

The Efficient Frontier from a LASSO Solver

A practical guide to the CLA–LASSO identity

Thomas Schmelzer

26 September 2026

Abstract

In a recent paper, Schmelzer and Hastie argue that Markowitz’s Critical Line Algorithm and the LASSO path trace the same curve. Here we use that identity to compute efficient frontiers with a stock LASSO solver, `lars_path` from `scikit-learn`. It handles long–short portfolios under a leverage cap, fixed leverage with varying risk appetite, and the classical long-only, fully invested frontier. Called naively, the last path stops at the maximum–Sharpe portfolio. One shift of the response, by an amount computed in advance, lets a single call reach the minimum-variance portfolio.

1 Introduction

As the trade-off between expected return and variance moves, the optimal portfolio moves along straight lines and changes direction only at finitely many *corner portfolios*, where an asset enters or leaves. Markowitz’s Critical Line Algorithm (CLA) [13] walks from one corner to the next. The LASSO [19] behaves the same way as its penalty moves, and the LARS algorithm [3] walks its corners, which statisticians call knots. Schmelzer and Hastie [16] prove that after a change of variables the two paths coincide. This note is the practical side of that result, in three settings:

1. long–short under a gross-exposure cap, where the plain LASSO path is the frontier (Section 5);
2. fixed leverage with varying risk appetite, where the same path is rescaled (Section 6);
3. long only and fully invested, where the nonnegative LASSO path, shifted once, covers the whole frontier in one call (Section 7, Algorithm 1).

The usual alternative is to solve a quadratic program on a grid of tilts. Each solve is only as accurate as its tolerance, and every corner between two grid points is missed. A path algorithm returns the corners themselves, and every other frontier portfolio is a linear interpolation between two of them.

2 Background

Markowitz [12] called a portfolio efficient if no other has a higher expected return E without a higher variance V , or a lower V without a lower E . Traced by minimising $\frac{1}{2}V - \lambda E$ over fully invested portfolios for $\lambda \geq 0$, they form the efficient frontier, with the minimum-variance portfolio at $\lambda = 0$. The CLA followed in 1956 [13]. Tobin [20] and Sharpe [17] moved the picture to the (σ, E) plane, where Sharpe located the dominant portfolio, the one with the highest Sharpe ratio, at the tangent from the riskless rate.

The LASSO fits least squares under $\|\beta\|_1 \leq t$ and so shrinks and selects at once. Its path is piecewise linear in t . Osborne, Presnell and Turlach [14] gave a homotopy for it and Efron et

al. [3] gave LARS, which also covers the positive LASSO. Later practice moved to coordinate descent on a grid of penalties [5], which, like a grid of tilts, returns points rather than corners.

The two literatures developed for decades without citing each other [16, §5]. The best-known bridge, Brodie et al. [2], read a Markowitz problem with a gross-exposure penalty as a LASSO, but swept the penalty at a fixed target return rather than tracing the frontier. Fan, Zhang and Yu [4] ran LARS along the gross-exposure cap of the minimum-variance portfolio. They removed the budget by regressing on a reference asset, which leaves an ℓ_1 constraint that is no longer the LASSO's, so their path is an approximation.

3 The test problem

We use the seeded 20-asset example of Figure 1 in [16]: a five-factor covariance $\Sigma = \text{diag}(d) + U \text{diag}(\delta)U^\top$, and μ the sample mean of 50 days of returns simulated from it. Its units are arbitrary and its expected returns large against its risk, so Sharpe ratios reach 2.1 per period, far above anything real assets offer. It serves to illustrate; Section 8 adds real returns. The companion script `make_figures.py` regenerates every figure and number in the note, and its tests check each against a tolerance. The statement on [code and data availability](#) at the end says where to find it.

4 From mean–variance to least squares

Let $\Sigma \succ 0$ be the covariance of n assets and μ their expected returns. With the Cholesky factor $\Sigma = LL^\top$, set $X = L^\top$ and $y = L^{-1}\mu$, so that $X^\top X = \Sigma$ and $X^\top y = \mu$. Then for every portfolio w

$$\frac{1}{2} w^\top \Sigma w - \mu^\top w = \frac{1}{2} \|y - Xw\|_2^2 - \frac{1}{2} \|y\|_2^2, \quad (1)$$

and under any constraints the mean–variance problem is a least-squares problem.

```
X = np.linalg.cholesky(Sigma, upper=True) # X.T @ X == Sigma
y = np.linalg.lstsq(X.T, mu)[0]          # X.T @ y == mu
w = np.linalg.lstsq(X, y)[0]             # unconstrained optimum Sigma^{-1} mu
```

The last line solves the normal equations $X^\top Xw = X^\top y$, which are $\Sigma w = \mu$. With constraints, the optimality conditions become normal equations restricted to the assets held, a linear system in the trade-off parameter. The weights stay linear until the holdings change, and those changes are the corners. This is a change of variables, not a statistical model: X has one row per asset, and nobody measured y [16, §4].

Only $X^\top X$ and $X^\top y$ matter, so any square root of Σ will do. If Σ is a sample covariance from a $T \times n$ return matrix R with $T > n$, the centred returns $X = (R - \mathbf{1}\bar{r}^\top)/\sqrt{T-1}$ work, with y from `lstsq`. For a factor model, stacking $\text{diag}(d)^{1/2}$ over $\text{diag}(\delta)^{1/2}U^\top$ gives an $(n+k) \times n$ matrix with no factorisation at all, and $y = (\text{diag}(d)^{-1/2}\mu, 0)$. Both reproduce the Cholesky path to rounding.

5 Long–short with a leverage cap

Parameters. λ : the tilt, or risk appetite. c : the gross-exposure cap, and κ : its multiplier, the LASSO penalty of Sections 5 and 6. $t = \mathbf{1}^\top v = 1/\lambda$: the budget of the rescaled long-only problem, and ν : its multiplier, the LASSO penalty of Section 7.

With the gross exposure $\|w\|_1$ capped at c ,

$$\min_w \frac{1}{2} w^\top \Sigma w - \mu^\top w \quad \text{s.t.} \quad \|w\|_1 \leq c \quad (2)$$

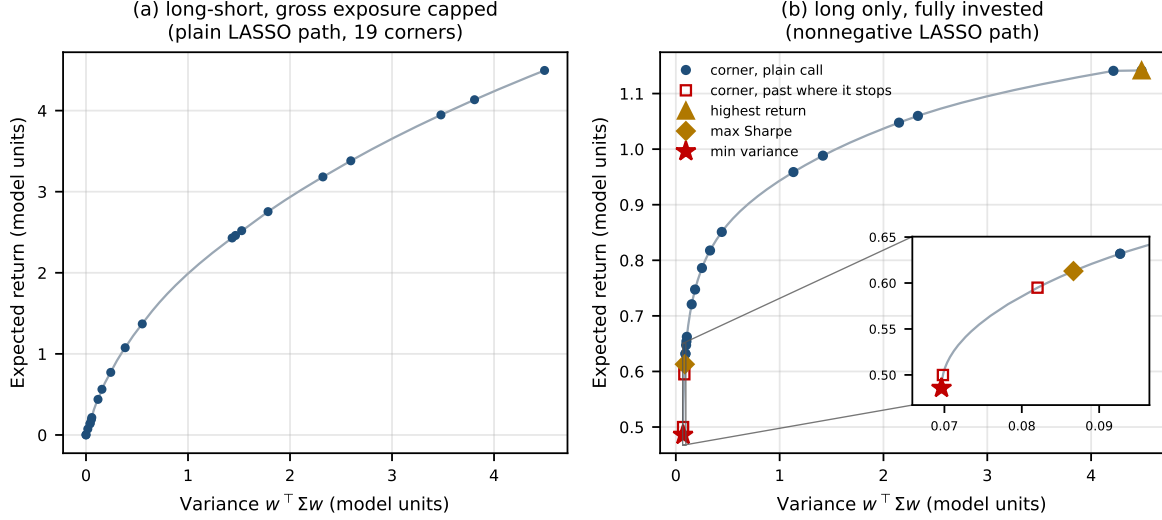


Figure 1: Both frontiers of the test problem, computed with a LASSO solver. (a) Long–short under a gross-exposure cap. (b) Long-only, fully invested: blue corners come from a plain call, red squares are the two it misses (Section 7). Marked are the highest-return (triangle), maximum-Sharpe (diamond) and minimum-variance (star) portfolios; the inset magnifies the low-risk end.

is, by (1), the LASSO in constrained form, the observation behind the sparse portfolios of Brodie et al. [2]. Its penalised form is $\min_{\beta} \frac{1}{2} \|y - X\beta\|_2^2 + \kappa \|\beta\|_1$, where $\kappa \geq 0$ is the multiplier of the cap and $c = \|\beta(\kappa)\|_1$. So the LASSO path is the frontier of (2) (Theorem 1 of [16]), and `scikit-learn` [15] computes it:

```
_, _, B = lars_path(X, y, method="lasso") # columns: start, corners, end
_, _, B = lars_path_gram(Xy=mu, Gram=Sigma, n_samples=len(mu), method="lasso")
```

The second line gives the same path from (Σ, μ) alone. In R the `lars` package [8] traces it too:

```
library(lars)
X <- chol(Sigma) # t(X) %*% X == Sigma
y <- backsolve(X, mu, transpose = TRUE) # t(X) %*% y == mu
fit <- lars(X, y, type = "lasso", normalize = FALSE, intercept = FALSE)
B <- t(coef(fit)) # columns: start, corners, end
```

It has no positivity constraint, so the long-only path of Section 7 needs `scikit-learn`. On the test problem the path has 21 columns: the empty portfolio at the start, 19 corners, each adding one asset, and the end. The path starts at $w = 0$, which is optimal while $\kappa \geq \max_j |\mu_j|$, and ends at $\kappa = 0$, where the cap stops binding at $w^* = \Sigma^{-1}\mu$ with $\|w^*\|_1 = 7.765$. That end maximises $\mu^\top w - \frac{1}{2} w^\top \Sigma w$, the second-order approximation to expected log growth, so it is the Kelly portfolio [10]. Since $\Sigma^{-1}\mu$ also maximises the scale-free Sharpe ratio, it points towards the maximum-Sharpe portfolio. On the test problem the Sharpe ratio rises from corner to corner, from 0.539 at the first to $\sqrt{\mu^\top \Sigma^{-1} \mu} = 2.120$ at the end; we have not shown that it must. Neither end is the minimum-variance portfolio, which needs the budget $\mathbf{1}^\top w = 1$ that (2) lacks.

6 Fixed leverage, varying risk appetite

A mandate usually fixes the leverage, say $c = 2$, and varies risk appetite through a tilt $\lambda > 0$. Substituting $w = \lambda v$ in $\min \frac{1}{2} w^\top \Sigma w - \lambda \mu^\top w$ subject to $\|w\|_1 \leq c$ gives (2) with cap c/λ . The

same path therefore serves: a column β with $t = \|\beta\|_1$, scaled to $w = (c/t)\beta$, is optimal at tilt $\lambda = c/t$.

```
t = np.abs(B[:, 1:]).sum(0)          # gross exposure of each column
W, tilts = B[:, 1:] * (c / t), c / t # the same portfolios at leverage c
```

At $c = 2$ the columns cover tilts from 745.6 down to 0.2576. Below that the cap is slack and the solution $\lambda\Sigma^{-1}\mu$ shrinks to zero, so nothing is missing.

7 Long only, fully invested

The classical frontier requires $w \geq 0$ and $\mathbf{1}^\top w = 1$. Substituting $w = \lambda v$ and using (1) turns the tilted problem into

$$\min_v \frac{1}{2} \|Xv - y\|_2^2 \quad \text{s.t.} \quad \mathbf{1}^\top v = t, \quad v \geq 0, \quad t = 1/\lambda, \quad (3)$$

with $w = v/t$ (Proposition 3 of [16]). For $v \geq 0$ the budget is $\|v\|_1$, so (3) is the nonnegative LASSO, whose penalty ν is the multiplier of the budget. `lars_path(..., positive=True)` traces it, and each column divided by its sum is a corner portfolio at tilt $1/\mathbf{1}^\top v$.

The stop. Called on (X, y) the path runs from $\nu = \max_j \mu_j$ down to $\nu = 0$ and stops there, because a LASSO solver takes no negative penalty. (If every $\mu_j \leq 0$ it returns only the origin.) On the test problem that gives 17 corners, down to tilt 0.1415. The stopping point $v_0 = \arg \min_{v \geq 0} \frac{1}{2} v^\top \Sigma v - \mu^\top v$ is the end of Section 5 with long-only weights. Writing $v = sw$ with $\mathbf{1}^\top w = 1$ and maximising over the scale $s \geq 0$ first leaves $\frac{1}{2}(\mu^\top w)^2 / w^\top \Sigma w$, half the squared Sharpe ratio, so $v_0 / \mathbf{1}^\top v_0$ is the long-only maximum-Sharpe portfolio, at tilt 0.1414. The frontier goes on, though: the budget still binds while ν turns negative (Figure 2), which a constraint allows and a penalty does not [16, §3.4]. The stop comes from the penalised form that LASSO solvers use, not from the problem.

The shift. Completing the square once more moves the penalty into the response:

$$\frac{1}{2} \|Xv - y\|_2^2 + \nu \mathbf{1}^\top v = \frac{1}{2} \|Xv - (y - \nu d)\|_2^2 + \text{const}, \quad X^\top d = \mathbf{1}. \quad (4)$$

If the nonnegative LASSO runs on $y - \nu_0 d$ for some $\nu_0 < 0$, its penalty $\nu' \geq 0$ acts as $\nu = \nu_0 + \nu'$ on the original problem, and one path covers every $\nu \geq \nu_0$. For the Cholesky factor $d = (X^\top)^{-1} \mathbf{1}$; for a tall X least squares returns the minimum-norm solution. `lars_path` reports ν' divided by the number of rows, and it stops after 500 steps unless told otherwise, which a large problem needs:

```
d = np.linalg.lstsq(X.T, np.ones(len(mu)))[0] # solves X.T @ d == 1
alphas, _, V = lars_path(X, y - nu0 * d, method="lasso", positive=True,
                        max_iter=10 * len(mu))
nu = nu0 + len(y) * alphas # budget multiplier per column
corners = V[:, 1:-1] # drop the origin and the end
W, tilts = corners / corners.sum(0), 1 / corners.sum(0)
```

Choosing ν_0 . Any ν_0 below the last corner works, and the last corner is known in advance. Beyond it the portfolio holds the minimum-variance support S , which is the support of $\arg \min_{u \geq 0} \|Xu - d\|_2$ [11]. There $v_S = a - \nu b$ with $a = \Sigma_{SS}^{-1} \mu_S$ and $b = \Sigma_{SS}^{-1} \mathbf{1}$, and the segment stays optimal while $\nu < a_i/b_i$ for $b_i > 0$ and, for $j \notin S$ with $\Sigma_{jS} b > 1$, while $\nu \leq (\mu_j - \Sigma_{jS} a) / (1 - \Sigma_{jS} b)$. The smallest bound is the last corner ν_{last} :

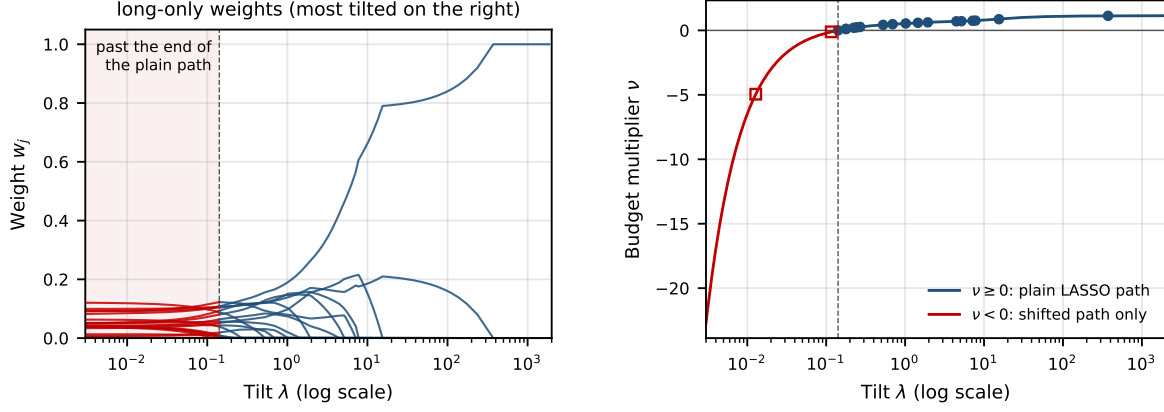


Figure 2: The long-only path of the test problem against the tilt λ . *Left*: the weights; blue where a plain call supplies them, red (shaded) where it does not. *Right*: the budget multiplier $\nu = (\mu - \Sigma v)_j$, j on the support. A plain call covers $\nu \geq 0$ and ends at $\nu = 0$, $\lambda = 0.1414$ (dashed); the shift (4) reaches the two further corners (squares) where $\nu < 0$.

```

S = nnls(X, d)[0] > 1e-9                                # minimum-variance support
a = np.linalg.solve(Sigma[np.ix_(S, S)], mu[S])
b = np.linalg.solve(Sigma[np.ix_(S, S)], np.ones(S.sum()))
h = 1 - Sigma[np.ix_(~S, S)] @ b
g = mu[~S] - Sigma[np.ix_(~S, S)] @ a
nu_last = min((a[b > 0] / b[b > 0]).min(),
              (g[h < 0] / h[h < 0]).min(initial=np.inf))
nu0 = nu_last - max(abs(nu_last), np.abs(mu).max())

```

No fixed value would be safe. ν_{last} is -4.95 on the test problem, -11.9 on one of the models of Table 1, and -0.178 on the stock data, where μ is small. The margin δ in $\nu_0 = \nu_{\text{last}} - \delta \max(|\nu_{\text{last}}|, \|\mu\|_\infty)$ has to be large enough that the end of the path does not merge with the last corner, and small enough not to cost accuracy, which falls roughly in proportion to $|\nu_0|$. On the stock data $\delta = 10^{-4}$ loses the last corner; from $\delta = 10^{-2}$ to 1 all 94 are found with KKT residuals below 4×10^{-13} , and at $\delta = 100$ the residual has grown to 2×10^{-11} . The code takes $\delta = 1$. The minimum-variance portfolio itself is the limit $\nu \rightarrow -\infty$; on the final segment it is $w_S \propto \Sigma_{SS}^{-1} \mathbf{1}$.

What ν means. By (4) the point with multiplier ν solves the same problem with μ replaced by $\mu - \nu \mathbf{1}$, so it is the long-only maximum-Sharpe portfolio at risk-free rate $r = \nu$, Black's zero-beta rate [1]. That is classical. The LASSO view adds that the rate is the path's own parameter, so for $\nu_0 < r < \max_j \mu_j$ the maximum-Sharpe portfolio is an interpolation between the two columns that bracket r :

```

assert nu[-1] < r < nu[0]                                # the path covers the rate r
k = np.argmax(nu < r)                                     # first column past r
s = (nu[k - 1] - r) / (nu[k - 1] - nu[k])
w_r = V[:, k - 1] + s * (V[:, k] - V[:, k - 1])
w_r /= w_r.sum()                                          # max-Sharpe portfolio at r

```

Weights versus frontier. Between corners the weights are linear in the tilt λ (or, in Section 5, in the cap c) and the variance is quadratic, so each piece of the frontier is an arc. Traced by the tilt λ at fixed constraints (Sections 6 and 7) it has no kinks either, since its slope $dE/dV = 1/(2\lambda)$ varies continuously. The corners then show only in the weights (Figure 2). The long-short path of Section 5 is traced by the cap instead and does kink (Figure 1(a)).

Algorithm 1: the long-only frontier from `lars_path`.

1. Build X, y with $X^\top X = \Sigma$, $X^\top y = \mu$, or use `lars_path_gram` with $Xy = \mu - \nu_0 \mathbf{1}$.
2. Solve $X^\top d = \mathbf{1}$, compute ν_{last} and set ν_0 as above.
3. Run the positive LASSO on $y - \nu_0 d$ with `max_iter` well above n , and recover $\nu = \nu_0 + n_{\text{rows}} \alpha$.
4. Drop the first and last columns; each remaining $v/\mathbf{1}^\top v$ is a corner, at tilt $1/\mathbf{1}^\top v$.
5. Take the minimum-variance end as $w_S \propto \Sigma_S^{-1} \mathbf{1}$, and any other point by interpolation, in $1/\lambda$ for a tilt or in ν for a rate.

8 Numerical evidence

Two kinds of evidence are reported. The first needs no second solver: each portfolio the path returns is checked against its own optimality conditions. For the cap with multiplier κ these are $(\mu - \Sigma w)_j = \kappa \text{sign}(w_j)$ on the support and $|(\mu - \Sigma w)_j| \leq \kappa$ off it; for the long-only problem at tilt λ they are $(\Sigma w - \lambda \mu)_j = \gamma$ on the support and $\geq \gamma$ off it, with $\mathbf{1}^\top w = 1$ and $w \geq 0$. The KKT residual is the worst violation, divided by $\|\mu\|_\infty$ or by the size of the gradient. The second compares with baselines that are independent of the path. The long-short problems go to Clarabel, an interior-point solver, through `cvxpy`; the long-only problem goes to `quadprog`, the active-set method of Goldfarb and Idnani [7].

Long-short. Every column has a KKT residual below 2.6×10^{-15} , and below 2.9×10^{-15} once rescaled to fixed leverage. At each column's cap Clarabel matches the path to 3.0×10^{-7} at the corners and 1.4×10^{-11} inside segments. The gap at the corners is the baseline's, since interior-point methods are least accurate where the holdings change. At fixed leverage the figures are 6.2×10^{-6} and 1.9×10^{-11} .

Long only. The shifted path finds 19 corners on the test problem, the 17 of a plain call and two more at tilts 0.1158 and 0.0128. They match the QP to 6.0×10^{-13} , points inside segments to 7.9×10^{-14} , the minimum-variance end to 1.0×10^{-16} , and the maximum-Sharpe portfolio at $r = 0$ to 3.4×10^{-15} . The corners and the minimum-variance end have KKT residuals below 3.6×10^{-14} . Table 1 repeats both checks on other sizes and seeds. Where the two disagree, at $n = 500$, the residual shows that the gap of 6.7×10^{-11} is the QP's. The number of corners need not equal the number of assets, because some assets leave and re-enter.

Real returns. On daily returns of 494 S&P 500 stocks over 1213 days (July 2021 to May 2026, from Yahoo Finance; the script that fetches them is published with the code), with the centred returns as X , the last corner sits at $\nu_{\text{last}} = -0.178$, about fifty times $\max_j |\mu_j|$ below zero. Algorithm 1 returns all 94 corners in 0.14 seconds, while one QP takes about 0.12 seconds, so a 100-point grid costs about 12 seconds and still misses the corners. Warm-starting helps less than one might hope: OSQP [18], set up once so that its factorisation is reused, with each tilt changing only the linear term and each solve starting from the last, takes 4.9 seconds for the same grid at tolerance 10^{-6} and agrees with the cold QP to 4.1×10^{-13} at every grid point. At 10^{-5} it takes 3.4 seconds but misses some points by 10^{-3} . Every corner has a KKT residual below 3.7×10^{-13} and matches its QP to 5.4×10^{-13} , and the minimum-variance portfolio, which holds 53 stocks, matches to 1.2×10^{-15} . The timings come from a laptop.

n	seeds	corners	ν_{last}	KKT residual	worst corner	min.-variance end	time (ms)
20	3	19	-0.57 to 0.12	1.1×10^{-14}	8.6×10^{-14}	2.4×10^{-16}	0.8
50	3	49	-11.9 to -0.74	1.0×10^{-13}	1.5×10^{-13}	1.7×10^{-16}	2.5
100	3	99–103	-0.56 to -0.38	2.5×10^{-14}	1.1×10^{-11}	3.3×10^{-16}	7.7
500	3	503	-0.45 to -0.26	2.1×10^{-13}	6.7×10^{-11}	4.3×10^{-16}	239

Table 1: Algorithm 1 on five-factor models, three seeds each: the worst KKT residual over the corners and the minimum-variance end, the largest ℓ_∞ weight discrepancy against an independent active-set QP at the same tilt, and the median time of one run of Algorithm 1, ν_{last} included, on a laptop.

9 When to use it

The path beats a grid of QP solves, cold- or warm-started, on both speed and accuracy. LARS is in R and `scikit-learn`, the positive LASSO only in `scikit-learn`, so corner portfolios need no portfolio software, and one call shows that the knots of a LASSO path and the turning points of a frontier are the same points. It is the wrong tool when Σ is singular, where the identity is not known to survive [16, §7], or when the portfolio needs more than one linear constraint. The LASSO that `scikit-learn` solves has exactly one besides the signs, $\|\beta\|_1 \leq t$: the gross-exposure cap of Sections 5 and 6, or, with `positive=True`, the budget $\mathbf{1}^\top w = 1$ of Section 7. The signs are either all free or all nonnegative. A dollar-neutral book adds the equality $\mathbf{1}^\top \beta = 0$, and a 130/30 book needs the budget and the gross cap at once. Path algorithms for a LASSO with general linear constraints exist [6, 9], but not in `scikit-learn`. A risk budget $w^\top \Sigma w \leq \sigma^2$ has no path of its own, but along each segment the variance is quadratic in the parameter, so the portfolio at a given σ comes from one scalar equation.

The numbers above test the implementation. Theorem 1 itself is proved in [16], and this note adds what a practitioner needs to use it: the shift, a ν_0 computed in advance, and a `max_iter` large enough for big problems.

Code and data availability

The software is at <https://github.com/Jebel-Quant/lasso>, in the directory `preprints/lasso/`, under the MIT license. `make_figures.py` regenerates every figure and number in the note; `uv run make_figures.py test` runs its checks. It is a single Python script that declares its own pinned dependencies. The R snippet of Section 5 is `R/long_short.R`, with a check against `lars_path`.

The test problem and the factor models of Table 1 are simulated inside the script from fixed seeds, so they need no data. The stock returns of Section 8 come from Yahoo Finance and are not redistributed. `fetch_sp500.py` downloads them: the S&P 500 constituents listed on Wikipedia, adjusted closing prices from 2021-06-01 to 2026-06-01, keeping names with no more than 5% of days missing. The note used the snapshot with SHA-256 `b5faa522...9937e`. A later download differs slightly, because the constituents change and Yahoo revises adjusted prices; the checks pass on it all the same.

Acknowledgements

I thank Trevor Hastie for many helpful discussions.

References

- [1] F. Black. Capital market equilibrium with restricted borrowing. *The Journal of Business*, 45(3):444–455, 1972.
- [2] 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.
- [3] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *The Annals of Statistics*, 32(2):407–499, 2004.
- [4] J. Fan, J. Zhang, and K. Yu. Vast portfolio selection with gross-exposure constraints. *Journal of the American Statistical Association*, 107(498):592–606, 2012.
- [5] J. Friedman, T. Hastie, and R. Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22, 2010.
- [6] 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.
- [7] D. Goldfarb and A. Idnani. A numerically stable dual method for solving strictly convex quadratic programs. *Mathematical Programming*, 27(1):1–33, 1983.
- [8] T. Hastie and B. Efron. **lars**: least angle regression, lasso and forward stagewise. R package version 1.3, 2022.
- [9] 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.
- [10] J. L. Kelly, Jr. A new interpretation of information rate. *Bell System Technical Journal*, 35(4):917–926, 1956.
- [11] C. L. Lawson and R. J. Hanson. *Solving Least Squares Problems*. Prentice-Hall, 1974.
- [12] H. M. Markowitz. Portfolio selection. *The Journal of Finance*, 7(1):77–91, 1952.
- [13] H. M. Markowitz. The optimization of a quadratic function subject to linear constraints. *Naval Research Logistics Quarterly*, 3(1–2):111–133, 1956.
- [14] 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.
- [15] F. Pedregosa et al. Scikit-learn: machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [16] T. Schmelzer and T. Hastie. The Critical Line Algorithm and the constrained LASSO: one curve, two literatures. arXiv preprint arXiv:2609.25704, 2026.
- [17] W. F. Sharpe. Capital asset prices: a theory of market equilibrium under conditions of risk. *The Journal of Finance*, 19(3):425–442, 1964.
- [18] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd. OSQP: an operator splitting solver for quadratic programs. *Mathematical Programming Computation*, 12(4):637–672, 2020.
- [19] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58(1):267–288, 1996.
- [20] J. Tobin. Liquidity preference as behavior towards risk. *The Review of Economic Studies*, 25(2):65–86, 1958.