

The Constrained LASSO by the Critical Line Algorithm

Two ways to hand a regression to a portfolio optimiser

Thomas Schmelzer

2 October 2026

Abstract

Schmelzer and Hastie show that Markowitz's Critical Line Algorithm (CLA) and the constrained LASSO trace the same piecewise-linear curve. A companion note used that identity in one direction, computing efficient frontiers with a LASSO solver. This note goes the other way: it computes LASSO paths with an unmodified CLA that knows nothing about regression. There are two ways to pose a LASSO as a portfolio problem. In the *penalty form* every unit of gross position costs one unit of expected return, and the CLA's risk-tolerance parameter is the penalty λ itself. Bounds on the coefficients become position limits, and linear equality constraints become rows of the budget matrix. In the *budget form* the ℓ_1 ball, rescaled, is the long-only, fully invested simplex of Markowitz's original problem, with a riskless cash asset holding the unused budget. The parameter is the reciprocal of the ℓ_1 bound, homogeneous constraints such as $\mathbf{1}^\top \beta = 0$ carry over unchanged, and cash entering the portfolio marks the end of the LASSO path. On the diabetes data, a $p > n$ design and compositional data, the CLA reproduces every LARS knot to 10^{-13} and meets the LASSO's optimality conditions to 10^{-15} at every knot.

1 Introduction

Markowitz's Critical Line Algorithm [7] traces the efficient set of a quadratic program as a parameter moves. The solution is affine in the parameter while the active set is fixed and turns a corner when it changes. The homotopy and LARS algorithms [8, 2] do the same for the LASSO [12]. Schmelzer and Hastie [10] prove that the two paths are one curve: with $\Sigma = X^\top X$ and $\mu = X^\top y$, the gross-exposure-constrained mean-variance program and the constrained LASSO [4, 3] have the same solutions, under the same linear constraints. A companion note [11] put a stock LASSO solver to work on portfolio problems.

This note runs the identity the other way. It takes the CLA as Markowitz wrote it, for a quadratic objective over a polyhedron with one parameter in the linear term, and feeds it LASSO problems. Nothing in the algorithm is adapted. The work is in the encoding, and there are two encodings:

1. the *penalty form* (Section 3), whose parameter is λ and which takes coefficient bounds and general linear equality constraints;
2. the *budget form* (Section 4), whose parameter is the reciprocal of the ℓ_1 bound. It is Markowitz's long-only, fully invested problem with a cash asset, and it takes homogeneous constraints.

Neither reduction is deep, and the note claims no new theorem. What it adds is a working instance. Each form is a few lines on top of a CLA of about a hundred lines, and the numbers in Section 5 show that together they reproduce LARS and a conic solver to rounding error. Section 6 sets out where both forms stop.

Notation The design is $X \in \mathbb{R}^{n \times p}$, with n observations and p predictors, and the response is $y \in \mathbb{R}^n$. Write $G = X^\top X$ and $c = X^\top y$. The constrained LASSO is

$$L_\lambda : \min_{\beta} \frac{1}{2} \|y - X\beta\|_2^2 + \lambda \|\beta\|_1 \quad \text{s.t.} \quad A\beta = b, \quad l \leq \beta \leq u, \quad (1)$$

with $A \in \mathbb{R}^{m \times p}$ and $l_j \leq 0 \leq u_j$, infinite entries allowed. We assume, as [10] does, that L_λ has a unique minimiser $\beta(\lambda)$ for every $\lambda > 0$, which is the generic case.

2 The Critical Line Algorithm

The CLA solves the family of quadratic programs

$$P_\gamma : \min_x \frac{1}{2} x^\top \Sigma x - (q + \gamma\mu)^\top x \quad \text{s.t.} \quad Ax = b, \quad l \leq x \leq u, \quad (2)$$

for every γ from $+\infty$ down to 0, with $\Sigma \succeq 0$. In portfolio terms x is the vector of holdings, Σ the covariance, μ the expected returns and γ the risk tolerance. The term q is a benchmark. When $q = \Sigma\bar{x}$, the objective is $\frac{1}{2}(x - \bar{x})^\top \Sigma(x - \bar{x}) - \gamma\mu^\top x$ up to a constant, the tracking variance against a portfolio $\bar{x}$ traded off against expected return. This is the same CLA after the change of variables $x - \bar{x}$, whose only effect is to shift the bounds and the right-hand side. With $q = 0$ it is Markowitz's problem exactly.

Segments Write $g = \Sigma x - q - \gamma\mu + A^\top \nu$ for the gradient of the Lagrangian. A point x with multipliers ν is optimal when $g_j = 0$ on the *free* coordinates $\mathcal{F} = \{j : l_j < x_j < u_j\}$, $g_j \geq 0$ where $x_j = l_j$, and $g_j \leq 0$ where $x_j = u_j$. Call the coordinates at a bound *blocked*, $\mathcal{B}$. With $\mathcal{F}$ and $\mathcal{B}$ fixed, $x_{\mathcal{F}}$ and ν solve the bordered system

$$\begin{bmatrix} \Sigma_{\mathcal{F}\mathcal{F}} & A_{\mathcal{F}}^\top \\ A_{\mathcal{F}} & 0 \end{bmatrix} \begin{bmatrix} x_{\mathcal{F}} \\ \nu \end{bmatrix} = \begin{bmatrix} q_{\mathcal{F}} - \Sigma_{\mathcal{F}\mathcal{B}}x_{\mathcal{B}} \\ b - A_{\mathcal{B}}x_{\mathcal{B}} \end{bmatrix} + \gamma \begin{bmatrix} \mu_{\mathcal{F}} \\ 0 \end{bmatrix}, \quad (3)$$

so $x_{\mathcal{F}} = \alpha + \gamma\delta$ and $\nu = \nu_0 + \gamma\nu_1$ are affine in γ . The gradient on the blocked coordinates is affine too, $g_{\mathcal{B}} = g_0 + \gamma g_1$. One factorisation and two right-hand sides give the whole segment.

Events Lowering γ , the segment ends at the largest γ below the current one at which either a free coordinate reaches a bound, $\gamma = (l_i - \alpha_i)/\delta_i$ for $\delta_i > 0$ or $(u_i - \alpha_i)/\delta_i$ for $\delta_i < 0$, or a blocked coordinate's gradient changes sign, $\gamma = -g_{0j}/g_{1j}$ for $g_{1j} > 0$ at a lower bound or $g_{1j} < 0$ at an upper one. The coordinate concerned changes sides, (3) is solved again, and the walk continues. If there is no event the last segment runs to $\gamma = 0$. Every corner where an event happens is a *turning point*.

The start As $\gamma \rightarrow \infty$ the linear term dominates, and the path starts at a vertex maximising $\mu^\top x$ over the feasible set. A linear program finds it. If the maximiser is unique and has exactly m coordinates strictly inside their bounds, the bordered matrix in (3) is nonsingular and the walk can begin. This is the nondegenerate case. The implementation checks the count and refuses to start otherwise.

3 The penalty form

Split each coefficient into the signs its bounds allow. Let S have a column e_j for every j with $u_j > 0$ and a column $-e_j$ for every j with $l_j < 0$, so $\beta = Sz$ with $z \geq 0$, and let $\bar{u}$ collect the matching bounds u_j and $-l_j$. A coefficient whose bounds are both zero is fixed and can be dropped, so every j has at least one column and S has full row rank.

Proposition 1 (Penalty as cost). *For $\lambda > 0$, $\beta(\lambda) = Sz(\lambda)$, where $z(\lambda)$ solves the CLA instance P_γ with*

$$\Sigma = S^\top GS, \quad \mu = -\mathbf{1}, \quad q = S^\top c, \quad \text{rows } ASz = b, \quad 0 \leq z \leq \bar{u}, \quad \gamma = \lambda.$$

Moreover q lies in the range of Σ , so the benchmark reading of Section 2 applies: the program tracks a least-squares fit and pays λ for every unit of gross position.

Proof. Up to the constant $\frac{1}{2}\|y\|^2$, the CLA objective at $\gamma = \lambda$ is $\frac{1}{2}\|y - XSz\|^2 + \lambda \mathbf{1}^\top z$. For a fixed β feasible in (1), the smallest $\mathbf{1}^\top z$ over $z \geq 0$ with $Sz = \beta$ and $z \leq \bar{u}$ is $\|\beta\|_1$. It is attained by putting β_j on the column of its sign, which respects $\bar{u}$ because $l_j \leq \beta_j \leq u_j$. The minimum of the CLA program over z is therefore the minimum of L_λ over β . At a minimiser no coordinate holds both signs, since lowering both by the same amount keeps β and reduces the cost by 2λ times that amount. Hence $Sz(\lambda)$ minimises L_λ , and by uniqueness equals $\beta(\lambda)$. For the range claim, $c = X^\top y$ lies in the range of G , say $c = Gv$. Since S has full row rank, $v = S\bar{x}$ for some $\bar{x}$, and then $q = S^\top GS\bar{x} = \Sigma\bar{x}$, where $S\bar{x}$ is a least-squares fit. $\square$

The covariance $S^\top GS$ is singular whenever both signs of a coefficient are allowed, because the two columns cancel. The bordered system (3) is nonetheless nonsingular on every segment with $\lambda > 0$: the argument in the proof keeps the two halves of a coefficient out of $\mathcal{F}$ together, and under the general-position assumption of [10] the columns of X on the support are independent. The halves come apart only at $\lambda = 0$, where the gradient of a free half's twin is 2λ and vanishes. In floating point that zero can land at $\gamma \approx 10^{-16}$ instead, so the implementation treats event parameters below 10^{-10} times the first one as zero.

The start Without equality rows the starting vertex is $z = 0$, all coordinates blocked and none needed free. The first event is the first entry, at $\gamma = \max_j |c_j|$ over the allowed signs, which is LARS's $\lambda_{\max}$. With equality rows the start is the linear program $\min \mathbf{1}^\top z$ over $ASz = b$, $0 \leq z \leq \bar{u}$, the point of smallest ℓ_1 norm on the constraint set. For generic A and $b \neq 0$ it is a unique vertex with m nonzeros. The path then holds still on its first segment, $\delta = 0$, while the multipliers absorb the growing penalty, until a second coefficient enters.

4 The budget form

The penalty form keeps λ as the parameter and gives up Markowitz's budget row. The budget form keeps the row and gives up the penalty. Its starting point is the bound form of the LASSO, $\min \frac{1}{2}\|y - X\beta\|^2$ over $\|\beta\|_1 \leq t$, which by Lagrangian duality has the same path, indexed by $t = \|\beta(\lambda)\|_1$ [10].

Proposition 2 (Budget as simplex). *Suppose $b = 0$ and $l_j, u_j \in \{0, \pm\infty\}$, so that every constraint is homogeneous. Write $\beta = tSw$ with $w \geq 0$, and add a riskless asset $w_0 \geq 0$ with $\mathbf{1}^\top w + w_0 = 1$. Then for $t > 0$ the bound-form LASSO is the CLA instance*

$$\Sigma = \begin{bmatrix} S^\top GS & 0 \\ 0 & 0 \end{bmatrix}, \quad \mu = \begin{bmatrix} S^\top c \\ 0 \end{bmatrix}, \quad q = 0, \quad \gamma = 1/t,$$

with rows $\mathbf{1}^\top w + w_0 = 1$ and $ASw = 0$ and bounds $w, w_0 \geq 0$. This is Markowitz's long-only, fully invested problem with a cash asset of zero variance and zero return. If η is the multiplier of the budget row and ν those of $ASw = 0$, then $\lambda = t\eta$ and the LASSO's multipliers for $A\beta = 0$ are $t\nu$.

Proof. As in Proposition 1, $\|\beta\|_1 \leq t$ is $\mathbf{1}^\top z \leq t$ for the split $z = tw$, and w_0 is its slack. The constraint $ASz = 0$ is unchanged by the scale t , which is where homogeneity is needed. A

nonzero right-hand side would become $b/t = \gamma b$, a constraint that moves with the parameter. Substituting $z = tw$ gives the objective $t^2(\frac{1}{2}w^\top S^\top GSw - \gamma c^\top Sw)$ plus a constant, and dividing by $t^2 > 0$ gives P_γ . For the multipliers, multiply the CLA's stationarity row for a free coordinate, $(S^\top GSw)_j - \gamma(S^\top c)_j + \eta + (S^\top A^\top \nu)_j = 0$, by t . This gives $(S^\top (G\beta - c))_j + t\eta + (S^\top A^\top t\nu)_j = 0$, the LASSO's stationarity condition with penalty $t\eta$ and constraint multipliers $t\nu$. The cash row reads $\eta \geq 0$ while cash is held at zero. $\square$

The two ends of the CLA path have plain meanings. At $\gamma = \infty$, $t = 0$, the start is the linear program $\max c^\top Sw$ over the simplex. Without equality rows that is a single asset, the predictor most correlated with the response, entering with the sign of its correlation and with $\lambda = \lim \eta/\gamma = \max_j |c_j|$. Under $\mathbf{1}^\top \beta = 0$ it is one long and one short position of weight $\frac{1}{2}$ each, the most and the least correlated predictors, entering together. Both starts are generically unique and nondegenerate, which is what the penalty form lacks when $b = 0$ (Section 6). At the other end, cash leaves its bound when its gradient η reaches zero, so $\lambda = 0$ and the budget goes slack: the LASSO path is over. Past that point the CLA still has a path, the ray $w = \beta(0)/t$ diluted by cash, but the twin halves' gradients vanish identically along it. The implementation therefore ends the run when cash enters, and lets that event win a tie. The tie matters when $p > n$: there every correlation vanishes at $\lambda = 0$ together with η , and rounding alone would decide which event the walk takes. Figure 2 shows the diabetes path in these terms.

The budget form gives the same knots as the penalty form only where the map from λ to t is one-to-one, which holds where $\|\beta(\lambda)\|_1$ is strictly decreasing [10, Proposition 1]. That fails only if the support's sign vector lies in the span of the active constraint normals. Under $\mathbf{1}^\top \beta = 0$ that would require every coefficient on the support to have the same sign, which is impossible for a nonzero β that sums to zero. So the zero-sum constraint is safe.

5 Numerical evidence

Table 1 lists the test problems. The diabetes data [2] have $n = 442$ patients and $p = 10$ standardised baseline variables, and the response is centred. *Wide* is a Gaussian design with $n = 40$ and $p = 100$, with 8 nonzero coefficients and noise of standard deviation 0.5. *Box* caps every diabetes coefficient at $|\beta_j| \leq u = 396.1$, half the largest least-squares coefficient. At $\lambda = 0$ the variables **bmi** and **s5** sit at the cap, and **s1** does not. *Pinned* requires the fitted value for the first patient to equal the observed one, one nonhomogeneous equality row. Its starting vertex puts all weight on **bmi**, the largest entry of that patient's covariate vector, and holds still until $\lambda = 4053$. *Zero-sum* uses log-compositions as predictors, in the style of Lin et al. [6]: $n = 60$ samples of $p = 30$ parts, with the constraint $\mathbf{1}^\top \beta = 0$ that makes the fit invariant to the reference part. *Zero-sum, signs* adds nonnegativity on the even-numbered parts.

Every path is checked in three ways. Where LARS applies it is the reference, compared knot by knot in both λ and β . Elsewhere the reference is an interior-point solve (Clarabel through CVXPY, tolerances 10^{-12}) at the midpoint of every segment, where the path is affine in λ and so known exactly from its two ends. This comparison is only as sharp as the solver, which reaches about 10^{-8} relative. Third, at every knot and every midpoint we evaluate the optimality conditions of (1) directly, without reference to either form. With $r = G\beta - c + A^\top \kappa$ they are $r_j = -\lambda \text{sign } \beta_j$ on the interior of the support, $|r_j| \leq \lambda$ at zero, the one-sided versions at a cap, and primal feasibility. The largest violation, relative to $\|c\|_\infty$, is the last column of the table.

Three observations. First, both forms return the same knots wherever both apply, and LARS's knots to 10^{-13} , including on the wide design, where 53 knots carry the path to the interpolating fit. Second, the interior-point gaps are the solver's, not the path's. The KKT residuals of the CLA's own output are at the level of rounding throughout, including the cases with no LARS to compare against. Third, the comparison exposed a quirk of the reference. In positive mode, the last point that `lars_path` returns is labelled $\alpha = 0$, but every active

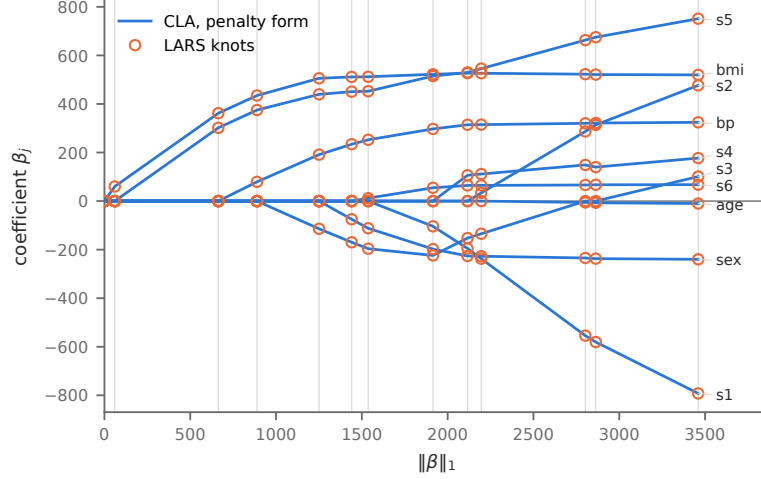


Figure 1: The diabetes LASSO path [2], computed by the CLA in the penalty form (lines). Circles: the knots `lars_path` returns [9]. All 13 knots agree to 2×10^{-13} , relative.

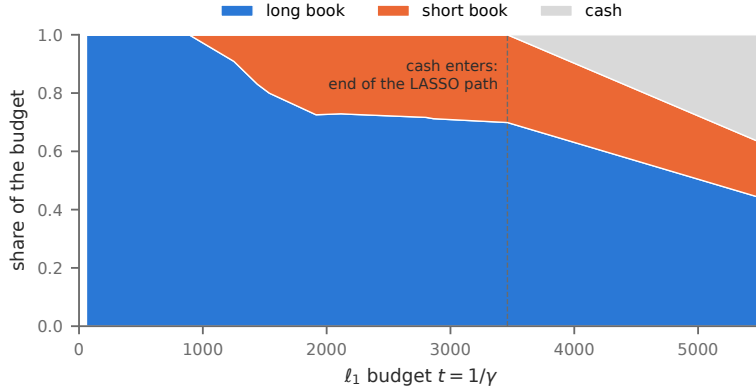


Figure 2: The same path seen by the budget form, as a portfolio. The long and short positions Sw and the cash weight w_0 share a budget of one, plotted against $t = 1/\gamma$. Cash enters at $t = \|\beta(0)\|_1 = 3460$, the end of the LASSO path. To its right the portfolio is the least-squares fit diluted by cash.

correlation there equals 30.65. It is the path at $\lambda = 30.65$, where the algorithm found nothing left to enter, rather than the nonnegative least-squares fit at $\lambda = 0$. Its KKT violation at $\lambda = 0$ is 3×10^{-2} . Both CLA forms carry on to the nonnegative least-squares solution, and the row in Table 1 compares that last knot with `scipy.optimize.nnls` [5]. All earlier knots are compared with LARS. Figure 3 shows the two constrained paths that have no LARS reference.

6 Where the two forms stop

Between them the forms cover coefficient bounds, nonhomogeneous equality rows with a nondegenerate start, and homogeneous equality rows with sign constraints. The gaps are all at the start or in the shape of the constraint.

Homogeneous rows in the penalty form With $b = 0$ the starting vertex is $z = 0$, with no free coordinates for $m > 0$ rows. This is a degenerate vertex: the multipliers at the start are not determined by the active set, and (3) is singular. The implementation detects this and refuses

problem	form	n	p	knots	reference	gap	KKT
plain	penalty	442	10	13	LARS	2.3×10^{-13}	2.3×10^{-16}
plain	budget	442	10	13	LARS	2.3×10^{-13}	4.0×10^{-16}
nonnegative	penalty	442	10	6	LARS	2.2×10^{-15}	2.4×10^{-16}
nonnegative	budget	442	10	6	LARS	1.9×10^{-15}	2.4×10^{-16}
wide	penalty	40	100	53	LARS	1.2×10^{-13}	4.2×10^{-16}
wide	budget	40	100	53	LARS	1.2×10^{-13}	7.3×10^{-16}
box	penalty	442	10	13	IPM	4.8×10^{-8}	1.8×10^{-16}
pinned	penalty	442	10	18	IPM	3.6×10^{-10}	1.4×10^{-15}
zero-sum	budget	60	30	30	IPM	5.8×10^{-8}	4.7×10^{-16}
zero-sum, signs	budget	60	30	19	IPM	1.3×10^{-10}	5.6×10^{-16}

Table 1: The CLA against LARS (knot by knot) and against an interior-point solver (IPM, at segment midpoints). *Gap*: largest discrepancy, relative to the largest coefficient. *KKT*: largest violation of the optimality conditions of (1) over all knots and midpoints, relative to $\|c\|_\infty$. Each problem goes through every form that accepts it.

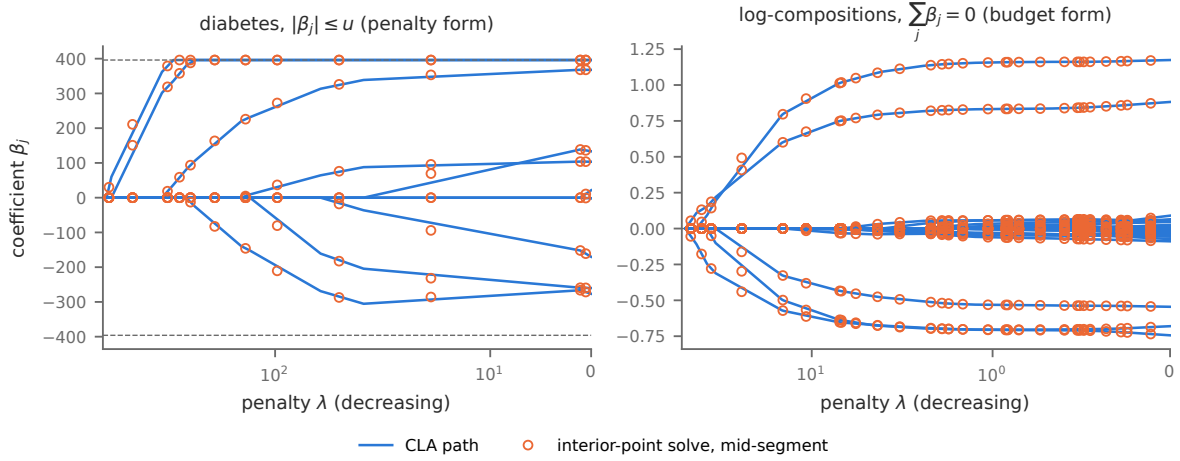


Figure 3: Two constrained paths with no LARS to check them against. Lines: the CLA. Circles: an interior-point solve at the midpoint of every segment. Left: the diabetes LASSO under $|\beta_j| \leq u$ (dashed), penalty form. Two coefficients reach the cap and stay there. Right: the compositional LASSO under $\mathbf{1}^\top \beta = 0$, budget form. The first two coefficients enter together, one positive and one negative.

to start. The budget form is the remedy, because rescaling to the simplex moves the start off the origin.

Flat starts A nonhomogeneous row can make the starting linear program non-unique. The fully invested constraint $\mathbf{1}^\top \beta = 1$ of sparse portfolio regression [1] is the standard case: every long-only β in the simplex has $\|\beta\|_1 = 1$ and minimises the ℓ_1 norm. The path’s first segment is then the least-squares fit over that face, constant in λ . This is the flat segment of [10, Proposition 2], and finding the point requires a quadratic program rather than a linear one. Markowitz’s own treatment of degenerate starts would handle it, but it is not implemented here.

Inequality rows Constraints $C\beta \leq h$ enter either form as slack assets with zero variance and zero return, one per row. With h in general position this changes nothing above. Homogeneous inequalities, such as the ordering constraints of a monotone fit, make the slack and the bound of the same coordinate active together at many vertices. This breaks linear independence,

and the walk needs the anti-cycling machinery of the active-set literature. Neither is in the implementation behind Table 1, so the note makes no numerical claim for them.

7 Conclusion

The identity of [10] says that the Critical Line Algorithm and the constrained LASSO trace one curve. This note shows the practical consequence: the CLA, in the form Markowitz wrote it, is a working constrained-LASSO path algorithm. The penalty form keeps λ and uses the bounds and rows. The budget form keeps the simplex and uses the scale invariance of homogeneous constraints. On the problems tested, the two forms agree with each other and with LARS to rounding error, and they meet the LASSO’s optimality conditions to 10^{-15} everywhere along the path.

Code and data availability

The note’s directory holds `make_figures.py`, a single Python script that declares its own pinned dependencies. It contains the CLA (the function `cla`), the two forms (`penalty_form`, `budget_form`) and every check. `uv run make_figures.py` regenerates the figures and Table 1, and `uv run make_figures.py test` runs the checks as tests. The diabetes data ship with scikit-learn, and the other problems are simulated from fixed seeds.

References

- [1] J. Brodie, I. Daubechies, C. De Mol, D. Giannone, and I. Loris. Sparse and stable Markowitz portfolios. *Proceedings of the National Academy of Sciences*, 106(30):12267–12272, 2009.
- [2] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *The Annals of Statistics*, 32(2):407–499, 2004.
- [3] B. R. Gaines, J. Kim, and H. Zhou. Algorithms for fitting the constrained lasso. *Journal of Computational and Graphical Statistics*, 27(4):861–871, 2018.
- [4] G. M. James, C. Paulson, and P. Rusmevichientong. Penalized and constrained optimization: an application to high-dimensional website advertising. *Journal of the American Statistical Association*, 115(529):107–122, 2020.
- [5] C. L. Lawson and R. J. Hanson. *Solving Least Squares Problems*. Prentice-Hall, 1974.
- [6] W. Lin, P. Shi, R. Feng, and H. Li. Variable selection in regression with compositional covariates. *Biometrika*, 101(4):785–797, 2014.
- [7] H. M. Markowitz. The optimization of a quadratic function subject to linear constraints. *Naval Research Logistics Quarterly*, 3(1–2):111–133, 1956.
- [8] M. R. Osborne, B. Presnell, and B. A. Turlach. A new approach to variable selection in least squares problems. *IMA Journal of Numerical Analysis*, 20(3):389–403, 2000.
- [9] F. Pedregosa et al. Scikit-learn: machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [10] T. Schmelzer and T. Hastie. The Critical Line Algorithm and the constrained LASSO: one curve, two literatures. arXiv preprint arXiv:2609.25704, 2026.
- [11] T. Schmelzer. The efficient frontier from a LASSO solver. arXiv preprint arXiv:2609.37108, 2026.
- [12] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58(1):267–288, 1996.