatrix} 1 & R^{s*}_c \\ R^{s*}_c & 1 \end{pmatrix} \begin{pmatrix} -|\beta^{s,ols}_{2c-1}| \\ -|\beta^{s,ols}_{2c}| \end{pmatrix}\right) \quad (62)
\end{aligned}
$$

Calculating the difference between (61) and (62) we have:

$$
\begin{aligned}
& Q(\theta^s_c - |\beta^{s,ols}_{2c-1}|, -|\beta^{s,ols}_{2c}|, R^*_c) - Q(-|\beta^{s,ols}_{2c-1}|, \theta^s_c - |\beta^{s,ols}_{2c}|, R^*_c)) \\
= \; & 2\theta^s_c\left((|\beta^{s,ols}_{2c-1}| + R^{s*}_c|\beta^{s,ols}_{2c}|) - \left(R^{s*}_c(|\beta^{s,ols}_{2c-1}| + |\beta^{s,ols}_{2c}|)\right)\right) \\
= \; & 2\theta^s_c(1 - R^{s*}_c)\left(|\beta^{s,ols}_{2c}| - |\beta^{s,ols}_{2c-1}|\right)
\end{aligned}
$$

However $|R^{s*}_c| < 1$; recall that our data contain no perfectly correlated pairs. Likewise $\theta^s_c > 0$ by assumption, and so $2\theta^s_c(1 - R^{s*}_c) > 0$, hence we have:

$$
\begin{aligned}
& sign\left(Q(\theta^s_c - |\beta^{s,ols}_{2c-1}|, -|\beta^{s,ols}_{2c}|, R^*_c) - Q(-|\beta^{s,ols}_{2c-1}|, \theta^s_c - |\beta^{s,ols}_{2c}|, R^*_c))\right) \\
= \; & sign\left(|\beta^{s,ols}_{2c}| - |\beta^{s,ols}_{2c-1}|\right)
\end{aligned}
$$

□

Proof of Lemma 5

Proof. Considering cases for which $\lambda > 0$, at an interior solution the non-negativity constraints are not binding, so that $\mu_{2c-1} = \mu_{2c} = 0$. Differentiating (27) with respect to z_{2c-1}, z_{2c}, and λ we have:

$$
\begin{aligned}
\frac{\partial L}{\partial z_{2c-1}} &= 2(z_{2c-1} - |\beta^{s,ols}_{2c-1}|) + 2R_c(z_{2c} - |\beta^{s,ols}_{2c-1}|) + \lambda = 0 \\
\frac{\partial L}{\partial z_{2c}} &= 2R_c(z_{2c} - |\beta^{s,ols}_{2c-1}|) + 2(z_{2c-1} - |\beta^{s,ols}_{2c-1}|) + \lambda = 0 \\
\frac{\partial L}{\partial \lambda} &= z_{2c-1} + z_{2c} - \theta^s_c = 0 \quad (63)
\end{aligned}
$$

If we add the first two equations:

$$2(1+R_c)(z_{2c-1}-|\beta_{2c-1}^{s,ols}|)+2(1+R_c)(z_{2c}-|\beta_{2c}^{s,ols}|)+2\lambda=0$$

rearranging terms this becomes:

$$2(1+R_c)(z_{2c-1}+z_{2c})-2(1+R_c)|\beta_{2c-1}^{s,ols}|-2(1+R_c)|\beta_{2c}^{s,ols}|+2\lambda=0 \tag{64}$$

now substitute from $\frac{\partial L}{\partial \lambda}=0$ to obtain:

$$2(1+R_c)\theta_c^s-2(1+R_c)|\beta_{2c-1}^{s,ols}|-2(1+R_c)|\beta_{2c}^{s,ols}|+2\lambda=0 \tag{65}$$

solving for θ_c^s we have:

$$\theta_c^s=|\beta_{2c-1}^{s,ols}|+|\beta_{2c}^{s,ols}|-\frac{\lambda}{1+R_c}$$

□

Proof of Lemma 6

Proof. Turning to our first order conditions for (27) we require:

$$\begin{aligned}\frac{\partial L}{\partial z_{2c-1}} &= 2(z_{2c-1}-|\beta_{2c-1}^{s,ols}|)+2R_c^{s*}(0-|\beta_{2c}^{s,ols}|)+\lambda=0\\ \frac{\partial L}{\partial \lambda} &= z_{2c-1}+0-\theta_c^s=0\end{aligned} \tag{66}$$

Substituting z_{2c-1} from the third expression into the first and solving for θ_c^s yields:

$$\theta_c^s=|\beta_{2c-1}^{s,ols}|+R_c^{s*}|\beta_{2c}^{s,ols}|-\frac{\lambda}{2}$$

□

Proof of Lemma 7

Proof. Start with the update for $(\beta_{2c-1}^{s,ols},\beta_{2c}^{s,ols})$. We solve:

$$\min_{\beta_{2c-1}^*,\beta_{2c}^*}\sum_{i=1}^{n}\big(y_i-\sum_{j\le 2c-2}x_{j,i}\beta_j^{s,ols}-\sum_{j\ge 2c+1}x_{j,i}\beta_j^{s-1,ols}-x_{2c-1,i}\beta_{2c-1}^*-x_{2c,i}\beta_{2c}^*\big)^2$$

This leads to the following first order conditions:

$$
\begin{aligned}
-2\sum_{i=1}^{n}\big(y_i - \sum_{j\le 2c-2} x_{j,i}\beta_j^{s,ols} - \sum_{j\ge 2c+1} x_{j,i}\beta_j^{s-1,ols} - x_{2c-1,i}\beta_{2c-1}^{*} - x_{2c,i}\beta_{2c}^{*}\big)x_{2c-1,i} &= 0\\
-2\sum_{i=1}^{n}\big(y_i - \sum_{j\le 2c-2} x_{j,i}\beta_j^{s,ols} - \sum_{j\ge 2c+1} x_{j,i}\beta_j^{s-1,ols} - x_{2c-1,i}\beta_{2c-1}^{*} - x_{2c,i}\beta_{2c}^{*}\big)x_{2c,i} &= 0
\end{aligned}
$$

Using our newly developed notation we can rewrite these conditions as:

$$
\begin{aligned}
-2\big(r_{y,2c-1} - \sum_{j\le 2c-2} r_{j,2c-1}\beta_j^{s,ols} - \sum_{j\ge 2c+1} r_{j,2c-1}\beta_j^{s-1,ols} - \beta_{2c-1}^{*} - r_{2c-1,2c}\beta_{2c}^{*}\big) &= 0\\
-2\big(r_{y,2c} - \sum_{j\le 2c-2} r_{j,2c}\beta_j^{s,ols} - \sum_{j\ge 2c+1} r_{j,2c}\beta_j^{s-1,ols} - r_{2c-1,2c}\beta_{2c-1}^{*} - \beta_{2c}^{*}\big) &= 0
\end{aligned}
$$

That is:

$$
\begin{pmatrix} 1 & r_{2c-1,2c} \\ r_{2c-1,2c} & 1 \end{pmatrix}
\begin{pmatrix} \beta_{2c-1}^{*} \\ \beta_{2c}^{*} \end{pmatrix}
=
\begin{pmatrix} r_{y,2c-1} - \sum_{j\le 2c-2} r_{j,2c-1}\beta_j^{s,ols} - \sum_{j\ge 2c+1} r_{j,2c-1}\beta_j^{s-1,ols} \\ r_{y,2c} - \sum_{j\le 2c-2} r_{j,2c}\beta_j^{s,ols} - \sum_{j\ge 2c+1} r_{j,2c}\beta_j^{s-1,ols} \end{pmatrix}
$$

This simplifies to:

$$
\begin{aligned}
\begin{pmatrix} \beta_{2c-1}^{*} \\ \beta_{2c}^{*} \end{pmatrix}
&= \frac{1}{1-r_{2c,2c-1}^2}
\begin{pmatrix} 1 & -r_{2c-1,2c} \\ -r_{2c-1,2c} & 1 \end{pmatrix}^{-1}
\begin{pmatrix} r_{y,2c-1} - \sum_{j\le 2c-2} r_{j,2c-1}\beta_j^{s,ols} - \sum_{j\ge 2c+1} r_{j,2c-1}\beta_j^{s-1,ols} \\ r_{y,2c} - \sum_{j\le 2c-2} r_{j,2c}\beta_j^{s,ols} - \sum_{j\ge 2c+1} r_{j,2c}\beta_j^{s-1,ols} \end{pmatrix}\\
&= \begin{pmatrix} \vec{s}'_{2c-1}\vec{\alpha}^{s,c,ols} \\ \vec{s}'_{2c}\vec{\alpha}^{s,c,ols} \end{pmatrix}
\end{aligned}
$$

But this is simply (45). Similar, and even more straightforward calculations show that (46) corresponds to the solution for $P3_c^s$.

□

Proof of Lemma 8: Starting with $a > c$, which we have by assumption, we see that $a - b > c - b$, and so, provided $c > b$, we divide by $c - b$ to obtain the left inequality in (bothsteep). Similarly, $a > c$ implies $a + b > c + b$, and so, provided $b < c$, we can divide both sides by $-(c + b)$ to obtain the right inequality of (bothsteep). □

Proof of Lemma 9: Starting with $b < -c$, expression (bbounded) tells us that $a + b > 0$, while by assumption $a > c$, hence we have:

$ac - b^2 > b^2 - ac$

$$ac + (bc - ab) - b^2 > b^2 + (bc - ab) - ac$$

$$(a + b)(c - b) > -(c + b)(a - b)$$

$$a + b > -(c + b)\frac{a - b}{c - b}$$

$$-\left(\frac{a + b}{c + b}\right) > \left(\frac{a - b}{c - b}\right)$$

which corresponds to the righthand inequality in (plus45Steeper). Moreover, given that $a > c$ while $b < 0$ we also know that $0 < c - b < a - b$, dividing by $c - b > 0$ gives us the lefthand inequality in (plus45Steeper). □

Proof of Lemma 10: Starting with $c < b$, we have:

$b^2 - ac < ac - b^2$

$$b^2 + (ba - bc) - ac < ac + (ba - bc) - b^2$$

$$-(c - b)(a + b) < (c + b)(a - b)$$

$$-(a + b) > (c + b)\frac{a - b}{c - b}$$

$$-\left(\frac{a + b}{c + b}\right) > \left(\frac{a - b}{c - b}\right)$$

which corresponds to (minus45Steeper). Moreover, given that $a > c$ while $b > 0$ we also know that $0 < c + b < a + b$ and so $-\frac{a+b}{c+b} < -1$. □

Appendix B: Extension to Weighted Least Squares

To this point we have examined the bicoordinate solution to cases of ordinary least squares. Our extension to the probit model first transformed the problem *via* the EM algorithm into a sequence of m-steps, each of which was tantamount to estimating a least squares regression. Now we consider the more general case in which we confront a weighted least squares problem, this encompasses the solution developed in the preceding sections as an important special case. Weighted least squares leads to different weights on each observation, and these differences in turn lead to different weights on the first order conditions for different coefficients. This complicates the solution, but does not render it intractable.

Formulation

Now we generalize problem $P2_c^{s\prime}$. Suppose we seek a solution to the following problem:

$$\min_{\vec{b}}(\vec{b}-\vec{\beta})'\mathcal{I}(\vec{b}-\vec{\beta}) \text{ subject to } \|\vec{b}\|_1 \leq T \tag{solveOriginal}$$

with $T > 0$ while:

$$\vec{b} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \text{ and } \vec{\beta} = \begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix} \tag{67}$$

where $\|\vec{b}\|_1 = |b_1| + |b_2|$ is the l_1 norm and

$$\mathcal{I} = \begin{pmatrix} I_{11} & I_{12} \\ I_{12} & I_{22} \end{pmatrix} \tag{68}$$

is a symmetric, positive definite matrix. When the diagonal terms of I are identical, this configuration coincides with the quadratic form in expression (12), where $I_{11} = I_{22} = 1$ and $I_{12} = I_{21} = R_c^{s*}$.

For convenience, rather than focusing on `solveOriginal` it is preferable to work with the following "canonical" problem:

$$\min_{x,y} f(x,y) \text{ subject to } |x| + |y| \leq T \tag{solveQuad}$$

where $T > 0$ and:

$$f(x,y) = (\vec{z}-\vec{z}_0)'Q(\vec{z}-\vec{z}_0) \tag{69}$$

while:

$$\vec{z} = \begin{pmatrix} x \\ y \end{pmatrix} \text{ and } \vec{z}_0 = \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} \tag{70}$$

with Q a symmetric, positive definite matrix:

$$Q = \begin{pmatrix} a & b \\ b & c \end{pmatrix} \tag{Qdef}$$

Notice that positive definiteness of Q implies that

$$ac - b^2 > 0 \tag{71}$$

Likewise, for the canonical problem we require:

$$x_0 > 0 \; y_0 > 0 \; x_0 + y_0 \geq T \; a \geq c > 0 \tag{72}$$

We form the link between ($solveQuad$) and ($solveOriginal$) as follows. Let's define the matrices S:

$$S = \begin{pmatrix} sign(\beta_1) & 0 \\ 0 & sign(\beta_2) \end{pmatrix}$$

and T:

$$T = \begin{pmatrix} \delta(I_{22} \leq I_{11}) & \delta(I_{11} < I_{22}) \\ \delta(I_{11} < I_{22}) & \delta(I_{22} \leq I_{11}) \end{pmatrix}$$

where δ is the Dirac function: $\delta(TRUE) = 1$ and $\delta(FALSE) = 0$.

Notice that $(ST)^{-1} = TS$, so we can rewrite ($solveOriginal$) as follows:

$$\min_{\vec{b}} (\vec{b} - \vec{\beta})' STTS\mathcal{I}STTS(\vec{b} - \vec{\beta}) \text{ subject to } |b_1| + |b_2| \leq T$$

Next, if we let $Q = TS\mathcal{I}ST$, while $\vec{z} = TS\vec{b}$ and $\vec{z}_0 = TS\vec{b}_0$ we have $|b_1| + |b_2| = |x| + |y|$, and so our problem coincides with ($solveQuad$). To be sedulously clear, if we start with ($solveOriginal$), we can reach the corresponding version of ($solveQuad$) by setting $b = sign(\beta_1) \times sign(\beta_2) \times I_{12}$, while if $I_{11} \geq I_{22}$ then $x_0 = sign(\beta_1)\beta_1$, $y_0 = sign(\beta_2)\beta_2$, $a = I_{11}$ and $c = I_{22}$, whereas if $I_{11} < I_{22}$ then $x_0 = sign(\beta_2)\beta_2$, $y_0 = sign(\beta_1)\beta_1$, $a = I_{22}$ and $c = I_{11}$.

Notice that once we have solved ($solveQuad$) to obtain (x, y) we can go back to the solution for ($solveOriginal$); (b_1, b_2) using the transformation $\vec{b} = ST\vec{z}$. In more excruciating detail, if $I_{22} \leq I_{11}$ we have $b_1 = sign(\beta_1)x$, and $b_2 = sign(\beta_2)y$, while if $I_{11} < I_{22}$ we have $b_1 = sign(\beta_1)y$, and $b_2 = sign(\beta_2)x$.

Preliminaries

Along the isoquant $f(x, y) = k$ we have:

$$\left.\frac{\partial y}{\partial x}\right|_{f=k} = -\left(\frac{a(x - x_0) + b(y - y_0)}{b(x - x_0) + c(y - y_0)}\right) \tag{73}$$

for any point (x, y) at which the isoquant of f has a slope of -1 we have:

$$\frac{\partial y}{\partial x}\Big|_{f=k} = -1 \;\rightarrow\; (y - y_0) = \left(\frac{a-b}{c-b}\right)(x - x_0)$$

the lefthand panel of figure 7 shows several such points, the open circle corresponds to (x_0, y_0). In fact, the set of all points at which the isoquant of f has a slope of -1 forms a straight line:

$$y = \left(y_0 - \left(\frac{a-b}{c-b}\right)x_0\right) + \left(\frac{a-b}{c-b}\right)x \qquad \text{(minus45)}$$

see the central panel of figure 7 where (minus45) corresponds to the line labeled τ^-. This line is of some practical use: if there is a tangency between an isoquant of f and the first quadrant constraint it will occur at the intersection of the line (minus45) with the first quadrant edge of the LASSO constraint, the line segment on which $x \in [0, T]$ and $y = T - x$. This outcome is depicted in the righthand panel of figure 7. Conversely, if (minus45) fails to intersect the northeast edge of the LASSO constraint, then there will not be a first quadrant tangency.

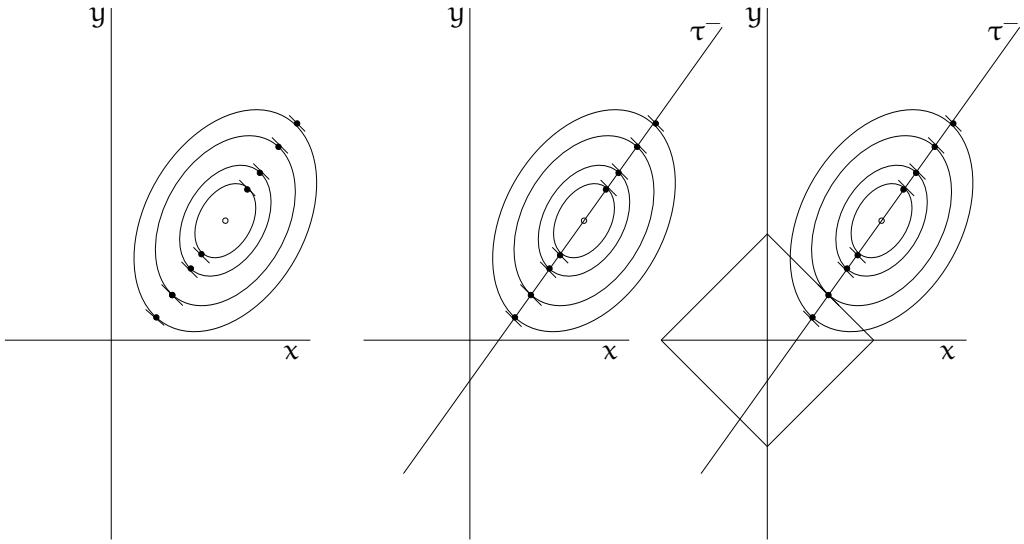


Figure 7: Right: Tangencies Center: minus45 Left: Interior Solution

We can likewise construct the set of points at which the the isolevels of f have slope 1:

$$\frac{\partial y}{\partial x}\Big|_{f=k} = 1 \;\rightarrow\; (y - y_0) = -\left(\frac{a+b}{c+b}\right)(x - x_0)$$

which comprise the following line:

$$y = \left(y_0 + \left(\frac{a+b}{c+b}\right)x_0\right) - \left(\frac{a+b}{c+b}\right)x \qquad \text{(plus45)}$$

We shall see that if (plus45) intersects the second quadrant edge of the LASSO constant, then we will have an interior solution in the second quadrant.

Some Useful Results

Notice that the slope of (plus45) is guaranteed to be negative unless $-a < b < -c$. In contrast, (minus45) slopes upwards unless $c < b < a$.

However, because by (71), we have $|b| < \sqrt{ac}$ where $\sqrt{ac}$ denotes the positive square root of ac. So $-b < \sqrt{ac}$, but the geometric mean is less than the arithmetic mean, hence:

$$-b < \sqrt{ac} \leq \frac{a+c}{2} \leq a \tag{bbounded}$$

and so it follows that (plus45) will not slope upwards unless:

$$c < -b \tag{notQ1a}$$

In contrast, we know that (minus45) will have a positive slope unless:

$$c < b \tag{notQ1b}$$

For future reference, let's consolidate these claims as:

Lemma 8: Provided that $|b| < c$, the line (minus45) slopes upward, with a slope in excess of 1, while (plus45) does not, exhibiting a slope more negative than -1:

$$1 < \left(\frac{a-b}{c-b}\right) \text{ and } -\left(\frac{a+b}{c+b}\right) < -1 \tag{bothsteep}$$

Likewise, we have:

Lemma 9: If $b < -c$ the (plus45) line slopes upward more steeply than (minus45):

$$1 < \left(\frac{a-b}{c-b}\right) < -\left(\frac{a+b}{c+b}\right) \tag{plus45Steeper}$$

Lemma 10: If instead $c < b$ the (minus45) line slopes downward more steeply than (plus45):

$$\left(\frac{a-b}{c-b}\right) < -\left(\frac{a+b}{c+b}\right) < -1 \tag{minus45Steeper}$$

Solutions

Solutions to (solveQuad) divide into several categories, and we organize them relative to the value of b. Let's consider them in turn.

$b < -c$

Conditional on (x_0, y_0) the solution to the LASSO problem will be unique provided Q has full rank, but when we have an extremely low value[9] of b there are no fewer than five potential and qualitatively different forms this solution might take on. The set of (x_0, y_0) values leading to each solution type are depicted in the right panel of figure 10, these include two interior solutions, the OLS estimates leading to LASSO estimates in first the first quadrant are marked IQ1, while those corresponding to a second quadrant interior solution are denoted IQ2. The set of values leading to a corner solution at $(T, 0)$ are labeled R, another set, marked T leads to a LASSO solution at $(0, T)$, while the set of (x_0, y_0) corresponding to the region labeled L correspond to a LASSO outcome at $(-T, 0)$. Let's work through these cases one at a time.

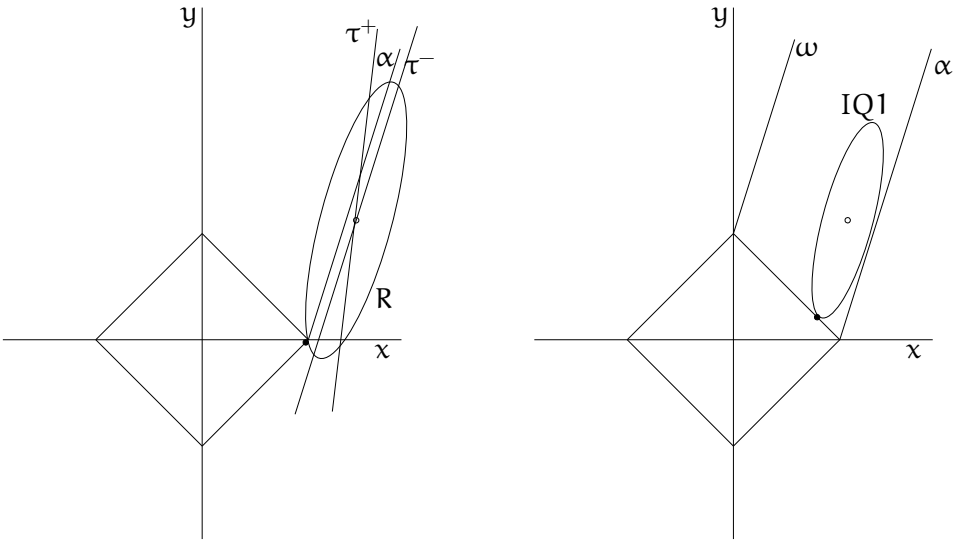


Figure 8: A Right Corner Solution A First Quadrant Interior Solution

[9]Notice that these cases can only arise when $c < a$, for if they have the same magnitude $c < b$ would violate the positive definiteness condition for Q.

A Solution at $(T, 0)$

With $b < -c$ we will have a corner solution at $(T, 0)$ when (x_0, y_0), denoted by a small open circle in the left panel of figure 8, is to the right of the line (alpha):

$$y = \frac{a-b}{c-b}(x - T) \tag{alpha}$$

Notice that (alpha), labeled α in figure 8, passes through $(T, 0)$ with slope $\frac{a-b}{c-b}$.

Algebraically, we can express (x_0, y_0) being to the right of (alpha) as:

$$y_0 \leq \frac{a-b}{c-b}(x_0 - T) \tag{rightB}$$

The region to the right of the α line is labeled R in the figure, the set of all possible tangencies between level curves of f and the first quadrant portion of the LASSO constraint comprise the line (minus45):

$$y = T - x \text{ with } 0 \leq x \leq T \tag{Quad1}$$

denoted in figure 8 by the line τ^-. Because τ^- is parallel to α, it never intersects the constraint diamond for the LASSO, precluding a first quadrant tangency.

The set of potential second and fourth quadrant tangencies, corresponding to expression (plus45), constitute the line τ^+. This line also passes through (x_0, y_0), and by Lemma 9 it is even more steeply sloped than τ^-, so it also misses the LASSO constraint diamond.

Thus we have no interior solution. Amongst the possible corner solutions, and for any (x_0, y_0) pair in region R, the right corner minimizes the loss of meeting the LASSO constraint; the ellipse centered on (x_0, y_0) that just grazes $(T, 0)$ represents the lowest attainable loss relative to OLS for any estimator in the LASSO diamond.

An Interior Solution in the First Quadrant

With $b < -c$ we will have a tangency in the first quadrant provided also that the line defined in (minus45) intersects the constraint in the first quadrant, that is, along the line:

$$y = T - x$$

with $0 \leq x \leq T$. This will happen when (x_0, y_0) lies between the line with positive slope $\frac{a-b}{c-b}$ passing through $(T, 0)$, marked α in the righthand panel of 8, and the parallel line that includes $(0, T)$, marked ω on the right of figure 8. Algebraically this occurs when:

$$\frac{a-b}{c-b}(x_0 - T) < y_0 < T + \frac{a-b}{c-b}x_0 \tag{IQ1}$$

in which case the solution will consist of:

$$\begin{pmatrix} x^* \\ y^* \end{pmatrix} = \begin{pmatrix} \frac{(c-b)T+(a-b)x_0-(c-b)y_0}{a+c-2b} \\ \frac{(a-b)T-(a-b)x_0+(c-b)y_0}{a+c-2b} \end{pmatrix} \tag{solveIQ1}$$

The right hand panel of figure 8 illustrates such a case–the OLS estimate corresponds to the open dot, while the ellipses are the level curves for f, the residual sum of squares.

A Solution at $(0, T)$

Still considering the case in which $b < -c$ we arrive at a corner solution at the "top" vertex, $(0, T)$, when (x_0, y_0) lies above the line with slope $\frac{a-b}{c-b}$ that passes through $(0, T)$, but below the line passing through the same vertex with slope $-\frac{a+b}{c+b}$:

$$T + \frac{a-b}{c-b}x_0 \leq y_0 \leq T - \left(\frac{a+b}{c+b}\right)x_0 \tag{topcorner}$$

Pairs (x_0, y_0) satisfying (topcorner) correspond to the region labeled T in figure 10, and tangencies with the constraint diamond are incompatible–the locus of potential tangencies with the left upper face of the diamond that encompasses these points lies too far to the right to intersect the upper left face of the diamond, while the potential tangencies with the righthand upper face of the constraint pass to the left of that face. These points can only result in a corner solution, and a quick check confirms that the least onerous element of the constraint is the corner at $(0, T)$.

An Interior Solution in the Second Quadrant

Still taking as given that $b < -c$, we will have an interior solution in the second quadrant provided (x_0, y_0) lies above the line with slope $-\frac{a+b}{c+b}$ that passes through $(0, T)$, but below the parallel line passing through $(-T, 0)$. In this case the solution will be found at the intersection of the line given in (plus45) with the second quadrant portion of the constraint:

$$T - \left(\frac{a+b}{c+b}\right)x_0 < y_0 < -\left(\frac{a+b}{c+b}\right)(x_0 + T) \tag{IQ2}$$

which crossing occurs at:

$$\begin{pmatrix} x^* \\ y^* \end{pmatrix} = \begin{pmatrix} \frac{-(c+b)T+(a+b)x_0+(c+b)y_0}{a+c+2b} \\ \frac{(a+b)T+(a+b)x_0+(c+b)y_0}{a+c+2b} \end{pmatrix} \tag{solveIQ2}$$

A Solution at $(-T, 0)$

Finally, and still with $b < -c$, we are led to a corner solution at the "left" vertex, $(-T, 0)$, when (x_0, y_0) lies above the line with slope $-\frac{a+b}{c+b}$ that passes through $(-T, 0)$:

$$-\left(\frac{a+b}{c+b}\right)(x_0 + T) \leq y_0 \tag{leftcorner}$$

$-c < b < c$

There are only three qualitatively distinct types of solution in this case, an interior solution in the first quadrant, a corner solution along the x axis, and a corner solution along the y axis. Notice that the ordinary least squares model satisfies this condition, as the combination of $a = c$ and $ac - b^2 > 0$ guarantee that $-c < b < c$. Let's first turn to the interior solution:

An Interior Solution in the First Quadrant

With $|b| < c$ we will have a tangency in the first quadrant provided we satisfy (IQ1), leading to a tangency at (*solveIQ1*).

A Corner Solution at $(T, 0)$

With $|b| < c$ we have a right corner solution at:

$$(x^*, y^*) = (T, 0) \tag{solverightcorner}$$

provided (x_0, y_0) lies below[10] the line passing through $(T, 0)$ with slope $\frac{a-b}{c-b}$, a condition already summarized as (*rightB*).

A Corner Solution at $(0, T)$

Again in the case of $|b| < c$ we have a top corner solution when the line with positive inclination $\frac{a-b}{c-b}$ that includes $(0, T)$ passes below (x_0, y_0), given above as the first of the two inequalities in condition

[10]Recall that by assumption (x_0, y_0) are in the closure of the first quadrant.

(topcorner).

$c < b$

There are but two qualitatively different solution classes corresponding to extremely high values[11] of b, an interior solution in the first quadrant, and a corner solution at $(T, 0)$.

A Corner Solution at $(T, 0)$

With $c < b$, consistent with Lemma 10, the set of possible first quadrant tangencies forms a negatively sloped line, denoted by τ^- in the lefthand panel of figure 11, while the set of possible second quadrant tangencies constitute an even more steeply sloped line, labeled τ^+ in figure 11. The line passing through $(T, 0)$ and parallel to τ^- marks the boundary of the set of (x_0, y_0) pairs that correspond to a corner solution. All of the (x_0, y_0) pairs above and to the right of this line, recall that by assumption (x_0, y_0) are in the first quadrant, correspond to a corner solution, as is depicted in the righthand panel of the figure. The formal condition for this is given by expression (rightbplus):

$$-\frac{a-b}{c-b}(T - x_0) \leq y_0 \qquad \text{(rightbplus)}$$

this region of the first quadrant, corresponding to solutions at the right corner of the constraint set, is labeled R in the diagram.

A First Quadrant Tangency

The lefthand panel of figure 11 depicts a typical tangency. Elements of the scalene triangle marked IQ1 correspond to interior solutions in quadrant one. Formally for the case of $b > c$ we have a first quadrant tangency when:

$$y_0 < -\frac{a-b}{c-b}(T - x_0) \qquad \text{(tangent1b)}$$

Notice that there are no corner solutions at $(0, T)$, save for the trivial case in which $(x_0, y_0) = (0, T)$, a solution that is encompassed among the tangencies. This is a byproduct of the steepness of the τ^{-1} curve, which slopes more steeply than the first quadrant constraint, and so even points

[11] Notice that these cases can only arise when $c < a$, for if they have the same magnitude $c < b$ would violate the positive definiteness condition for Q.

along the y axis above $(0, T)$ lead to interior solutions, provided y_0 is not too large, or to right corner solutions when y_0 exceeds $-\frac{a-b}{c-b}(T - x_0)$.

Connecting with λ

We now have a comprehensive solution for (x, y) in terms of (x_0, y_0, T). Whereas what we need is to solve in terms of (x_0, y_0, λ). Our attention now turns to making the connection. Firstly let's revisit ($solveQuad$) and formulate the Lagrangian:

$$\mathcal{L} = f(x, y) + \lambda\big(|x| + |y| - T\big) \tag{74}$$

The first order conditions for a maximum are:

$$\frac{\partial}{\partial x}\mathcal{L} = f_x(x, y) + \lambda sign(x) = 0$$

$$\frac{\partial}{\partial y}\mathcal{L} = f_y(x, y) + \lambda sign(y) = 0$$

which leaves us with:

$$\begin{pmatrix} a & b \\ b & c \end{pmatrix} \begin{pmatrix} x - x_0 \\ y - y_0 \end{pmatrix} = -\frac{\lambda}{2} \begin{pmatrix} sign(x) \\ sign(y) \end{pmatrix} \tag{75}$$

At an interior solution we have:

$$\hat{x} = x_0 - \frac{\lambda}{2}\left(\frac{csign(x) - bsign(y)}{ac - b^2}\right) \tag{interiorx}$$

and:

$$\hat{y} = y_0 - \frac{\lambda}{2}\left(\frac{-bsign(x) + asign(y)}{ac - b^2}\right) \tag{interiory}$$

Corner solutions require a bit of extra care. Consider a solution at the right corner. We know that the derivative of the objective function with respect to T will equal $-\lambda$, that is:

$$\frac{\partial f}{\partial T}(T, 0) = -\lambda$$

Computing the resulting derivative and solving for T we have:

$$T = x_0 + \frac{b}{a}y_0 - \frac{\lambda}{2a} \tag{RightX}$$

and so our solution pair $(\hat{x}, \hat{y})$ is:

$$(\hat{x}, \hat{y}) = \left(x_0 + \frac{b}{a}y_0 - \frac{\lambda}{2a}, 0\right) \tag{rightex}$$

For a solution at the left corner we must have:

$$\frac{\partial f}{\partial T}(-T, 0) = -\lambda$$

which leads to:

$$T = -x_0 - \frac{b}{a}y_0 - \frac{\lambda}{2a} \tag{LeftX}$$

and so the solution pair will be:

$$(\hat{x}, \hat{y}) = \left(x_0 - \frac{b}{a}y_0 - \frac{\lambda}{2a}, 0\right) \tag{leftex}$$

Likewise, at a top corner solution we have:

$$T = \frac{b}{c}x_0 + y_0 - \frac{\lambda}{2c} \tag{TopY}$$

leading to $(\hat{x}, \hat{y})$:

$$(\hat{x}, \hat{y}) = \left(0, \frac{b}{c}x_0 + y_0 - \frac{\lambda}{2c}\right) \tag{topwhy}$$

Naturally, these solutions only make sense when $T \geq 0$. When this condition is violated we will be at the degenerate special case of a corner: $(x, y) = (0, 0)$.

So, couldn't we have skipped the preceding pages and simply jumped to expressions ($\mathtt{interiorx}$), ($\mathtt{interiory}$), ($\mathtt{rightex}$), ($\mathtt{leftex}$) and ($\mathtt{topwhy}$)? Yes, provided we could have correctly guessed the values for $\mathtt{sign}(x)$ and $\mathtt{sign}(y)$. The real point of the preceding pages was to derive the conditions for both types of interior solution and for each of the three potential corner solutions, and these we have in terms of T. We need them relative to λ, and so, by a straightforward process of equating our earlier solutions for (x, y) expressed in terms of T with our new ones, expressed relative to λ we now solve for T in terms of (x_0, y_0, λ) for each of our special cases.

We already have expressions for T in the case of our corner solutions: ($\mathtt{RightX}$), ($\mathtt{LeftX}$), ($\mathtt{TopY}$). The interior solutions are straightforward. At a first quadrant interior solution the constraint that $x + y \leq T$ is binding, so starting with ($\mathtt{interiorx}$) and ($\mathtt{interiory}$) we can substitute to find T:

$$T = x_0 + y_0 - \frac{\lambda}{2}\left(\frac{a + c - 2b}{ac - b^2}\right) \tag{TIQ1}$$

likewise, at an interior solution in the second quadrant we will have $-x + y = T$, substituting from (interiorx) and (interiory) yields:

$$T = -x_0 + y_0 - \frac{\lambda}{2}\left(\frac{a + c + 2b}{ac - b^2}\right) \tag{TIQ2}$$

$b < -c$

Now let's revisit the five ranges of solution that can emerge in this case, connecting our earlier analysis in terms of T with the Lagrange multiplier λ. Figure 12 provides a graphical depiction of the link between (x_0, y_0) and the type of solution we will encounter when $b < -c$. Let's consider these in turn. The label "IQ1" indicates an interior solution in the first quartile, while "IQ2" corresponds to a second quartile interior solution. The labels "Left", "Top", and "Right" indicate the corner solution that will emerge for the indicated set of (x_0, y_0) pairs. The region labeled "Origin" corresponds to the corner solution at which both $\hat{x}$ and $\hat{y}$ are equal to zero.

Right Corner

Substituting from (RightX) into (rightB) and simplifying we have:

$$y_0 < \frac{a - b}{ac - b^2}\frac{\lambda}{2} \tag{76}$$

we must also confirm that T is positive, substituting from (RightX) this condition becomes:

$$y_0 < \frac{\lambda}{2b} - \frac{a}{b}x_0 \tag{TPRlambda}$$

So, when $b < -c$ and we satisfy both conditions (76) and (TPRlambda) we have a corner solution given by (rightex).

Interior Solution: First Quadrant

Substituting from (TIQ1) into (IQ1) and simplifying we have:

$$\frac{\lambda}{2}\frac{c - b}{ac - b^2} < x_0 \text{ and } \frac{\lambda}{2}\frac{a - b}{ac - b^2} < y_0 \tag{IQ1lambda}$$

Inspection of (TIQ1) reveals that if we satisfy both of these conditions, T is guaranteed to be nonnegative, so when $b < -c$, (IQ1lambda) fully characterizes the sufficient conditions for an interior solution with $\hat{x}$ given by (interiorx) and $\hat{y}$ corresponding to (interiory):

$$(\hat{x}, \hat{y}) = \{x_0 - \frac{\lambda}{2}(\frac{c-b}{ac-b^2}), y_0 - \frac{\lambda}{2}(\frac{a-b}{ac-b^2})\} \tag{IQ1Sol}$$

Top Corner

Now we start with (topcorner), and replace T with our expression in (TopY). After a little manipulation, this expression becomes:

$$-\frac{c+b}{ac-b^2}\frac{\lambda}{2} \leq x_0 \leq \frac{c-b}{ac-b^2}\frac{\lambda}{2} \tag{TClambda}$$

Expression (TopY) reveals that to ensure $T > 0$ we need:

$$y_0 > \frac{\lambda}{2c} - \frac{b}{c}x_0 \tag{TPTlambda}$$

This tells us that we will have a corner solution given by (topwhy) if and only if we satisfy both (TClambda) and (TPRlambda).

Interior Solution: Second Quadrant

In the case of an interior solution in the second quadrant we substitute from (TIQ2) into (IQ2). When the chalk dust settles, we are left with:

$$x_0 < -\frac{c+b}{ac-b^2}\frac{\lambda}{2} \text{ and } \frac{\lambda}{2}\frac{a+b}{ac-b^2} < y_0 \tag{IQ2lambda}$$

As with the conditions for an interior solution in the first quadrant, given by (IQ1lambda), satisfying (IQ2lambda) is sufficient to guarantee $T > 0$, leaving us with $\hat{x}$ given by (interiorx) and $\hat{y}$ as in expression (interiory).

$$(\hat{x}, \hat{y}) = \{x_0 + \frac{\lambda}{2}(\frac{c+b}{ac-b^2}), y_0 - \frac{\lambda}{2}(\frac{a+b}{ac-b^2})\} \tag{IQ2Sol}$$

Left Corner

Now we substitute (LeftX) into (leftcorner). The resulting expression simplifies to:

$$y_0 < \frac{a+b}{ac-b^2}\frac{\lambda}{2} \tag{77}$$

We will have $T > 0$ whenever:

$$y_0 > -\frac{\lambda}{2b} - \frac{a}{b}x_0 \tag{TPLlambda}$$

so, when $b < c$ conditions (77) and (TPLlambda) are jointly necessary and sufficient for us to have a corner solution characterized by (leftex).

Finally, we will have a solution at the origin, with $\hat{x}, \hat{y}) = (0, 0)$ if and only if we simultaneously **fail** to satisfy (TPRlambda), (TPTlambda), and (TPLlambda). This condition can be expressed as:

$$\frac{\lambda}{2b} - \frac{a}{b}x_0 \leq y_0 \leq \min\{-\frac{\lambda}{2b} - \frac{a}{b}x_0, \frac{\lambda}{2c} - \frac{b}{c}x_0\} \tag{78}$$

Mediocre b Values

The case $|b| < c$ encompasses least squares regression with homoscedastic errors, and it contains fewer cases than the more complicated situation that confronts us when $b < -c$. It is helpful to refer to figure 13, which depicts $0 < b < c$. In particular, with $|b| < c$, (IQ1lambda) is necessary and sufficient for an interior solution, which will correspond to expression (IQ1Sol).

For a top corner solution, with $(\hat{x}, \hat{y})$ as given by (topwhy), the righthand inequality in expression (TClambda):

$$x_0 \leq \frac{c-b}{ac-b^2}\frac{\lambda}{2}$$

and:

$$\frac{\lambda}{2c} - \frac{b}{c}x_0 < y_0 \tag{79}$$

provide necessary and sufficient conditions. The latter of these two conditions, (79) guarantees that $T > 0$, which we have when we encounter (x_0, y_0) above:

$$y = \frac{\lambda}{2c} - \frac{b}{c}x \tag{80}$$

For a right corner solution, in which $(\hat{x}, \hat{y})$ is given by (rightex), it is necessary and sufficient for us simultaneously to satisfy (76) and (81):

$$\frac{\lambda}{2a} - \frac{b}{a}y_0 \leq x_0 \tag{81}$$

The latter condition guarantees that (x_0, y_0) is to the right of the line:

$$y = \frac{\lambda}{2b} - \frac{a}{b}x \tag{82}$$

Finally, we will have a solution at $(\hat{x}, \hat{y}) = (0, 0)$ when (x_0, y_0) satisfy:

$$y_0 \leq \frac{\lambda}{2c} - \frac{b}{c}x_0 \text{ and } x_0 \leq \frac{\lambda}{2a} - \frac{b}{a}y_0 \tag{83}$$

When $b \in (-c, 0)$ the analytics are the same, but the lines corresponding to (80) and to (82) each slope upward instead of downward as they do in figure 13.

$b > c$

When we encounter large positive values, for b, that is $b > c$, matters become starkly simple, see figure 14 for a graphical representation.

Let's start by substituting T from (RightX) into expression (rightbplus). This yields:

$$y_0 \leq \frac{a-b}{ac-b^2}\frac{\lambda}{2} \tag{RightAnswer}$$

For there to be a right corner solution we also require that $T > 0$, which, substituting from (RightX) leaves us with:

$$\frac{\lambda}{2a} - \frac{b}{a}y_0 \leq x_0 \tag{RightBigB}$$

In this case we have a corner solution with $\hat{x}$ given by (rightex), while $\hat{y} = 0$.

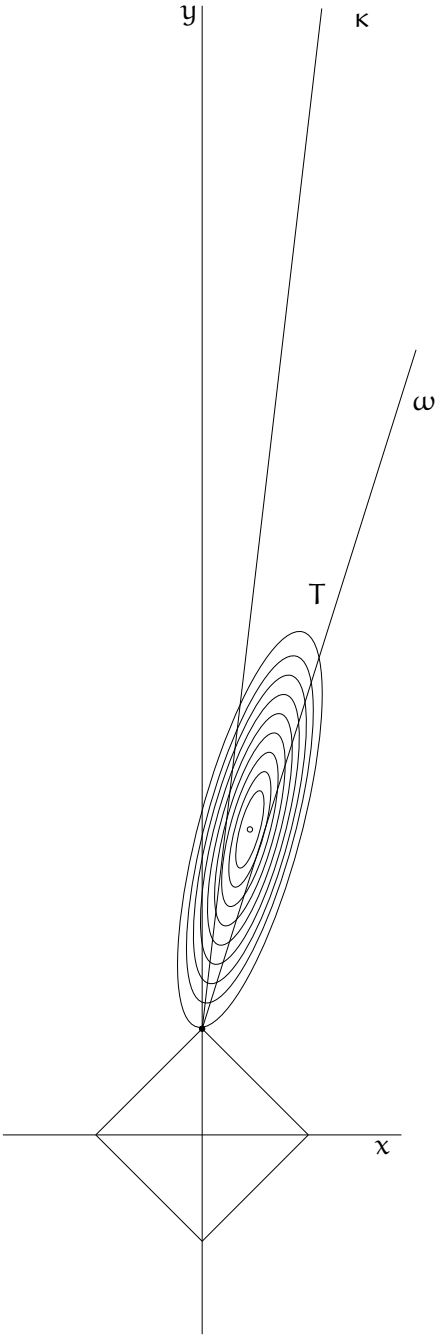
On the other hand, for an interior solution at (IQ1Sol), we must *fail* condition (RightBigB) while at the same time we have $T > 0$. Substituting from (ΠQ1) this second condition for an interior solution becomes:

$$\frac{\lambda}{2}\frac{a+c-2b}{ac-b^2} - x_0 < y_0 \tag{84}$$

We will have a solution at the origin when:

$$y_0 \leq \min\{\frac{\lambda}{2}\frac{a+c-2b}{ac-b^2} - x_0, \frac{\lambda}{2b} - \frac{a}{b}x_0\} \tag{85}$$

Glancing at 14 notice the long shared border between the region marked IQ1, corresponding to (x_0, y_0) pairs that lead to an interior solution, and the region indicated by the self explanatory label Origin. Notice also that the boundary of the set of (x_0, y_0) pairs that lead to a solution at the origin has a kink at $y_0 = \frac{\lambda}{2}\frac{a-b}{ac-b^2}$.



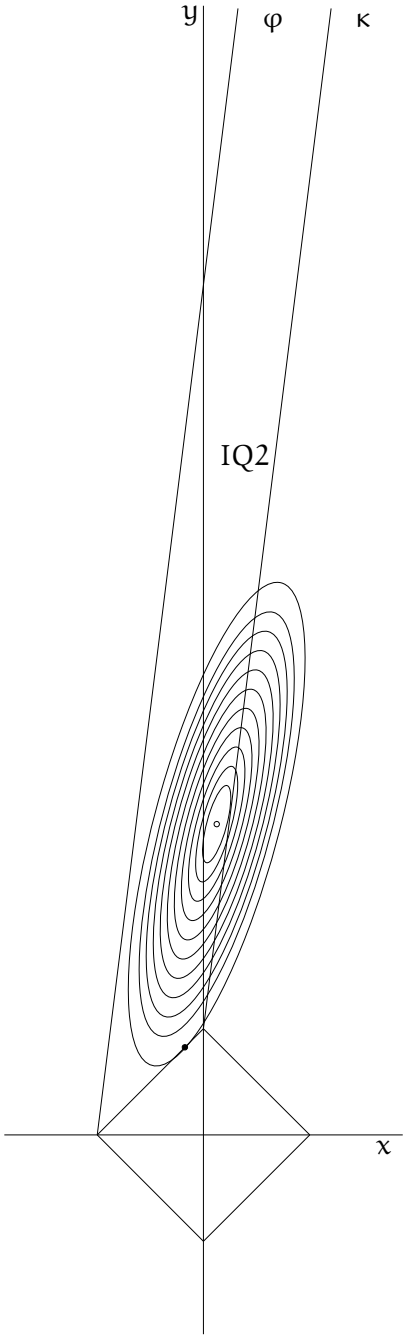


Figure 9: A Top Corner Solution

A Second Quadrant Interior Solution

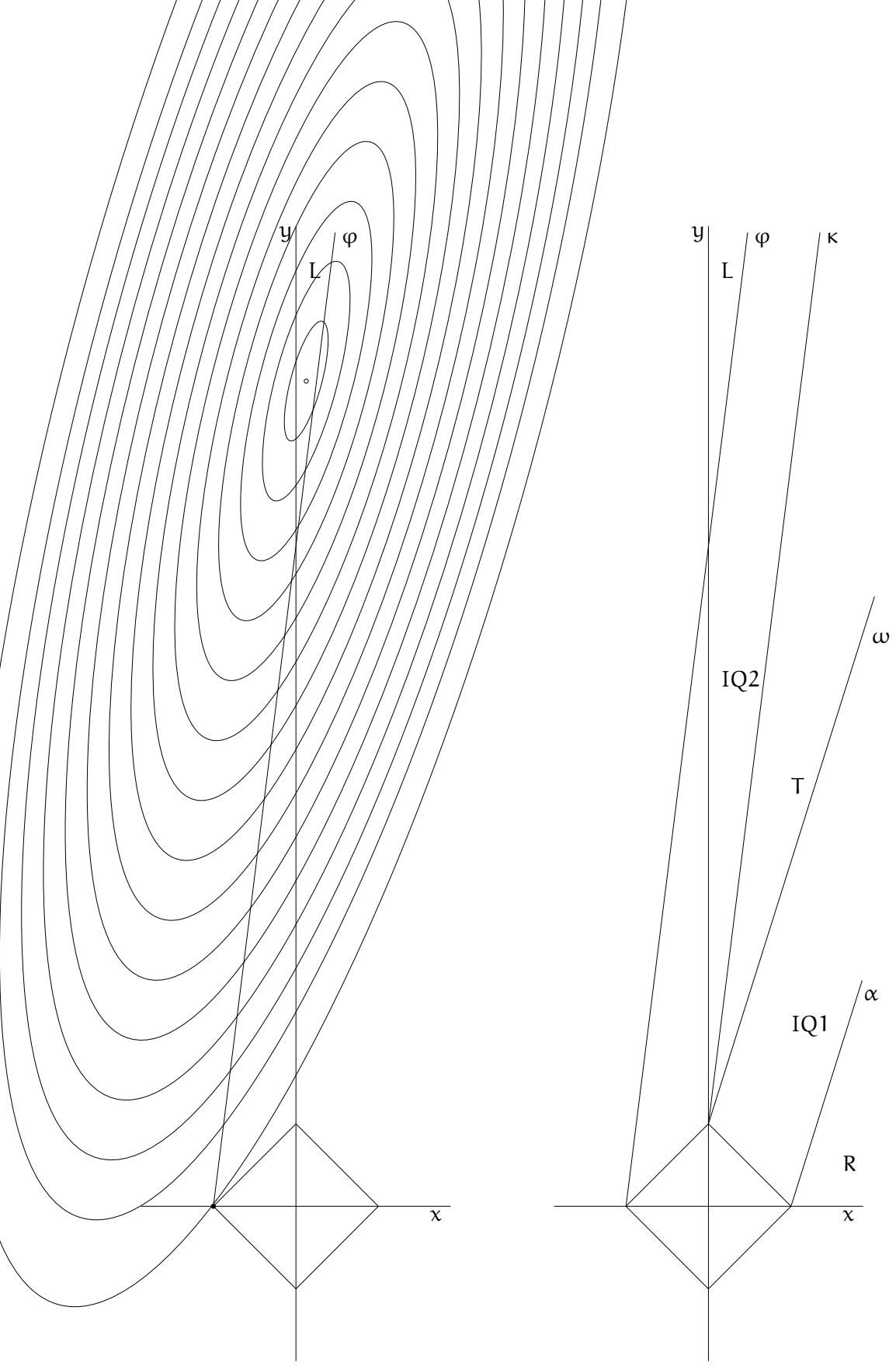
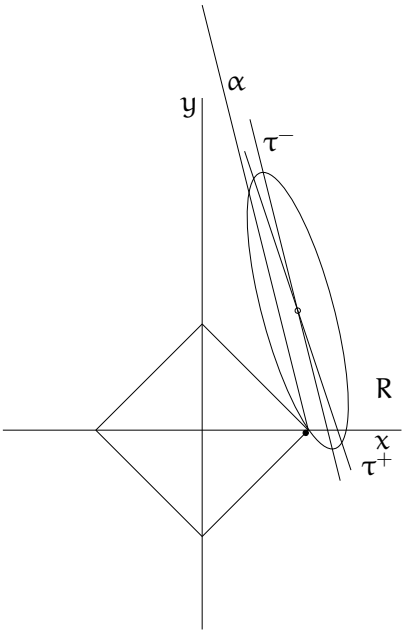


Figure 10: A Left Corner Solution Solution Types



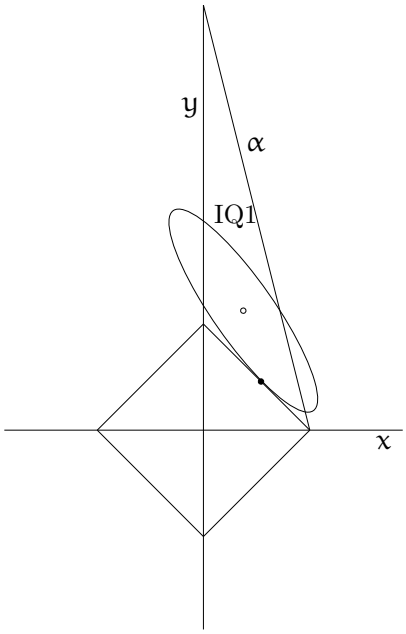


Figure 11: A Right Corner Solution

A First Quadrant Interior Solution

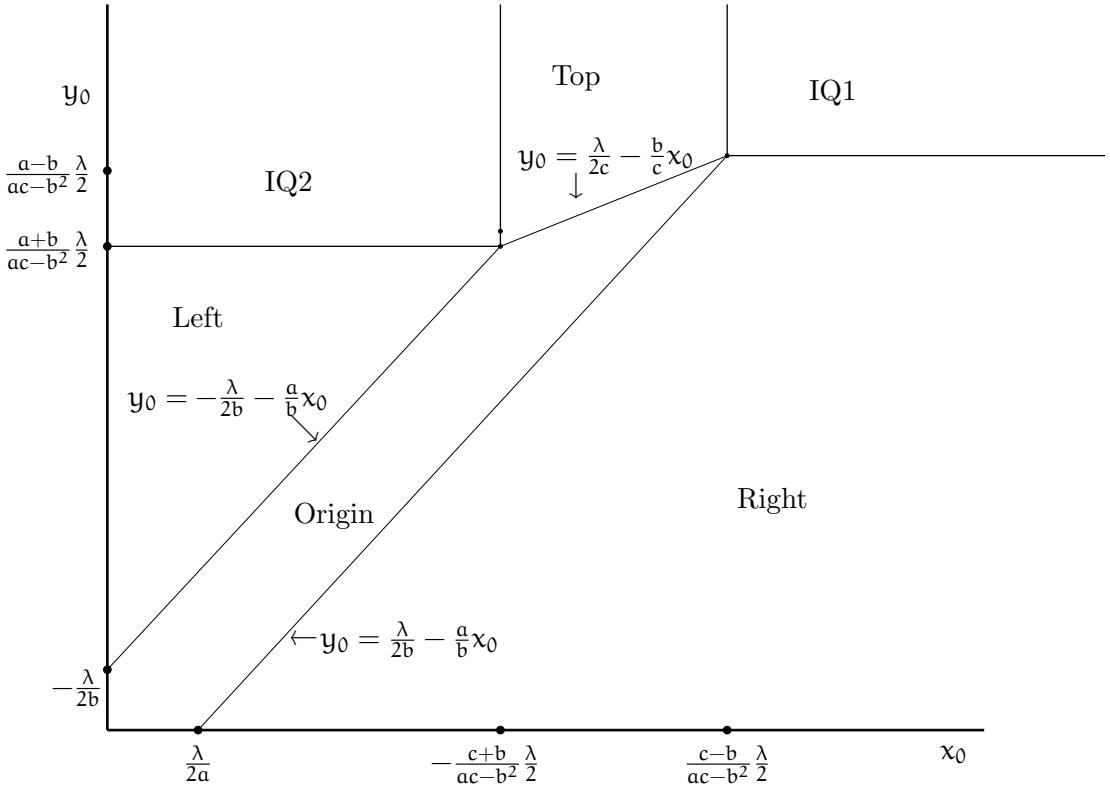


Figure 12: Solution types for (x_0, y_0) when $b < -c$

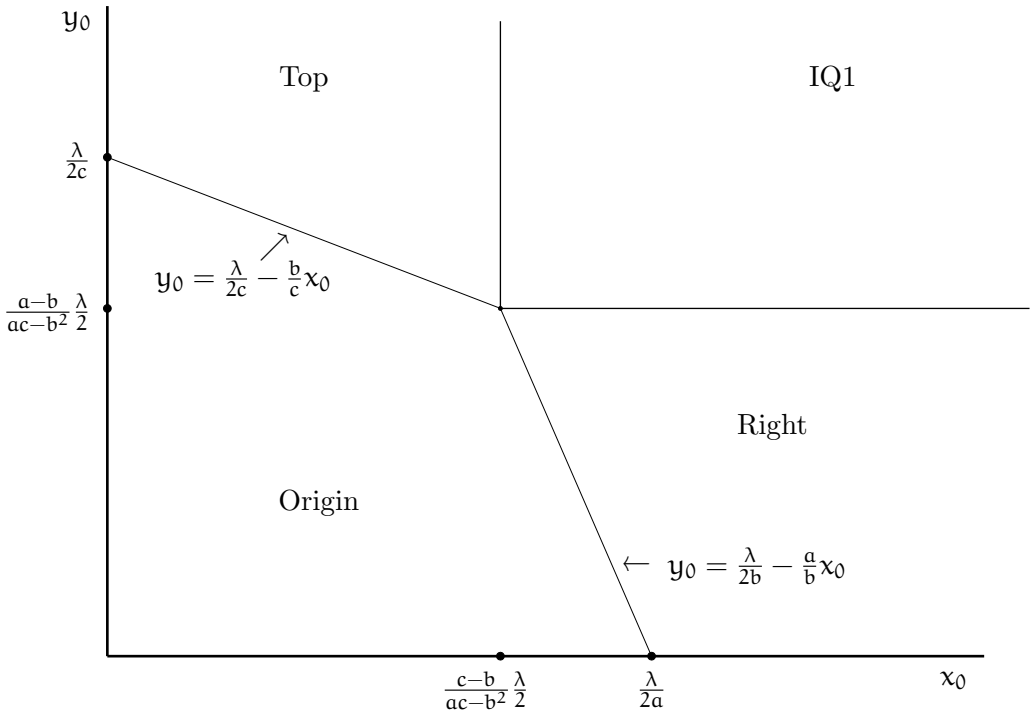


Figure 13: Solution types for (x_0, y_0) when $|b| < c$.

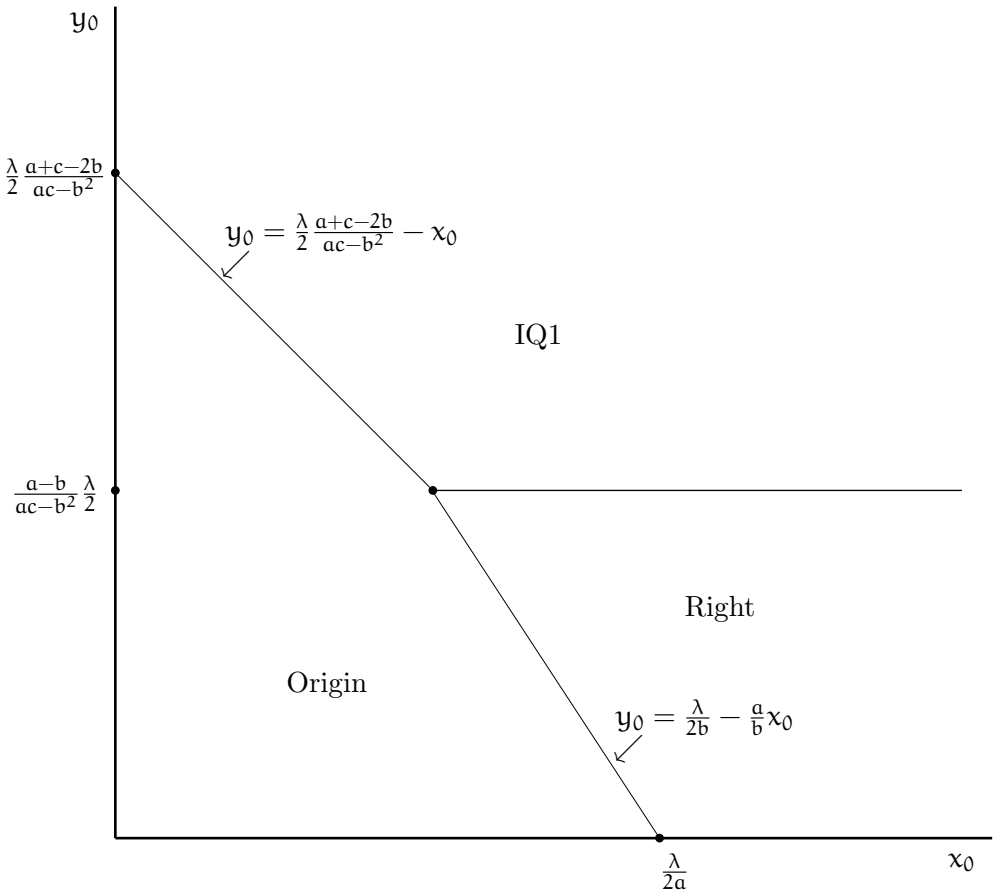


Figure 14: Solution types for (x_0, y_0) when $c < b$.