Portfolio Allocation
From QP to ML Optimization Algorithms

Thierry Roncalli*

*Amundi Asset Management[1], France

This version: February 25th, 2019

[1] The opinions expressed in this presentation are those of the authors and are not meant to represent the opinions or official positions of Amundi Asset Management. This presentation is based on joint works with Edmond Lezmi, Thibault Bourgeron, Joan Gonzalvez, Jean-Charles Richard and Jiali Xu.

The Markowitz optimization problem

- $x = (x_1, \ldots, x_n)$ is the vector of weights in the portfolio
- $\mu = \mathbb{E}[R]$ and $\Sigma = \mathbb{E}\left[(R-\mu)(R-\mu)^\top\right]$ are the vector of expected returns and the covariance matrix of asset returns
- We note $\mu(x) = x^\top \mu$ the expected return of the portfolio and $\sigma(x) = \sqrt{x^\top \Sigma x}$ the portfolio volatility

Asset allocation problems (Markowitz, 1952)

1. σ-problem:
$$\max \mu(x) \quad \text{s.t.} \quad \sigma(x) \leq \sigma^\star$$
2. μ-problem:
$$\min \sigma(x) \quad \text{s.t.} \quad \mu(x) \geq \mu^\star$$

The Markowitz solution problem

QP trick (Markowitz, 1952 and 1956)

Transform the previous problems into a QP problem:

$$\begin{aligned} x^{\star}(\gamma) &= \arg\min \frac{1}{2} x^{\top} \Sigma x - \gamma x^{\top} \mu \\ &\text{s.t.} \quad \mathbf{1}_n^{\top} x = 1 \end{aligned}$$

Solving σ- and μ-problems are equivalent to QP + bisection algorithm

- Mann, H.B. (1943), Quadratic Forms with Linear Constraints, *American Mathematical Monthly*, 50, pp. 430-433.
- Martin, A.D. (1955), Mathematical Programming of Portfolio Selections, *Management Science*, 1(2), pp. 152-166.
- Frank, M., and Wolfe, P. (1956), An Algorithm for Quadratic Programming, *Naval Research Logistics Quarterly*, 3, pp. 95-110.
- Hildreth, C. (1957), A Quadratic Programming Procedure, *Naval Research Logistics Quarterly*, 4, pp. 79-85.
- Barankin, E.W., and Dorfman, R. (1958), On Quadratic Programming, *University of California Publications in Statistics*, 2(13), pp. 285-318.
- Beale, E.M.L. (1959), On Quadratic Programming, *Naval Research Logistics Quarterly*, 6(3), pp. 227-243.
- Wolfe, P. (1959), The Simplex method for Quadratic Programming, *Econometrica*, 27, pp. 382-398.

Primal QP problem

Definition

A quadratic programming (QP) problem is an optimization problem with a quadratic objective function and linear inequality constraints:

$$
\begin{aligned}
x^\star &= \arg\min \frac{1}{2} x^\top Q x - x^\top R \\
\text{s.t.} &\quad Sx \leq T
\end{aligned}
$$

where x is a $n \times 1$ vector, Q is a $n \times n$ matrix and R is a $n \times 1$ vector

We have

$$
Sx \leq T \Leftrightarrow \left\{ \begin{array}{l} Ax = B \\ Cx \leq D \\ x^{\min} \leq x \leq x^{\max} \end{array} \right.
$$

because:

$$
Ax = B \Leftrightarrow \left\{ \begin{array}{l} Ax \geq B \\ Ax \leq B \end{array} \right.
$$

Constrained ordinary least squares

$$\hat{\beta}^{\mathrm{ols}} = \arg\min \frac{1}{2} \mathrm{RSS}(\beta)$$

where:

$$\begin{aligned} \mathrm{RSS}(\beta) &= (Y - X\beta)^\top (Y - X\beta) \\ &= Y^\top Y + \beta^\top \left(X^\top X\right)\beta - 2\beta^\top \left(X^\top Y\right) \end{aligned}$$

We deduce that:

$$\begin{aligned} \hat{\beta}^{\mathrm{ols}} &= \arg\min \frac{1}{2}\beta^\top Q \beta - \beta^\top R \\ & \text{s.t.} \quad \begin{cases} A\beta = B \\ C\beta \leq D \\ \beta^{\min} \leq \beta \leq \beta^{\max} \end{cases} \end{aligned}$$

where $Q = X^\top X$ and $R = X^\top Y$

Relationship between linear regression and Markowitz optimization

- Linear regression:

$$Y = X\beta + \varepsilon$$

The solution is equal to:

$$\hat{\beta}^{\mathrm{ols}} = \left(X^\top X\right)^{-1} X^\top Y$$

- Markowitz optimization with empirical covariance matrix $\hat{\Sigma}$ and empirical expected returns $\hat{\mu}$:

$$\gamma \mathbf{1}_n = Rx + \varepsilon$$

where R is the matrix of (centered) asset returns (number of observations $\times$ number of assets) .The solution is equal to:

$$\begin{aligned}\hat{x}^{\mathrm{mvo}} &= \left(R^\top R\right)^{-1} R^\top \gamma \mathbf{1}_n \\ &= \gamma \hat{\Sigma}^{-1} \hat{\mu}\end{aligned}$$

Portfolio optimization with a benchmark

Let $\mu(x \mid b) = (x-b)^\top \mu$ be the expected excess return and $\sigma(x \mid b) = \sqrt{(x-b)^\top \Sigma (x-b)}$ be the tracking error volatility, where b is the benchmark

The objective function is:

$$\begin{aligned} f(x \mid b) &= \frac{1}{2}(x-b)^\top \Sigma (x-b) - \gamma (x-b)^\top \mu \\ &\propto \frac{1}{2} x^\top \Sigma x - \gamma x^\top \left(\mu + \frac{1}{\gamma} \Sigma b \right) \end{aligned}$$

$\Rightarrow$ QP problem with $Q = \Sigma$ and $R = \gamma \tilde{\mu}$ where $\tilde{\mu} = \mu + \frac{1}{\gamma}\Sigma b$ is the regularized vector of expected returns

- Tracking error constraints $\Leftrightarrow$ regularization of the QP problem
- If b is the risk-free asset, the regularized QP solution is the capital market line (Roncalli, 2013)

Index sampling

The portfolio sampling problem

We have:

$$x^\star = \arg\min \frac{1}{2}(x-b)^\top \Sigma (x-b) \quad \text{u.c.} \quad \begin{cases} \mathbf{1}_n^\top x = 1 \\ x \geq \mathbf{0}_n \\ \sum_{i=1}^n \mathbb{1}\{x_i > 0\} \leq n_x \end{cases}$$

where b is the vector of index weights

Index sampling

Heuristic algorithm

1. We set $x_{(0)}^{\max} = \mathbf{1}_n$. At the iteration k, we solve the QP problem by taking into account the upper bounds $x_{(k)}^{\max}$:

$$
\begin{aligned}
x_{(k)}^{\star} &= \arg\min \frac{1}{2}\left(x_{(k)} - b\right)^{\top} \Sigma \left(x_{(k)} - b\right) \\
\text{s.t.} &\quad \mathbf{1}_n^{\top} x_{(k)} = 1,\ \mathbf{0}_n \leq x_{(k)} \leq x_{(k)}^{\max}
\end{aligned}
$$

2. We then update the upper bounds $x_{(k)}^{\max}$ by deleting the stock with the lowest non-zero optimized weight
3. We iterate the two steps until $\sum_{i=1}^{n} \mathbb{1}\left\{x_{(k),i}^{*} > 0\right\} \leq n_x$

The heuristic algorithm is the fastest method (vs backward elimination, forward selection, MIQP, etc.)

Dual QP problem

The Lagrange function is equal to:

$$\mathcal{L}(x;\lambda) = \frac{1}{2}x^\top Qx - x^\top R + \lambda^\top (Sx - T)$$

We deduce that the dual problem problem is defined by:

$$\begin{aligned} \lambda^\star &= \arg\max \left\{ \inf_x \mathcal{L}(x;\lambda) \right\} \\ \text{s.t.} &\quad \lambda \geq 0 \end{aligned}$$

Duality theorem

We can show that the dual program is another quadratic program:

$$\begin{aligned} \lambda^\star &= \arg\min \frac{1}{2}\lambda^\top \bar{Q}\lambda - \lambda^\top \bar{R} \\ \text{s.t.} &\quad \lambda \geq 0 \end{aligned}$$

with $\bar{Q} = SQ^{-1}S^\top$ and $\bar{R} = SQ^{-1}R - T$

Support vector machines

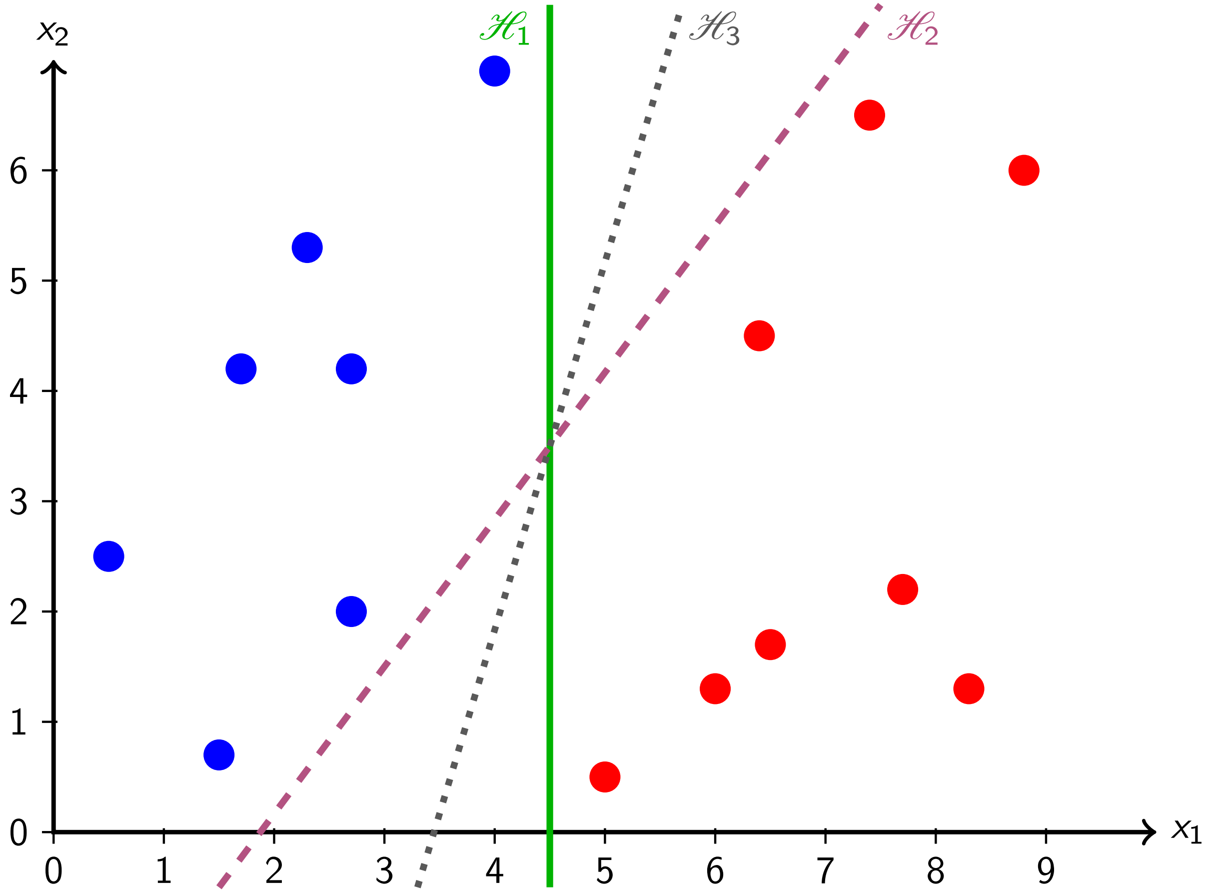


Figure: Separating hyperplane picking

Source: Roncalli (2019).

Support vector machines

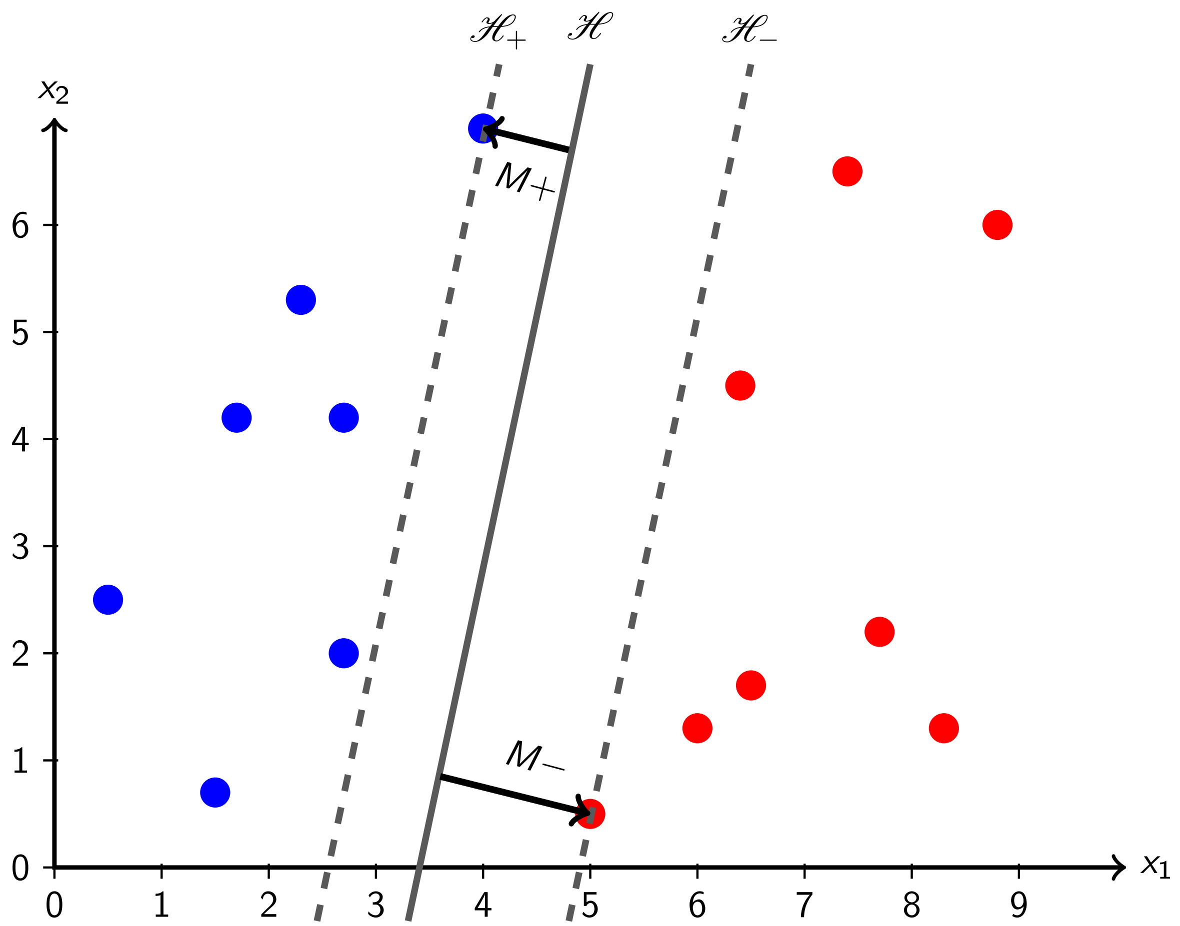


Figure: Margins of separation

Source: Roncalli (2019).

Support vector machines

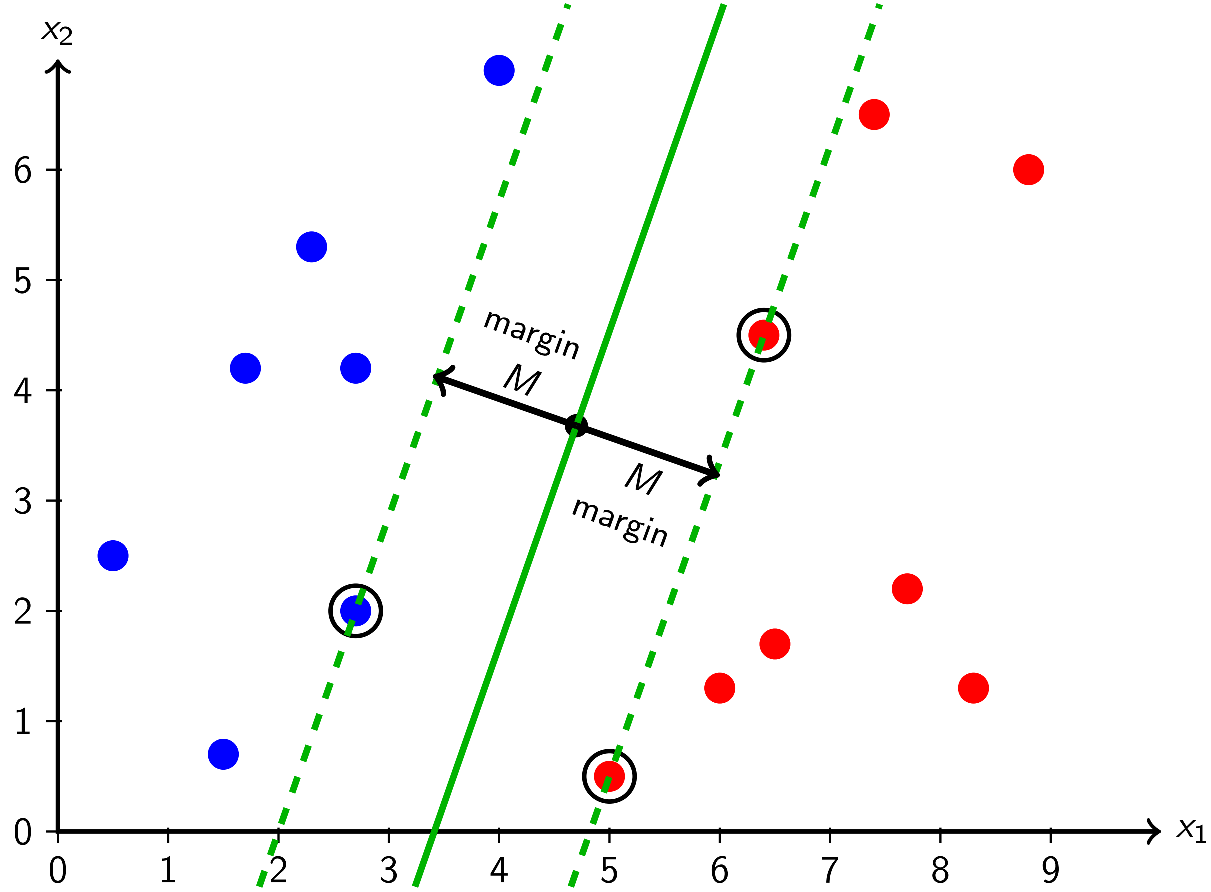


Figure: Optimal hyperplane

Source: Roncalli (2019).

Support vector machines

Hard margin classification

Let $y_i = \beta_0 + x_i^\top \beta$. The maximization problem is:

$$\left\{\hat{\beta}_0, \hat{\beta}\right\} = \arg\max M \quad \text{s.t.} \quad \begin{cases} f(x_i) \geq M & \text{if} \quad y_i = +1 \\ f(x_i) \leq -M & \text{if} \quad y_i = -1 \end{cases}$$

Primal QP

We can show that:

$$\left\{\hat{\beta}_0, \hat{\beta}\right\} = \arg\min \frac{1}{2} \|\beta\|_2^2 \quad \text{s.t.} \quad y_i \left(\beta_0 + x_i^\top \beta\right) \geq 1 \qquad \text{for } i = 1, \ldots, n$$

and $\hat{M} = 1 / \|\beta\|_2$

Support vector machines

Dual QP (Chervonenkis-Cortes-Vapnik)

Let α be the vector of Lagrange multipliers. We have:

$$\begin{aligned} \hat{\alpha} &= \arg\min \frac{1}{2}\alpha^\top \Gamma \alpha - \alpha^\top \mathbf{1}_n \\ &\text{s.t.} \quad \left\{ \begin{array}{l} y^\top \alpha = 0 \\ \alpha \geq \mathbf{0}_n \end{array} \right. \end{aligned}$$

where $\Gamma_{i,j} = y_i y_j x_i^\top x_j$. It follows that $\hat{\beta} = \sum_{i=1}^n \hat{\alpha}_i y_i x_i$ and:

$$\hat{\beta}_0 = \frac{\sum_{i=1}^n \mathbb{1}\left\{\hat{\alpha}_i > 0\right\} \cdot \left(y_i - x_i^\top \hat{\beta}\right)}{\sum_{i=1}^n \mathbb{1}\left\{\hat{\alpha}_i > 0\right\}}$$

We can classify new observations by considering the following rule:

$$\hat{y} = \operatorname{sign}\left(\hat{\beta}_0 + x^\top \hat{\beta}\right)$$

Support vector machines

Dimension of the problem

- Primal QP $\Rightarrow (m+1, n)$
- Dual QP $\Rightarrow (n, n+1)$

Extension to:

- Soft margin classification (binary hinge loss, squared hinge loss, ramp loss, etc.)
- LS-SVM regression
- ε-SVM regression
- Non-linear SVM and kernel functions

Dual QP everywhere!

The Lasso revolution

Least absolute shrinkage and selection operator (lasso)

The lasso method consists in adding a L_1 penalty function to the least square problem:

$$\begin{aligned} \hat{\beta}^{\text{lasso}}(\tau) &= \arg\min \frac{1}{2}(Y - X\beta)^\top (Y - X\beta) \\ \text{s.t.} &\quad \|\beta\|_1 \leq \tau \end{aligned}$$

Alternatively, we have:

$$\hat{\beta}^{\text{lasso}}(\lambda) = \arg\min \frac{1}{2}(Y - X\beta)^\top (Y - X\beta) + \lambda \|\beta\|_1$$

Lasso regression

We have:

$$\mathrm{RSS}(\beta) = \mathrm{RSS}\left(\hat{\beta}^{\mathrm{ols}}\right) + \left(\beta - \hat{\beta}^{\mathrm{ols}}\right)^{\top} X^{\top} X \left(\beta - \hat{\beta}^{\mathrm{ols}}\right)$$

If we consider the equation $\mathrm{RSS}(\beta) = c$, we distinguish three cases:

$c < \mathrm{RSS}\left(\hat{\beta}^{\mathrm{ols}}\right)$	$c = \mathrm{RSS}\left(\hat{\beta}^{\mathrm{ols}}\right)$	$c > \mathrm{RSS}\left(\hat{\beta}^{\mathrm{ols}}\right)$
No solution	One solution $\hat{\beta}^{\mathrm{ols}}$	An ellipsoid

What does this result become when imposing the lasso constraint $\|\beta\|_1 \leq \tau$?

Sparsity theorem

$$\exists \eta > 0 : \forall \tau < \eta, \ \min\left(\left|\hat{\beta}_j^{\mathrm{lasso}}(\tau)\right|\right) = 0$$

The Lasso regression

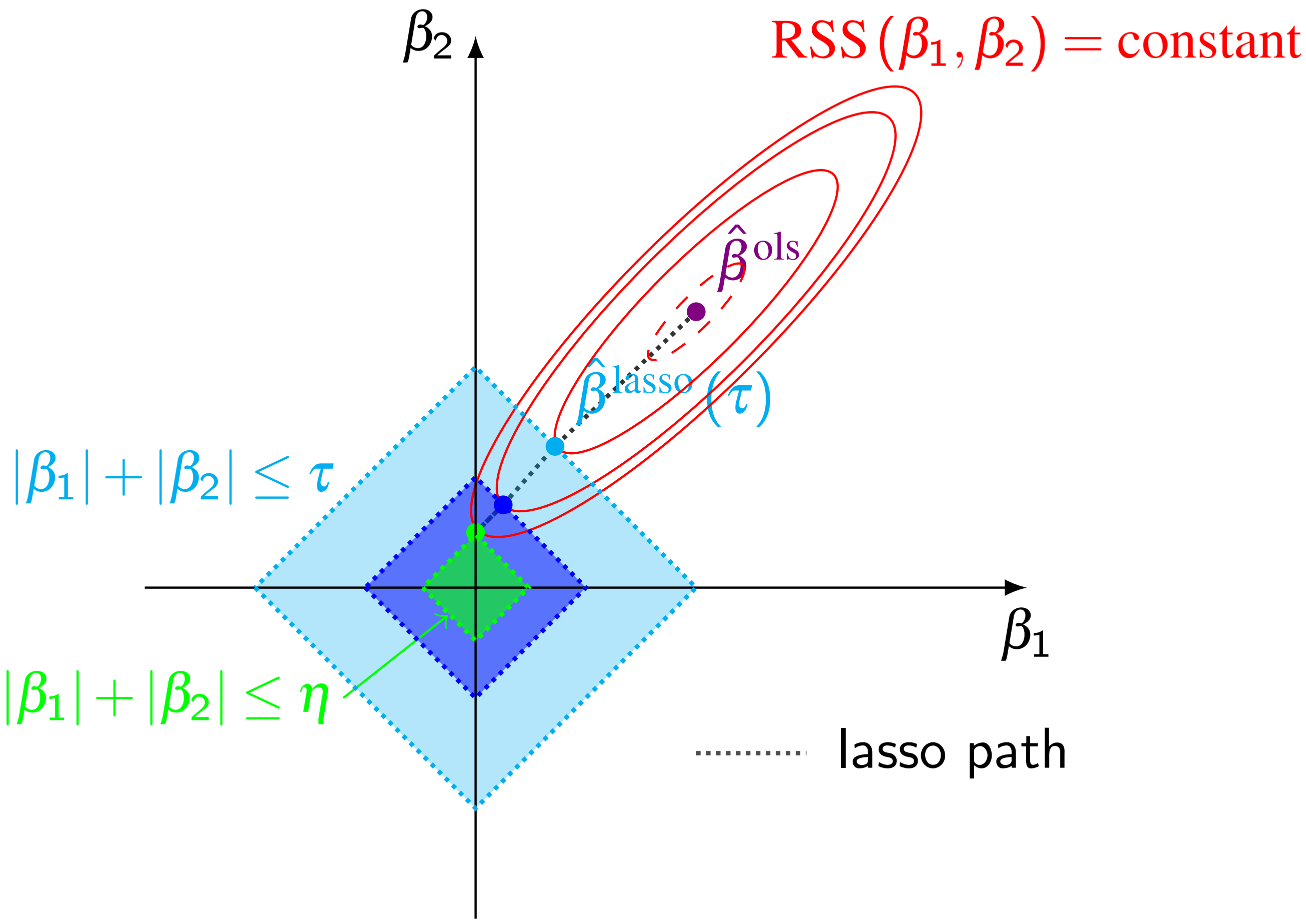
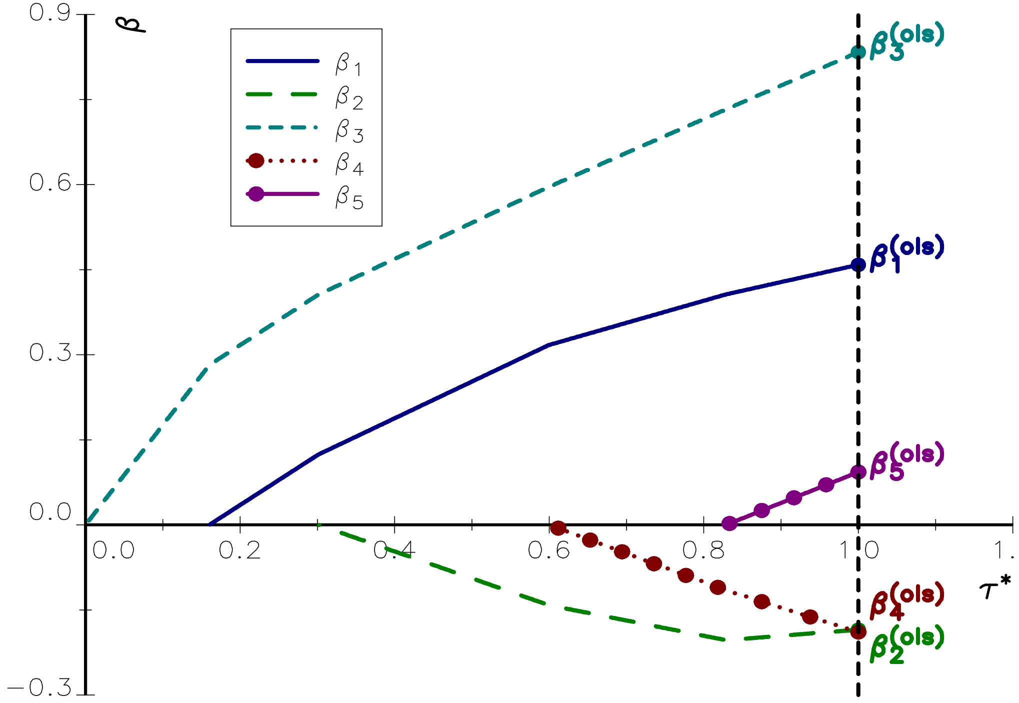


Figure: Interpretation of the lasso regression

Source: Roncalli (2019).

Lasso regression

Figure: Variable selection with the lasso regression

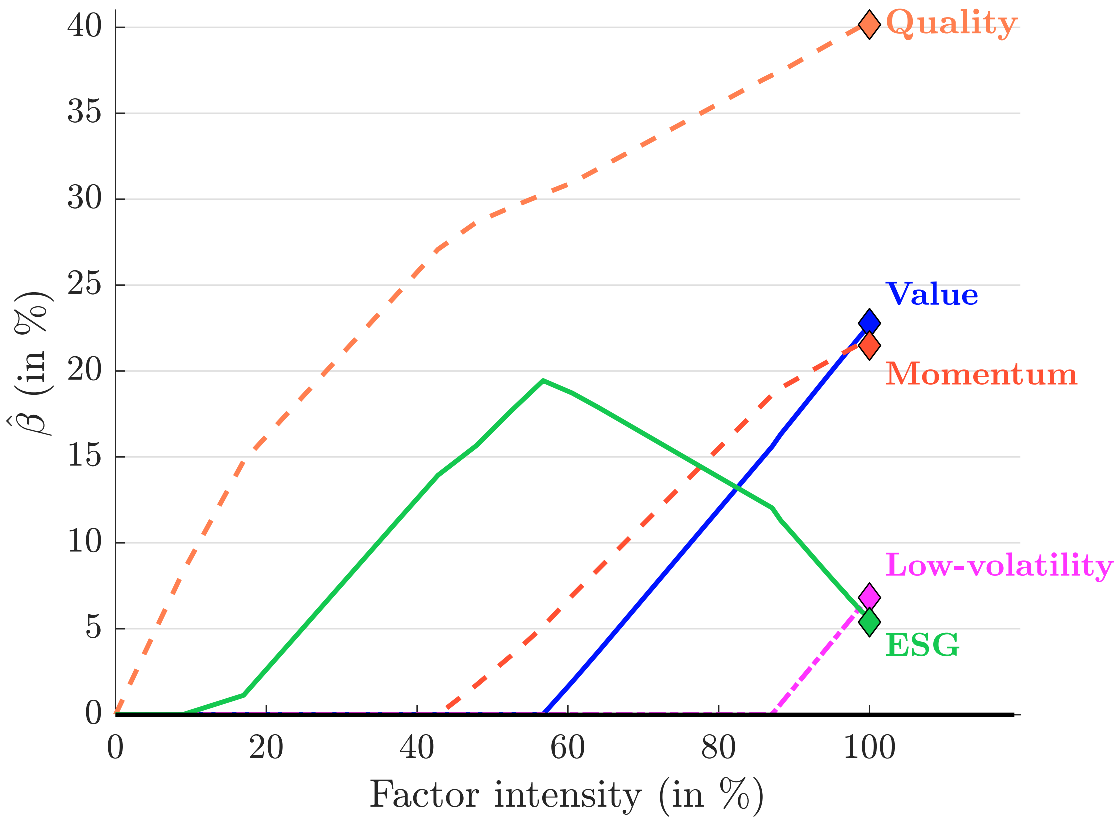


Source: Roncalli (2019).

Lasso ordering: $\mathbf{x}_3 \succ \mathbf{x}_1 \succ \mathbf{x}_2 \succ \mathbf{x}_4 \succ \mathbf{x}_5$

Factor selection in the stock market

Figure: Lasso selection (North America, 2014 – 2017)

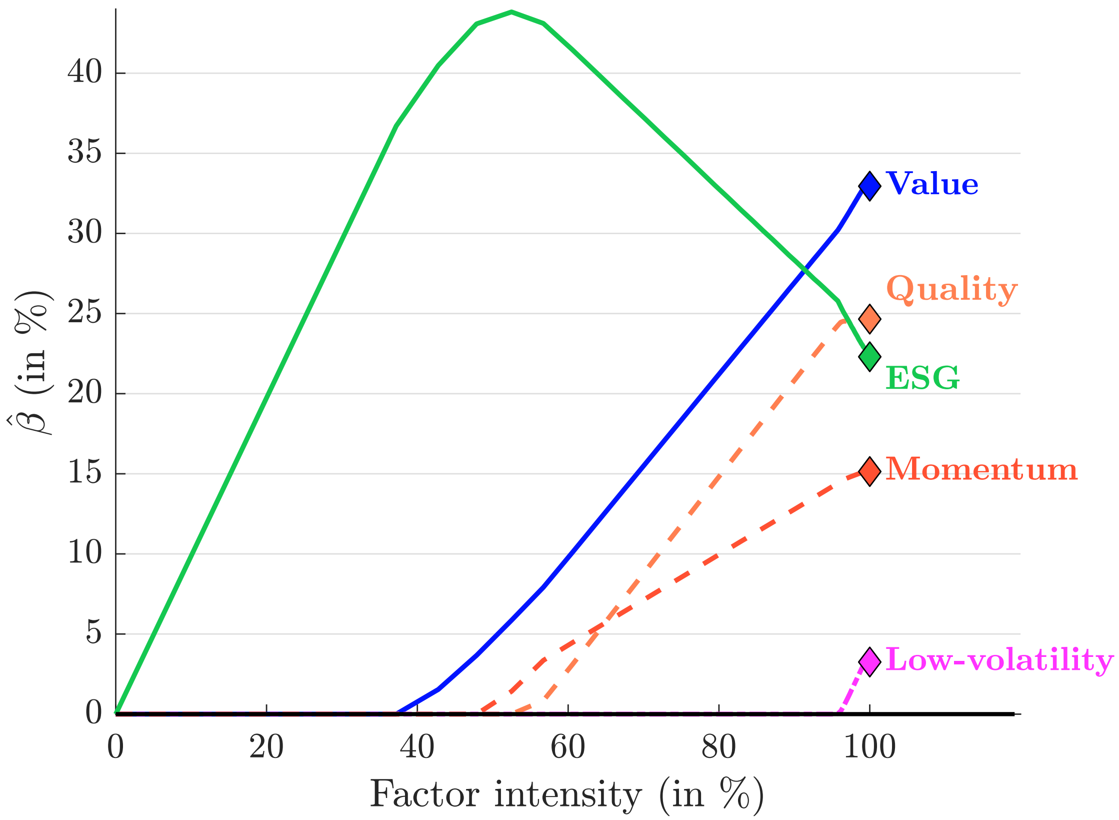


- Quality $\succ$ ESG $\succ$ Momentum $\succ$ Value $\succ$ Low-volatility
- The ESG-Value correlation puzzle!

Source: **Bennani *et al.* (2018).**

Factor selection in the stock market

Figure: Lasso selection (Eurozone, 2014 – 2017)



- ESG ≻ Value ≻ Momentum ≻ Quality ≻ Low-volatility
- The ESG-Quality correlation puzzle!

Source: **Bennani *et al.* (2018).**

Solving the lasso regression problem

We introduce the parametrization:

$$\beta = \beta^{+} - \beta^{-}$$

under the constraints $\beta^{+} \geq \mathbf{0}_n$ and $\beta^{-} \geq \mathbf{0}_n$. We deduce that:

$$\|\beta\|_1 = \sum_{j=1}^{m} \left|\beta_j^{+} - \beta_j^{-}\right| = \sum_{j=1}^{m} \left|\beta_j^{+}\right| + \sum_{j=1}^{m} \left|\beta_j^{-}\right| = \mathbf{1}^\top \beta^{+} + \mathbf{1}^\top \beta^{-}$$

Since we have:

$$\beta = \begin{pmatrix} I_m & -I_m \end{pmatrix} \begin{pmatrix} \beta^{+} \\ \beta^{-} \end{pmatrix}$$

the augmented QP program is specified as follows:

$$\begin{aligned} \hat{\theta} &= \arg\min \frac{1}{2} \theta^\top Q \theta - \theta^\top R \\ \text{s.t.} & \quad \theta \geq \mathbf{0}_{2m} \end{aligned}$$

where $\theta = (\beta^{+}, \beta^{-})$, $\tilde{X} = \begin{pmatrix} X & -X \end{pmatrix}$, $Q = \tilde{X}^\top \tilde{X}$ and $R = \tilde{X}^\top Y + \lambda \mathbf{1}_{2m}$. If we denote $A = \begin{pmatrix} I_m & -I_m \end{pmatrix}$, we obtain $\hat{\beta}^{\text{lasso}}(\lambda) = A\hat{\theta}$

Solving the lasso regression problem

Augmented QP program of the lasso regression

If we consider the τ-problem, we obtain another augmented QP program:

$$\begin{aligned} \hat{\theta} &= \arg\min \frac{1}{2}\theta^\top Q\theta - \theta^\top R \\ &\text{s.t.} \quad \left\{ \begin{array}{l} C\theta \geq D \\ \theta \geq \mathbf{0}_{2m} \end{array} \right. \end{aligned}$$

where $Q = \tilde{X}^\top \tilde{X}$, $R = \tilde{X}^\top Y$, $C = -\mathbf{1}_{2m}^\top$ and $D = -\tau$. Again, we have $\hat{\beta}(\tau) = A\hat{\theta}$

Portfolio allocation with turnover management

Long-only MVO portfolios with a turnover constraint

The optimization problem becomes:

$$
\begin{aligned}
x^{\star} & = \arg\min \frac{1}{2} x^{\top} \Sigma x - \gamma x^{\top} \mu \\
\text{s.t.} & \quad \left\{ \begin{array}{l} \sum_{i=1}^{n} x_i = 1 \\ \sum_{i=1}^{n} \left| x_i - x_i^0 \right| \leq \tau^+ \\ 0 \leq x_i \leq 1 \end{array} \right.
\end{aligned}
$$

where τ^+ is the maximum turnover with respect to Portfolio x^0

Portfolio allocation with turnover management

Scherer (2007) introduces the additional variables x_i^- and x_i^+ such that:

$$x_i = x_i^0 + x_i^+ - x_i^-$$

with $x_i^- \geq 0$ and $x_i^+ \geq 0$. x_i^+ indicates then a positive weight change with respect to the initial weight x_i^0 whereas x_i^- indicates a negative weight change. The expression of the turnover becomes:

$$\sum_{i=1}^{n} \left| x_i - x_i^0 \right| = \sum_{i=1}^{n} \left| x_i^+ - x_i^- \right| = \sum_{i=1}^{n} x_i^+ + \sum_{i=1}^{n} x_i^-$$

because one of the variables x_i^+ or x_i^- is necessarily equal to zero

Portfolio allocation with turnover management

The γ-problem of Markowitz becomes

$$
\begin{aligned}
x^{\star} &= \arg\min \frac{1}{2} x^{\top} \Sigma x - \gamma x^{\top} \mu \\
\text{s.t.} &\quad \left\{
\begin{array}{l}
\sum_{i=1}^{n} x_i = 1 \\
x_i = x_i^0 + x_i^+ - x_i^- \\
\sum_{i=1}^{n} x_i^+ + \sum_{i=1}^{n} x_i^- \leq \tau^+ \\
0 \leq x_i \leq 1 \\
0 \leq x_i^- \leq 1 \\
0 \leq x_i^+ \leq 1
\end{array}
\right.
\end{aligned}
$$

Portfolio allocation with turnover management

We obtain an augmented QP problem of dimension $3n$:

$$\begin{aligned} X^{\star} &= \arg\min \frac{1}{2} X^{\top} Q X - X^{\top} R \\ & \text{s.t.} \quad \left\{ \begin{array}{l} AX = B \\ CX \geq D \\ \mathbf{0}_{3n} \leq X \leq \mathbf{1}_{3n} \end{array} \right. \end{aligned}$$

where:

$$\begin{aligned} X &= \left(x_1, \ldots, x_n, x_1^-, \ldots, x_n^-, x_1^+, \ldots, x_n^+ \right) \\ Q &= \left(\begin{array}{ccc} \Sigma & \mathbf{0}_{n \times n} & \mathbf{0}_{n \times n} \\ \mathbf{0}_{n \times n} & \mathbf{0}_{n \times n} & \mathbf{0}_{n \times n} \\ \mathbf{0}_{n \times n} & \mathbf{0}_{n \times n} & \mathbf{0}_{n \times n} \end{array} \right), R = \left(\begin{array}{c} \mu \\ \mathbf{0}_n \\ \mathbf{0}_n \end{array} \right), A = \left(\begin{array}{ccc} \mathbf{1}_n^{\top} & \mathbf{0}_n^{\top} & \mathbf{0}_n^{\top} \\ I_n & I_n & -I_n \end{array} \right) \\ B &= \left(\begin{array}{c} 1 \\ x^0 \end{array} \right), C = \left(\begin{array}{ccc} \mathbf{0}_n^{\top} & -\mathbf{1}_n^{\top} & -\mathbf{1}_n^{\top} \end{array} \right) \text{ and } D = -\tau^{+} \end{aligned}$$

Extension to transaction costs

Let c_i^- and c_i^+ be the bid and ask transactions costs. The γ-problem of Markowitz becomes:

$$
\begin{aligned}
x^\star & = \arg\min \frac{1}{2} x^\top \Sigma x - \gamma \left(\sum x_i \mu_i - \sum x_i^- c_i^- - \sum x_i^+ c_i^+ \right) \\
& \text{u.c.} \quad \left\{
\begin{array}{l}
\sum x_i + \sum x_i^- c_i^- + \sum x_i^+ c_i^+ = 1 \\
x_i = x_i^0 + x_i^+ - x_i^- \\
0 \leq x_i \leq 1 \\
0 \leq x_i^- \leq 1 \\
0 \leq x_i^+ \leq 1
\end{array}
\right.
\end{aligned}
$$

Extension to transaction costs

We obtain an augmented QP problem of dimension $3n$:

$$
\begin{aligned}
X^\star &= \arg\min \frac{1}{2} X^\top Q X - X^\top R \\
&\text{s.t.} \quad \left\{
\begin{array}{l}
AX = B \\
CX \geq D \\
\mathbf{0}_{3n} \leq X \leq \mathbf{1}_{3n}
\end{array}
\right.
\end{aligned}
$$

where:

$$
\begin{aligned}
X &= \left(x_1, \ldots, x_n, x_1^-, \ldots, x_n^-, x_1^+, \ldots, x_n^+\right) \\
Q &= \left(\begin{array}{ccc}
\Sigma & \mathbf{0}_{n\times n} & \mathbf{0}_{n\times n} \\
\mathbf{0}_{n\times n} & \mathbf{0}_{n\times n} & \mathbf{0}_{n\times n} \\
\mathbf{0}_{n\times n} & \mathbf{0}_{n\times n} & \mathbf{0}_{n\times n}
\end{array}\right), \; R = \left(\begin{array}{c}
\mu \\
-c^- \\
-c^+
\end{array}\right), \\
A &= \left(\begin{array}{ccc}
\mathbf{1}_n^\top & \left(c^-\right)^\top & \left(c^+\right)^\top \\
I_n & I_n & -I_n
\end{array}\right) \text{ and } B = \left(\begin{array}{c}
1 \\
x^0
\end{array}\right)
\end{aligned}
$$

Numerical optimization

The fall and the rise of the steepest-descent method

In the 1980s:

- Conjugate gradient methods (Fletcher–Reeves, Polak–Ribiere, etc.)
- Quasi-Newton methods (NR, BFGS, DFP, etc.)

In the 1990s:

- Neural networks
- Learning rules: Descent, Momentum/Nesterov and Adaptive learning methods

In the 2000s:

- Gradient descent: Batch gradient descent (BGD), Stochatic gradient descent (SGD), Mini-batch gradient descent (MGD)
- Coordinate descent: Cyclical coordinate descent (CCD), Random coordinate descent (RCD)

Numerical optimization

Machine learning problems

- Non-smooth objective function
- Non-unique solution
- Large-scale dimension

Optimization in machine learning requires to reinvent numerical optimization

Coordinate descent methods

Descent method

The descent algorithm is defined by the following rule:

$$x^{(k+1)} = x^{(k)} + \Delta x^{(k)} = x^{(k)} - \eta D^{(k)}$$

At the k^{th} Iteration, the current solution $x^{(k)}$ is updated by going in the opposite direction to $D^{(k)}$ (generally, we set $D^{(k)} = \partial_x f\left(x^{(k)}\right)$)

Coordinate descent method

Coordinate descent is a modification of the descent algorithm by minimizing the function along one coordinate at each step:

$$x_i^{(k+1)} = x_i^{(k)} + \Delta x_i^{(k)} = x_i^{(k)} - \eta D_i^{(k)}$$

$\Rightarrow$ The coordinate descent algorithm becomes a scalar problem

Cyclical coordinate descent (CCD)

Choice of the variable i

1. Random coordinate descent (RCD)
   We assign a random number between 1 and n to the index i (Nesterov, 2012)
2. Cyclical coordinate descent (CCD)
   We cyclically iterate through the coordinates (Tseng, 2001):

$$x_i^{(k+1)} = \arg\min_x f\left(x_1^{(k+1)}, \ldots, x_{i-1}^{(k+1)}, x, x_{i+1}^{(k)}, \ldots, x_n^{(k)}\right)$$

An example

If we consider the following function:

$$f(x_1, x_2, x_3) = (x_1 - 1)^2 + x_2^2 - x_2 + (x_3 - 2)^4 e^{x_1 - x_2 + 3}$$

the CCD algorithm is defined by the following iterations:

$$\left\{ \begin{array}{l} x_1^{(k+1)} = x_1^{(k)} - \eta \left(2 \left(x_1^{(k)} - 1 \right) + \left(x_3^{(k)} - 2 \right)^4 e^{x_1^{(k)} - x_2^{(k)} + 3} \right) \\ x_2^{(k+1)} = x_2^{(k)} - \eta \left(2 x_2^{(k)} - 1 - \left(x_3^{(k)} - 2 \right)^4 e^{x_1^{(k+1)} - x_2^{(k)} + 3} \right) \\ x_3^{(k+1)} = x_3^{(k)} - \eta \left(4 \left(x_3^{(k)} - 2 \right)^3 e^{x_1^{(k+1)} - x_2^{(k+1)} + 3} \right) \end{array} \right.$$

An example (Cont'd)

Table: CCD algorithm ($\eta = 0.25$)

k	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$	$D_1^{(k)}$	$D_2^{(k)}$	$D_3^{(k)}$
0	1.0000	1.0000	1.0000			
1	-4.0214	0.7831	1.1646	20.0855	0.8675	-0.6582
2	-1.5307	0.8834	2.2121	-9.9626	-0.4013	-4.1902
3	-0.2663	0.6949	2.1388	-5.0578	0.7540	0.2932
4	0.3661	0.5988	2.0962	-2.5297	0.3845	0.1703
5	0.6827	0.5499	2.0758	-1.2663	0.1957	0.0818
6	0.8412	0.5252	2.0638	-0.6338	0.0989	0.0480
7	0.9205	0.5127	2.0560	-0.3172	0.0498	0.0314
8	0.9602	0.5064	2.0504	-0.1588	0.0251	0.0222
9	0.9800	0.5033	2.0463	-0.0795	0.0126	0.0166
∞	1.0000	0.5000	2.0000	0.0000	0.0000	0.0000

Linear regression

We consider the linear regression:

$$Y = X\beta + \varepsilon$$

where Y is a $n \times 1$ vector, X is a $n \times m$ matrix and β is a $m \times 1$ vector. The optimization problem is:

$$\hat{\beta} = \arg\min f(\beta) = \frac{1}{2}(Y - X\beta)^\top (Y - X\beta)$$

Since we have $\partial_\beta f(\beta) = -X^\top (Y - X\beta)$, we deduce that:

$$\begin{aligned}
\frac{\partial f(\beta)}{\partial \beta_j} &= x_j^\top (X\beta - Y) \\
&= x_j^\top \left(x_j\beta_j + X_{(-j)}\beta_{(-j)} - Y\right) \\
&= x_j^\top x_j\beta_j + x_j^\top X_{(-j)}\beta_{(-j)} - x_j^\top Y
\end{aligned}$$

where x_j is the $n \times 1$ vector corresponding to the j^{th} variable and $X_{(-j)}$ is the $n \times (m-1)$ matrix (without the j^{th} variable)

Linear regression

At the optimum, we have $\partial_{\beta_j} f(\beta) = 0$ or:

$$\beta_j = \frac{x_j^\top Y - x_j^\top X_{(-j)}\beta_{(-j)}}{x_j^\top x_j} = \frac{x_j^\top \left(Y - X_{(-j)}\beta_{(-j)}\right)}{x_j^\top x_j}$$

CCD algorithm for the linear regression

We have:

$$\beta_j^{(k+1)} = \frac{x_j^\top \left(Y - \sum_{j'=1}^{j-1} x_{j'}\beta_{j'}^{(k+1)} - \sum_{j'=j+1}^{m} x_{j'}\beta_{j'}^{(k)}\right)}{x_j^\top x_j}$$

$\Rightarrow$ Introducing pointwise constraints is straightforward

Lasso regression

The objective function becomes:

$$f(\beta) = \frac{1}{2}(Y - X\beta)^\top (Y - X\beta) + \lambda \|\beta\|_1$$

Since the norm is separable – $\|\beta\|_1 = \sum_{j=1}^m |\beta_j|$, the first-order condition is:

$$x_j^\top (X\beta - Y) + \lambda \partial |\beta_j| = 0$$

CCD algorithm for the lasso regression

We have:

$$\beta_j^{(k+1)} = \frac{1}{x_j^\top x_j} \mathscr{S}_\lambda \left(x_j^\top \left(Y - \sum_{j'=1}^{j-1} x_{j'} \beta_{j'}^{(k+1)} - \sum_{j'=j+1}^{m} x_{j'} \beta_{j'}^{(k)} \right) \right)$$

where $\mathscr{S}_\lambda(v)$ is the soft-thresholding operator:

$$\mathscr{S}_\lambda(v) = \operatorname{sign}(v) \cdot (|v| - \lambda)_+$$

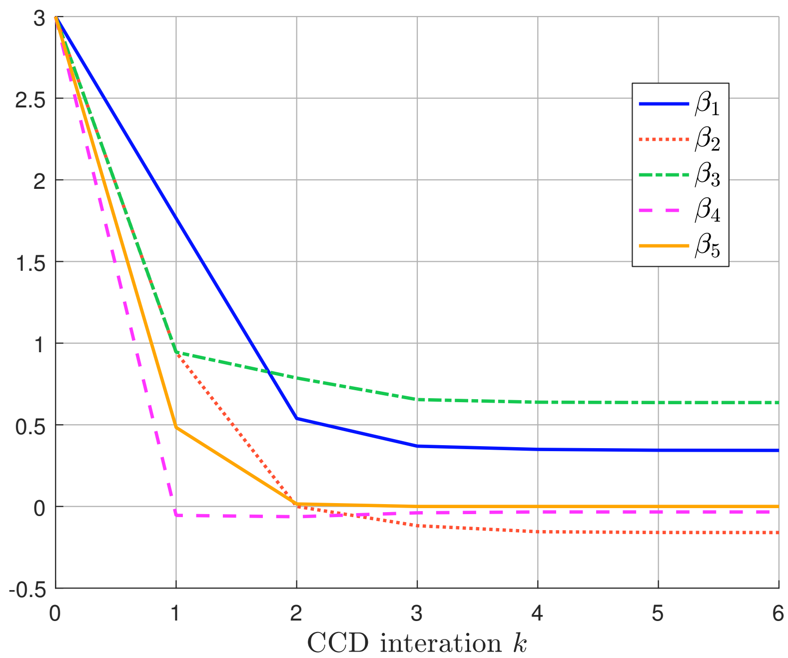
Lasso regression

Table: Matlab code

```
for k = 1:nIters
    for j = 1:m
        x_j = X(:,j);
        X_j = X;
        X_j(:,j) = zeros(n,1);
        if lambda > 0
            v = x_j'*(Y - X_j*beta);
            beta(j) = max(abs(v) - lambda,0) * sign(v) / (x_j'*x_j);
        else
            beta(j) = x_j'*(Y - X_j*beta) / (x_j'*x_j);
        end
    end
end
```

Lasso regression

Figure: Convergence of the CCD algorithm (lasso regression)



1. The dimension problem is $(2m, 2m)$ for QP and $(1, 0)$ for CCD!
2. CCD is faster for lasso regression than for linear regression (because of the soft-thresholding operator)!

Suppose $n = 50\,000$ and $m = 1\,000\,000$ (DNA problem)

Alternative direction method of multipliers

Definition

The alternating direction method of multipliers (ADMM) is an algorithm introduced by Gabay and Mercier (1976) to solve problems which can be expressed as:

$$\begin{aligned} \{x^{\star}, z^{\star}\} &= \arg\min f(x) + g(z) \\ \text{s.t.} &\quad Ax + Bz = c \end{aligned}$$

The algorithm is:

$$\begin{aligned} x^{(k)} &= \arg\min \left\{ f(x) + \frac{\varphi}{2} \left\| Ax + Bz^{(k-1)} - c + u^{(k-1)} \right\|_2^2 \right\} \\ z^{(k)} &= \arg\min \left\{ g(z) + \frac{\varphi}{2} \left\| Ax^{(k)} + Bz - c + u^{(k-1)} \right\|_2^2 \right\} \\ u^{(k)} &= u^{(k-1)} + \left(Ax^{(k)} + Bz^{(k)} - c \right) \end{aligned}$$

An example

We consider the following optimization problem:

$$x^{\star} = \arg\min f(x) \quad \text{s.t.} \quad x^{-} \leq x \leq x^{+}$$

It can be written as:

$$\{x^{\star}, z^{\star}\} = \arg\min f(x) + g(z) \quad \text{s.t.} \quad x - z = \mathbf{0}_n$$

where $g(z) = \mathbb{1}_{\Omega}(x)$ and $\Omega = \{x : x^{-} \leq x \leq x^{+}\}$. By setting $\varphi = \frac{1}{2}$, the z-step becomes:

$$\begin{aligned} z^{(k)} &= \arg\min \left\{ g(z) + \frac{1}{2} \left\| x^{(k)} - z + u^{(k-1)} \right\|_2^2 \right\} \\ &= \mathbf{prox}_g \left(x^{(k)} + u^{(k-1)} \right) \end{aligned}$$

where the proximal operator is the box projection:

$$\mathbf{prox}_g(v) = x^{-} \odot \mathbb{1}\left\{v < x^{-}\right\} + v \odot \mathbb{1}\left\{x^{-} \leq v \leq x^{+}\right\} + x^{+} \odot \mathbb{1}\left\{v > x^{+}\right\}$$

An example (Cont'd)

The ADMM algorithm is then:

$$
\begin{aligned}
x^{(k)} &= \arg\min \left\{ f(x) + \frac{1}{2} \left\| x - z^{(k-1)} + u^{(k-1)} \right\|_2^2 \right\} \\
z^{(k)} &= \mathbf{prox}_g \left(x^{(k)} + u^{(k-1)} \right) \\
u^{(k)} &= u^{(k-1)} + \left(x^{(k)} - z^{(k)} \right)
\end{aligned}
$$

$\Rightarrow$ Solving the constrained optimization problem consists in solving the unconstrained optimization problem, applying the box projection and iterating these steps until convergence

The Cholesky trick

We consider the following problem:

$$
\begin{aligned}
x^{\star} & = \arg\max \mathscr{U}(x) \\
& \text{s.t.} \quad \left\{ \begin{array}{l} x \in \Omega \\ \sqrt{x^{\top}\Sigma x} \leq \bar{\sigma} \end{array} \right.
\end{aligned}
$$

We have:

$$
\begin{aligned}
\{x^{\star}, z^{\star}\} & = \arg\min f(x) + g(z) \\
& \text{s.t.} \quad -Lx + z = \mathbf{0}_n
\end{aligned}
$$

where $f(x) = -\mathscr{U}(x) + \mathbb{1}_{\Omega}(x)$, $g(z) = \mathbb{1}_{\mathscr{E}}(z)$, $\mathscr{E} = \left\{ z \in \mathbb{R}^n : \|z\|_2^2 \leq \bar{\sigma}^2 \right\}$ and L is the upper Cholesky decomposition matrix of Σ:

$$
\|z\|_2^2 = z^{\top}z = x^{\top}L^{\top}Lx = x^{\top}\Sigma x = \sigma^2(x)
$$

$\Rightarrow$ The cholesky trick has been used by Gonzalvez *et al.* (2019) for solving trend-following strategies using the ADMM algorithm in the context of Bayesian learning

Proximal operator

Definition

The proximal operator $\mathbf{prox}_f(v)$ of the function $f(x)$ is defined by:

$$\mathbf{prox}_f(v) = x^\star = \arg\min_x \left\{ f(x) + \frac{1}{2}\|x - v\|_2^2 \right\}$$

If $f(x) = -\ln x$, we have:

$$f(x) + \frac{1}{2}\|x - v\|_2^2 = -\ln x + \frac{1}{2}(x - v)^2 = -\ln x + \frac{1}{2}x^2 - xv + \frac{1}{2}v^2$$

The first-order condition is $-x^{-1} + x - v = 0$. It follows that:

$$\mathbf{prox}_f(v) = \frac{v + \sqrt{v^2 + 4}}{2}$$

If $f(x) = -\lambda \sum_{i=1}^n \ln x_i$, we have $(\mathbf{prox}_f(v))_i = \dfrac{v_i + \sqrt{v_i^2 + 4\lambda}}{2}$

An example

We consider the following optimization problem:

$$x^{\star} = \arg\min f(x) - \lambda \sum_{i=1}^{n} \ln x_i$$

We set $z = x$ and $g(z) = -\lambda \sum_{i=1}^{n} \ln x_i$. The ADMM algorithm becomes

$$\begin{aligned}
x^{(k)} &= \arg\min \left\{ f(x) + \frac{\varphi}{2} \left\| x - z^{(k-1)} + u^{(k-1)} \right\|_2^2 \right\} \\
v^{(k)} &= x^{(k)} + u^{(k-1)} \\
z^{(k)} &= \frac{v^{(k)} + \sqrt{v^{(k)} \odot v^{(k)} + 4\lambda}}{2} \\
u^{(k)} &= u^{(k-1)} + \left(x^{(k)} - z^{(k)} \right)
\end{aligned}$$

If $f(x)$ is a quadratic function, the x-step is straightforward

Proximal operators and projections

If we assume that $f(x) = \mathbb{1}_\Omega(x)$ where Ω is a convex set, we have:

$$\mathbf{prox}_f(v) = \arg\min_x \left\{ \mathbb{1}_\Omega(x) + \frac{1}{2}\|x - v\|_2^2 \right\} = \mathscr{P}_\Omega(v)$$

where $\mathscr{P}_\Omega(v)$ is the standard projection. Parikh and Boyd (2014) show that:

Ω	$\mathscr{P}_\Omega(v)$	Ω	$\mathscr{P}_\Omega(v)$
$Ax = B$	$v - A^\dagger(Av - B)$	$c^\top x \leqslant d$	$v - \dfrac{(c^\top v - d)_+}{\|c\|_2^2} c$
$a^\top x = b$	$v - \dfrac{(a^\top v - b)}{\|a\|_2^2} a$	$x^- \leqslant x \leqslant x^+$	$\mathscr{T}(v; x^-, x^+)$

where $\mathscr{T}(v; x^-, x^+)$ is the truncation operator

Norm constraints

We have $\mathbf{prox}_{\lambda \max}(v) = \min(v, s^{\star})$ where $s^{\star}$ is given by:

$$s^{\star} = \left\{ s \in \mathbb{R} : \sum_{i=1}^{n} (v_i - s)_+ = \lambda \right\}$$

If $f(x)$ is a L_p-norm function and $\mathscr{B}_p(c, \lambda)$ is the L_p-ball with center c and radius λ, we have:

p	$\mathbf{prox}_{\lambda f}(v)$	$\mathscr{P}_{\mathscr{B}_p(\mathbf{0}_n,\lambda)}(v)$
$p = 1$	$S_\lambda(v) = (\lvert v \rvert - \lambda \mathbf{1})_+ \odot \operatorname{sign}(v)$	$v - \mathbf{prox}_{\lambda \max}(\lvert v \rvert) \odot \operatorname{sign}(v)$
$p = 2$	$\left(1 - \dfrac{1}{\max(\lambda, \lVert v \rVert_2)}\right) v$	$v - \mathbf{prox}_{\lambda \lVert \cdot \rVert_2}(\lvert v \rvert)$
$p = \infty$	$\mathbf{prox}_{\lambda \max}(\lvert v \rvert) \odot \operatorname{sign}(v)$	$\mathscr{T}(v; -\lambda, \lambda)$

In the case where the center c is not equal to $\mathbf{0}_n$, we have:

$$\mathscr{P}_{\mathscr{B}_p(c,\lambda)}(v) = \mathscr{P}_{\mathscr{B}_p(\mathbf{0}_n,\lambda)}(v - c) + c$$

ADMM and constraints

We consider the following optimization problem:

$$\begin{aligned} x^{\star} &= \arg\min f(x) \\ \text{s.t.} & \quad x \in \Omega \end{aligned}$$

where Ω is a complex set of constraints:

$$\Omega = \Omega_1 \cap \Omega_2 \cap \cdots \Omega_m$$

We set $z = x$ and $g(z) = \mathbb{1}_{\Omega}(z)$. The ADMM algorithm becomes

$$\begin{aligned} x^{(k)} &= \arg\min \left\{ f(x) + \frac{\varphi}{2} \left\| x - z^{(k-1)} + u^{(k-1)} \right\|_2^2 \right\} \\ v^{(k)} &= x^{(k)} + u^{(k-1)} \\ z^{(k)} &= \mathcal{P}_{\Omega}\left(v^{(k)}\right) \\ u^{(k)} &= u^{(k-1)} + \left(x^{(k)} - z^{(k)}\right) \end{aligned}$$

The question is how to compute $\mathcal{P}_{\Omega}(v)$

Dykstra's algorithm

We consider the proximal problem $x^{\star} = \mathbf{prox}_f(v)$ where $f(x) = \mathbb{1}_{\Omega}(x)$ and:

$$\Omega = \Omega_1 \cap \Omega_2 \cap \cdots \cap \Omega_m$$

The Dykstra's algorithm is:

1. The x-update is:

$$x^{(k)} = \mathscr{P}_{\Omega_{\mathrm{mod}(k,m)}}\left(x^{(k-1)} + z^{(k-m)}\right)$$

2. The z-update is:

$$z^{(k)} = x^{(k-1)} + z^{(k-m)} - x^{(k)}$$

where $x^{(0)} = v$, $z^{(k)} = \mathbf{0}_n$ for $k < 0$ and $\mathrm{mod}(k,m)$ denotes the modulo operator taking values in $\{1,\ldots,m\}$

Dykstra's algorithm

Successive projections of $\mathcal{P}_{\Omega_k}\left(x^{(k-1)}\right)$ does not work!

Successive projections of $\mathcal{P}_{\Omega_k}\left(x^{(k-1)}+z^{(k-m)}\right)$ does work!

Mean-variance optimization with mixed penalties

The Markowitz portfolio optimization problem becomes:

$$
\begin{aligned}
x^{\star} &= \arg\min \frac{1}{2} x^{\top} \Sigma x - \gamma x^{\top} \mu + \frac{1}{2} \rho_2 \left\| \Gamma_2 \left(x - x_0 \right) \right\|_2^2 + \rho_p \left\| \Gamma_p \left(x - x_0 \right) \right\|_p^p \\
\text{s.t.} \quad & x \in \Omega
\end{aligned}
$$

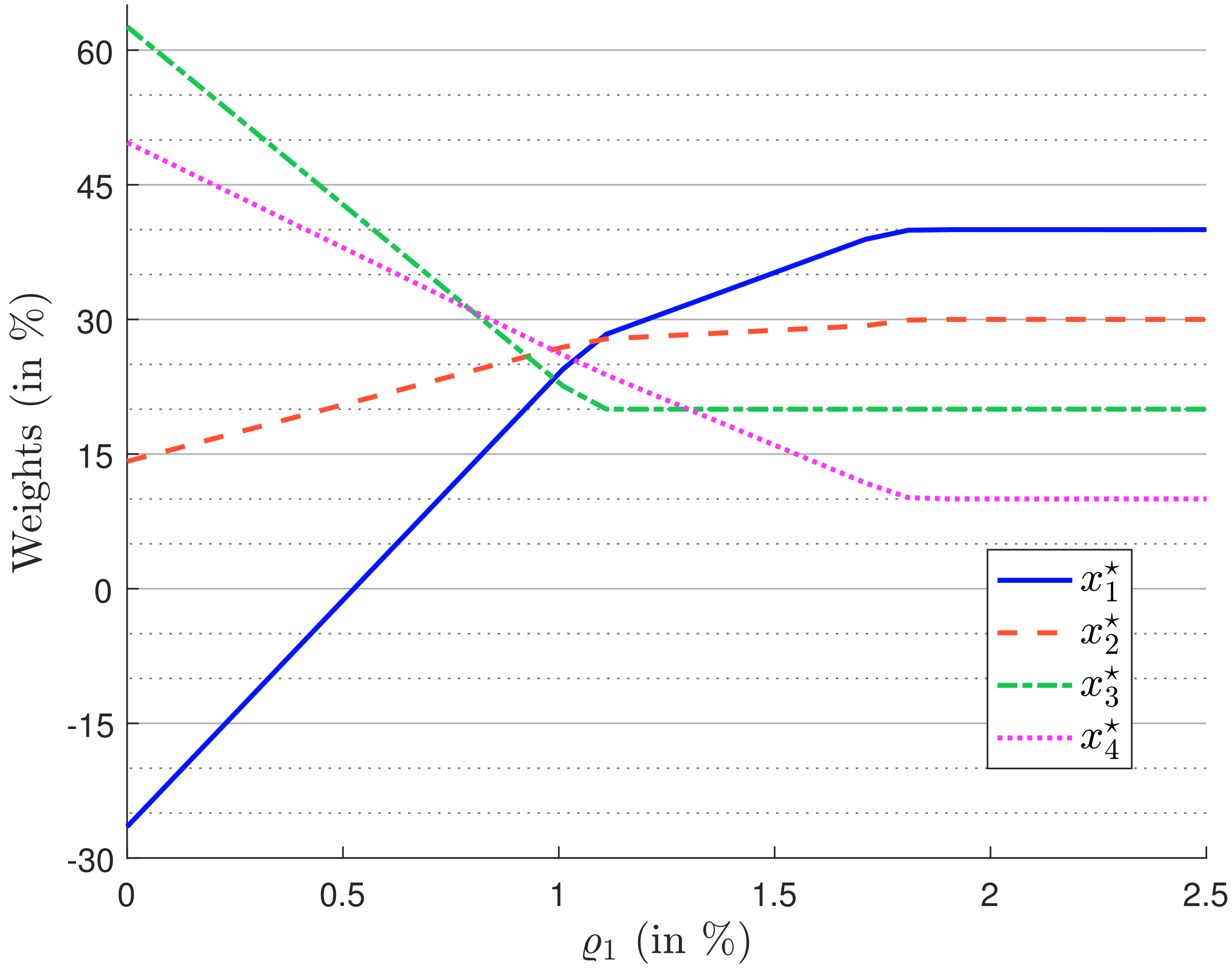
where $p > 0$.

We have the following properties:

- The penalties L_p for $p \geqslant 1$ are used for regularization
- The penalties L_p for $p \leqslant 1$ are used for sparsity
- The case $p = 1$ corresponds to the lasso regression

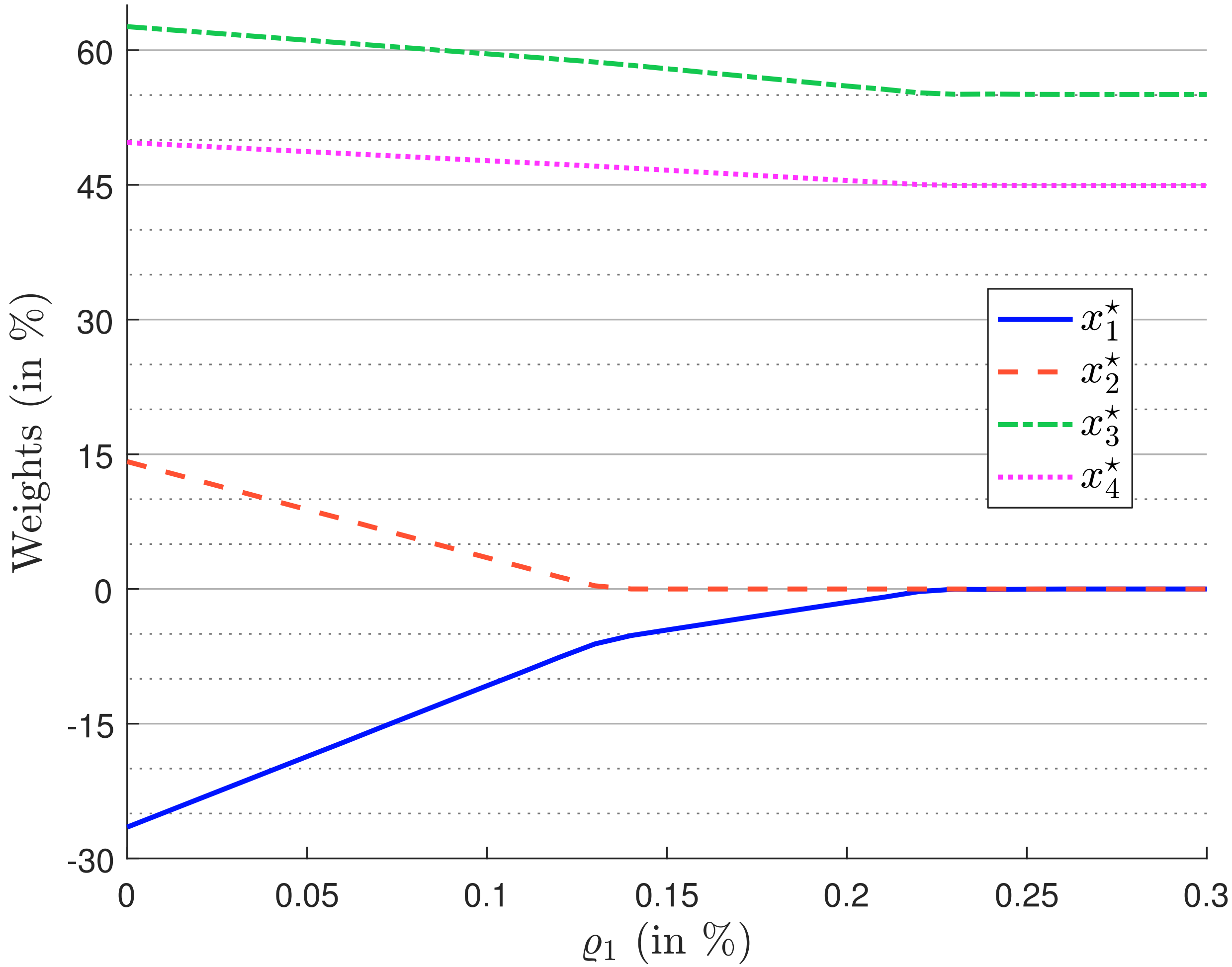
Mixed penalties

Figure: Lasso regularization with a target portfolio (relative sparsity)



Mixed penalties

Figure: Lasso regularization without a target portfolio (absolute sparsity)



Solving the mixed penalty problem

If Ω is a set of linear constraints ($Ax = B$, $Cx \geq D$, $x^{-} \leq x \leq x^{+}$), the mixed penalty problem can be written as:

$$\begin{aligned} \{x^{\star}, z^{\star}\} &= \arg\min f(x) + g(z) \\ \text{s.t.} &\quad x - z = \mathbf{0} \end{aligned}$$

where:

$$f(x) = \frac{1}{2} x^{\top} \Sigma x - \gamma x^{\top} \mu + \frac{1}{2} \varrho_2 \left\| \Gamma_2 (x - x_0) \right\|_2^2 + \mathbb{1}_{\Omega}(x)$$

and:

$$g(z) = \varrho_p \left\| \Gamma_p (z - x_0) \right\|_p^p$$

The ADMM algorithm is implemented as follows:

1. the x-step is a QP problem
2. the z-step is the L_p projection

Solving the mixed penalty problem

If Ω is more complex, the mixed penalty problem can be written as:

$$\begin{aligned} \{x^\star, z^\star\} &= \arg\min f(x) + g(z) \\ \text{s.t.} &\quad x - z = \mathbf{0}_n \end{aligned}$$

where:

$$f(x) = \frac{1}{2} x^\top \Sigma x - \gamma x^\top \mu + \frac{1}{2} \rho_2 \left\| \Gamma_2 (x - x_0) \right\|_2^2 \propto \frac{1}{2} x^\top (\Sigma + \Lambda) x - x^\top (\gamma\mu + \Lambda x_0)$$

$\Lambda = \rho_2 \Gamma_2^\top \Gamma_2$ and:

$$g(z) = \mathbb{1}_\Omega(z) + \rho_p \left\| \Gamma_p (z - x_0) \right\|_p^p$$

The ADMM algorithm is implemented as follows:

1. the x-step is:

$$x^{(k)} = \left(\Sigma + \Lambda + \frac{\varphi}{2} I_n \right)^{-1} \left(\gamma\mu + \Lambda x_0 + \varphi \left(z^{(k-1)} - u^{(k-1)} \right) \right)$$

2. the z-step is given by the Dykstra's algorithm

Risk budgeting portfolio

We consider the following risk measure:

$$\mathcal{R}(x) = -x^\top (\mu - r) + c \cdot \sqrt{x^\top \Sigma x}$$

The risk contribution of Asset i is given by:

$$\mathcal{RC}_i(x) = x_i \cdot \left(-(\mu_i - r) + c \frac{(\Sigma x)_i}{\sqrt{x^\top \Sigma x}} \right)$$

Roncalli (2013) defines the risk budgeting (RB) portfolio as:

$$\left\{ \begin{array}{l} \mathcal{RC}_i(x) = b_i \mathcal{R}(x) \\ b_i > 0, \; x_i \geq 0 \\ \sum_{i=1}^n b_i = 1, \; \sum_{i=1}^n x_i = 1 \end{array} \right.$$

where b_i is the risk budget of Asset i

Wrong formulation of the optimization problem

Since we have:

$$\frac{1}{b_i}\mathscr{RC}_i(x) = \frac{1}{b_j}\mathscr{RC}_j(x) \qquad \text{for all } i,j$$

the RB portfolio is the solution of the optimization problem:

$$\begin{aligned} x_{\mathrm{RB}} &= \arg\min \sum_{i=1}^{n}\sum_{j=1}^{n}\left(\frac{1}{b_i}\mathscr{RC}_i(x) - \frac{1}{b_j}\mathscr{RC}_j(x)\right)^2 \\ & \text{s.t. } \left\{ \begin{array}{l} \mathbf{1}^\top x = 1 \\ x \geq \mathbf{0} \end{array} \right. \end{aligned}$$

Right formulation of the optimization problem

Roncalli (2013) shows that:

$$x_{\mathrm{RB}} = \frac{x^{\star}(\lambda)}{\mathbf{1}^{\top} x^{\star}(\lambda)}$$

where $x^{\star}(\lambda)$ is the solution of the Lagrange problem

$$\begin{aligned} x^{\star}(\lambda) &= \arg\min \mathscr{R}(x) - \lambda \sum_{i=1}^{n} b_i \ln x_i \\ \text{s.t.} \quad & x \geq \mathbf{0} \end{aligned}$$

where λ is an arbitrary positive scalar

The CCD algorithm

Griveau-Billion *et al.* (2013) propose applying the CCD algorithm to find the solution of the objective function:

$$f(x) = -x^\top \pi + c\sqrt{x^\top \Sigma x} - \lambda \sum_{i=1}^{n} b_i \ln x_i$$

where $\pi = \mu - r$. For the cycle $k+1$ and the i^{th} coordinate of the CCD algorithm, we have:

$$x_i = \frac{-c\left(\sigma_i \sum_{j \neq i} x_j \rho_{i,j} \sigma_j\right) + \pi_i \sigma(x) + \sqrt{\left(c\left(\sigma_i \sum_{j \neq i} x_j \rho_{i,j} \sigma_j\right) - \pi_i \sigma(x)\right)^2 + 4\lambda c b_i \sigma_i^2 \sigma(x)}}{2c\sigma_i^2}$$

In this equation, we have the following CCD correspondence:

- $x_i \to x_i^{(k+1)}$
- $x_j \to x_j^{(k+1)}$ if $j < i$
- $x_j \to x_j^{(k)}$ if $j > i$
- $x \to \left(x_1^{(k+1)}, \ldots, x_{i-1}^{(k+1)}, x_i^{(k)}, x_{i+1}^{(k)}, \ldots, x_n^{(k)}\right)$

Theory of constrained risk budgeting

We have

$$\left\{\begin{array}{l} \mathscr{RC}_i(x) = b_i \mathscr{R}(x) \\ x \in \mathscr{S} \\ x \in \Omega \end{array}\right.$$

where $\mathscr{S}$ is the standard simplex and $x \in \Omega$ is the set of additional constraints

The least squares solution

Bai *et al.* (2016) propose to solve the following optimization program:

$$\begin{aligned} \{x^\star(\mathscr{S},\Omega),\theta^\star\} &= \arg\min \sum_{i=1}^{n}\left(\frac{1}{b_i}\mathscr{RC}_i(x)-\theta\right)^2 \\ \text{s.t.} &\quad x\in\mathscr{S}\cap\Omega \end{aligned}$$

The Richard-Roncalli solution

Richard and Roncalli (2019) argue that the right optimization problem is:

$$\begin{aligned} x^{\star}(\mathcal{S},\Omega) &= \arg\min \mathcal{R}(x) \\ & \text{s.t.} \quad \left\{ \begin{array}{l} \sum_{i=1}^{n} b_i \ln x_i \geq \kappa^{\star} \\ x \in \mathcal{S} \cap \Omega \end{array} \right. \end{aligned}$$

where $\kappa^{\star}$ is a constant to be determined. They consider the Lagrange formulation:

$$\begin{aligned} x^{\star}(\Omega,\lambda) &= \arg\min \mathcal{R}(x) - \lambda \sum_{i=1}^{n} b_i \ln x_i \\ \text{s.t.} & \quad x \in \Omega \end{aligned}$$

The constrained risk budgeting portfolio is defined by:

$$x^{\star}(\mathcal{S},\Omega) = \left\{ x^{\star}(\Omega,\lambda^{\star}) : \sum_{i=1}^{n} x_i^{\star}(\Omega,\lambda^{\star}) = 1 \right\}$$

Numerical solution

We note:

$$\mathscr{L}(x;\lambda) = \mathscr{R}(x) - \lambda \sum_{i=1}^{n} b_i \ln x_i + \mathbb{1}_{\Omega}(x)$$

The risk budgeting portfolio is computed by:

1. Solving $x^{\star}(\Omega,\lambda) = \arg\min \mathscr{L}(x;\lambda)$ for a given value of λ (x-step)
2. Finding the optimal value $\lambda^{\star}$ such that $\sum_{i=1}^{n} x_i^{\star}(\Omega,\lambda^{\star}) = 1$ (λ-step)

Bisection algorithm for the λ-step

We consider two scalars a_λ and b_λ such that $a_\lambda < b_\lambda$ and $\lambda^\star \in [a_\lambda, b_\lambda]$
We note ε_λ the convergence criterion of the bisection algorithm
repeat
  We calculate $\lambda = \dfrac{a_\lambda + b_\lambda}{2}$
  We compute $x^\star(\lambda)$ the solution of the minimization problem:

$$x^\star(\lambda) = \arg\min \mathcal{L}(x;\lambda)$$

  if $\sum_{i=1}^{n} x_i^\star(\lambda) < 1$ **then**
    $a_\lambda \leftarrow \lambda$
  else
    $b_\lambda \leftarrow \lambda$
  end if
until $\left| \sum_{i=1}^{n} x_i^\star(\lambda) - 1 \right| \leq \varepsilon_\lambda$
return $\lambda^\star \leftarrow \lambda$ and $x^\star(\mathcal{S}, \Omega) \leftarrow x^\star(\lambda^\star)$

CCD algorithm for the x-step

Thanks to Tseng (2001), CCD algorithm can solve:

$$\arg\min f(x) = f_0(x) + \sum_{i=1}^{n} f_i(x_i)$$

where f_0 is strictly convex and differentiable and the functions f_i are non-differentiable. We have:

$$\mathcal{L}(x;\lambda) = \underbrace{-x^\top \pi + c\sqrt{x^\top \Sigma x} - \lambda \sum_{i=1}^{n} b_i \ln x_i}_{\mathcal{L}_0(x;\lambda)} + \mathbf{1}_\Omega(x)$$

1. For separable constraints $\Omega = \bigcap_{i=1}^{n} \Omega_i$, the CCD algorithm consists in adding the projection $x_i = \mathscr{P}_{\Omega_i}(x_i)$ at each iteration
2. For non-separable constraints, CCD cannot be used

ADMM algorithm for the x-step

We exploit the separability of $\mathcal{L}(x;\lambda)$:

$$\begin{aligned} \{x^\star(\lambda), z^\star(\lambda)\} &= \arg\min f(x) + g(z) \\ \text{s.t.} &\quad x - z = 0 \end{aligned}$$

where:

$$\mathcal{L}(x;\lambda) = \underbrace{\mathcal{R}(x) - \lambda \sum_{i=1}^n b_i \ln x_i}_{f(x)} + \underbrace{\mathbb{1}_\Omega(x)}_{g(x)} \qquad (\#1)$$

or:

$$\mathcal{L}(x;\lambda) = \underbrace{\mathcal{R}(x) + \mathbb{1}_\Omega(x)}_{f(x)} + \underbrace{-\lambda \sum_{i=1}^n b_i \ln x_i}_{g(x)} \qquad (\#2)$$

Formulation	(#1)	(#2)
$\arg\min f^{(k)}(x)$	NR/BFGS/CCD	QP/SQP
$\arg\min g^{(k)}(z)$	Projection/Dykstra	Proximal (logaithmic barrier)

Comprehensive algorithm

Table: Computational time using our Matlab implementation (relative value)

Algorithm	x-update	(1)	(2)	(3)
ADMM	Newton	2	1	1
ADMM	BFGS	380	280	25
ADMM	QP	220	120	110
ADMM	CCD	10	9	8
CCD		1	1	

(1) $\varphi = 1$ + classical bisection

(2) $\varphi = 1$ + accelerated bisection

(3) Adaptive method $\varphi^{(k)}$ + accelerated bisection

Python implementation: CCD and ADMM-QP are the best algorithms!

How does the ERC property hold?

We consider a universe of five assets. Their volatilities are equal to 15%, 20%, 25%, 30% and 10%. The correlation matrix of asset returns is given by the following matrix:

$$\rho = \begin{pmatrix} 1.00 & & & & \\ 0.10 & 1.00 & & & \\ 0.40 & 0.70 & 1.00 & & \\ 0.50 & 0.40 & 0.80 & 1.00 & \\ 0.50 & 0.40 & 0.05 & 0.10 & 1.00 \end{pmatrix}$$

We assume that the current portfolio is $x_0 = (25\%, 25\%, 10\%, 15\%, 30\%)$

We would like to obtain an ERC portfolio with the following constraints:

$$x_0 - 5\% \leq x \leq x_0 + 5\%$$

How does the ERC property hold?

Table: Volatility breakdown (in %) of current and ERC portfolios

Asset	Current portfolio				ERC portfolio			
	x_i	$\mathcal{MR}_i$	$\mathcal{RC}_i$	$\mathcal{RC}_i^\star$	x_i	$\mathcal{MR}_i$	$\mathcal{RC}_i$	$\mathcal{RC}_i^\star$
1	25.00	10.00	2.50	20.21	22.40	10.61	2.38	20.00
2	**25.00**	15.40	3.85	31.10	**16.51**	14.39	2.38	20.00
3	10.00	20.30	2.03	16.41	12.03	19.74	2.38	20.00
4	10.00	22.24	2.22	17.98	10.51	22.60	2.38	20.00
5	**30.00**	5.90	1.77	14.30	**38.54**	6.16	2.38	20.00
$\sigma(x)$			12.37				11.88	

How does the ERC property hold?

Table: Volatility breakdown (in %) of naive and least squares solutions

Asset	Naive solution				Least squares solution			
	x_i	$\mathcal{MR}_i$	$\mathcal{RC}_i$	$\mathcal{RC}_i^\star$	x_i	$\mathcal{MR}_i$	$\mathcal{RC}_i$	$\mathcal{RC}_i^\star$
1	22.84	10.25	2.34	**19.30**	23.13	10.32	2.39	**19.70**
2	**20.00**	14.98	3.00	24.70	**20.00**	14.86	2.97	24.53
3	12.34	20.18	2.49	**20.53**	11.39	20.07	2.29	**18.87**
4	9.83	22.46	2.21	**18.20**	10.48	22.55	2.36	**19.51**
5	**35.00**	5.99	2.10	17.28	**35.00**	6.02	2.11	17.39
$\sigma(x)$			12.13				12.11	

How does the ERC property hold?

Table: Volatility breakdown (in %) of the constrained ERC portfolio

Asset	x_i	$\mathcal{MR}_i$	$\mathcal{RC}_i$	$\mathcal{RC}_i^\star$	λ_i^-	λ_i^+
1	22.89	10.28	2.35	**19.39**	0.00	0.00
2	**20.00**	14.90	2.98	24.55	3.13	0.00
3	11.69	20.13	2.35	**19.39**	0.00	0.00
4	10.42	22.57	2.35	**19.39**	0.00	0.00
5	**35.00**	6.00	2.10	17.29	0.00	0.73
$\sigma(x)$			12.14		$\lambda = 11.76$	

Smart beta portfolios without small cap bias

We consider a CW index composed of seven stocks. The weights are equal to 34%, 25%, 20%, 15%, 3%, 2% and 1%. We assume that the volatilities of these stocks are equal to 15%, 16%, 17%, 18%, 19%, 20% and 21%, whereas the correlation matrix of stock returns is given by:

$$\rho = \begin{pmatrix} 1.00 & & & & & & \\ 0.75 & 1.00 & & & & & \\ 0.73 & 0.75 & 1.00 & & & & \\ 0.70 & 0.70 & 0.75 & 1.00 & & & \\ 0.65 & 0.68 & 0.69 & 0.75 & 1.00 & & \\ 0.62 & 0.65 & 0.63 & 0.67 & 0.70 & 1.00 & \\ 0.60 & 0.60 & 0.65 & 0.68 & 0.75 & 0.80 & 1.00 \end{pmatrix}$$

Smart beta portfolios without small cap bias

- LC-ERC (large cap ERC): Apply the ERC on the large cap universe
- LS-ERC (least squares ERC): Solve the RB portfolio by adding small cap constraints on the LS problem
- C-ERC (Constrained ERC): Solve the RB portfolio by imposing the weight constraints:

$$\begin{cases} 0 \leq x_i & \text{if } i \notin \Omega_{\mathscr{SC}} \\ x_{\mathrm{cw},i} \leq x_i \leq x_{\mathrm{cw},i} & \text{if } i \in \Omega_{\mathscr{SC}} \end{cases}$$

Smart beta portfolios without small cap bias

Table: Volatility breakdown (in %) of constrained ERC portfolios

Asset	CW		ERC		LC-ERC		LS-ERC		C-ERC	
	x_i	$\mathcal{RC}_i^\star$	x_i	$\mathcal{RC}_i^\star$	x_i	$\mathcal{RC}_i^\star$	x_i	$\mathcal{RC}_i^\star$	x_i	$\mathcal{RC}_i^\star$
1	34.00	32.08	17.22	14.29	25.81	**23.39**	26.62	**24.23**	25.87	23.46
2	25.00	24.82	15.90	14.29	24.06	**23.44**	24.20	**23.63**	24.07	23.46
3	20.00	20.92	14.78	14.29	22.44	**23.44**	22.09	**23.08**	22.46	23.46
4	15.00	16.01	13.83	14.29	21.69	**23.57**	21.09	**22.89**	21.59	23.46
5	3.00	3.10	13.17	14.29	3.00	3.10	3.00	3.10	3.00	3.10
6	2.00	2.03	12.86	14.29	2.00	2.02	2.00	2.02	2.00	2.02
7	1.00	1.05	12.23	14.29	1.00	1.05	1.00	1.05	1.00	1.05
$\sigma(x)$	14.50		15.23		14.68		14.66		14.68	

Managing the portfolio turnover

The turnover of Portfolio x with respect to Portfolio x_0 is equal to:

$$\tau(x \mid x_0) = \sum_{i=1}^{n} \left| x_i - x_{0,i} \right| = \| x - x_0 \|_1$$

Therefore, the corresponding Lagrange function is:

$$\mathcal{L}(x;\lambda) = \mathcal{R}(x) - \lambda \sum_{i=1}^{n} b_i \ln x_i + \mathbb{1}_{\Omega}(x)$$

where $\Omega = \{x \in R : \tau(x \mid x_0) \leq \tau^\star\}$ and $\tau^\star$ is the turnover limit. If we use the previous algorithms, the only difficulty is calculating the proximal operator of $g(x) = \mathbb{1}_{\Omega}(x)$:

$$\mathbf{prox}_g(x) = \mathbf{prox}_f(x - x_0) + x_0$$

where $f(x) = \mathbb{1}_{\Omega'}(x)$ and $\Omega' = \{x \in R : \|x\|_1 \leq \tau^\star\}$. We deduce that:

$$\mathbf{prox}_g(x) = x - \mathbf{prox}_{\tau^\star \max}(|x - x_0|) \odot \operatorname{sign}(x - x_0)$$

where $\mathbf{prox}_{\lambda \max}(v)$ is the proximal operator of the pointwise maximum function (see Slide 49)

Managing the portfolio turnover

We consider a universe of eight asset classes: (1) US 10Y Bonds, (2) Euro 10Y Bonds, (3) Investment Grade Bonds, (4) High Yield Bonds, (5) US Equities, (6) Euro Equities, (7) Japan Equities and (8) EM Equities

Table: Volatility and correlation matrix of asset returns (in %)

		1	2	3	4	5	6	7	8
σ_i		5.0	5.0	7.0	10.0	15.0	15.0	15.0	18.0
$\rho_{i,j}$	1	100							
	2	80	100						
	3	60	40	100					
	4	−20	−20	50	100				
	5	−10	−20	30	60	100			
	6	−20	−10	20	60	90	100		
	7	−20	−20	20	50	70	60	100	
	8	−20	−20	30	60	70	70	70	100

Managing the portfolio turnover

We assume that the current allocation is a 50/50 asset mix policy, where the weight of each asset class is 12.5%.

Table: Constrained RB portfolios (in %) with turnover control

Asset	$\tau^{\star}$							
	0.00	10.00	20.00	30.00	40.00	50.00	60.00	70.00
1	12.50	14.86	17.28	19.68	22.01	24.28	26.58	26.83
2	12.50	15.14	17.72	20.32	22.99	25.72	28.42	28.68
3	12.50	12.50	12.50	12.50	12.50	12.50	11.65	11.41
4	12.50	12.50	12.50	12.50	12.50	11.50	9.90	9.80
5	12.50	11.20	9.70	8.49	7.27	6.28	5.66	5.61
6	12.50	12.02	10.36	9.02	7.69	6.63	5.95	5.90
7	12.50	12.50	11.72	10.16	8.66	7.47	6.71	6.66
8	12.50	9.28	8.22	7.33	6.39	5.62	5.14	5.11
$\tau(x^{\star} \mid x_0)$	0.00	10.00	20.00	30.00	40.00	50.00	60.00	61.02

The last column corresponds to the risk parity portfolio (75% of bonds)

Unsolved problems

- Cardinality constraints:

Strategy	Constraints
Sampling	$\operatorname{card}(x_i \neq 0) = m$
Short	$\operatorname{card}(x_i < 0) = m$
Long-/short	$\operatorname{card}(x_i < 0) = \operatorname{card}(x_i > 0)$
Stock picking	$\operatorname{card}(x_i > \varepsilon) = m$

- Scaling puzzle and the homogeneity property of the risk measure

Conclusion

- QP algorithm = universal algorithm in MVO-type asset allocation problems
- Machine learning ⇒ new optimization algorithms
  - Non-smooth objective function
  - Large-scale dimension
- Ridge/Lasso regularization ⇒ basic of modern portfolio optimization
- The 4 pillars are:
  1. CCD
  2. ADMM
  3. Proximal operators
  4. Dykstra's algorithm
- Applications: Robo-advisors, Smart beta portfolios, Dynamic risk parity strategies, Turnover management, etc.

References I

Bennani, L., Le Guenadal, T., Lepetit, F., Ly, L., Mortier, V., Roncalli, T. and Sekine, T. (2018)
How ESG Investing has Impacted the Asset Pricing in the Equity Market?, *Amundi Discussion Paper*, 36.

Bourgeron, T., Lezmi, E., and Roncalli, T. (2018)
Robust Asset Allocation for Robo-Advisors, *SSRN*, `www.ssrn.com/abstract=3261635`.

Cortes, C., and Vapnik, V. (1995)
Support-vector Networks, *Machine Learning*, 20(3), pp. 273-297.

Dykstra, R.L. (1983)
An Algorithm for Restricted Least Squares Regression, *Journal of the American Statistical Association*, 78(384), pp. 837-842.

Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J. (2010)
Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers, *Foundations and Trends® in Machine learning*, 3(1), pp. 1-122.

Friedman, J., Hastie, T., Höfling, H., and Tibshirani, R. (2007)
Pathwise Coordinate Optimization, *Annals of Applied Statistics*, 1(2), pp. 302-332.

References II

Friedman, J., Hastie, T., and Tibshirani, R. (2010)
Regularization Paths for Generalized Linear Models via Coordinate Descent, *Journal of Statistical Software*, 33(1), pp. 1-22.

Gabay, D., and Mercier, B. (1976)
A Dual Algorithm for the Solution of Nonlinear Variational Problems via Finite Element Approximation, *Computers & Mathematics with Applications*, 2(1), pp. 17-40.

Gonzalvez, J., Lezmi, E., Roncalli, T., and Xu, J. (2019)
Financial Applications of Gaussian Processes and Bayesian Optimization, *Amundi Working Paper*, forthcoming.

Gould, N.I., and Toint, P.L. (2000)
A Quadratic Programming Bibliography, *Numerical Analysis Group Internal Report*, 142 pages.

Griveau-Billion, T., Richard, J-C., and Roncalli, T. (2013)
A Fast Algorithm for Computing High-dimensional Risk Parity Portfolios, *SSRN*, `www.ssrn.com/abstract=2325255`.

Markowitz H. (1952)
Portfolio Selection, *Journal of Finance*, 7(1), pp. 77-91.

References III

Markowitz H. (1956)
The Optimization of a Quadratic Function Subject to Linear Constraints, *Naval Research Logistics Quarterly*, 3(1-2), pp. 111-133.

Nesterov, Y. (2012)
Efficiency of Coordinate Descent Methods on Huge-scale Optimization Problems, *SIAM Journal on Optimization*, 22(2), pp. 341-362.

Parikh, N., and Boyd, S. (2014)
Proximal Algorithms, *Foundations and Trends® in Optimization*, 1(3), pp. 127-239.

Richard, J-C., and Roncalli, T. (2019)
Constrained Risk Budgeting Portfolios: Theory, Algorithms, Applications & Puzzles, *SSRN*, `www.ssrn.com/abstract=3331184`.

Roncalli, T. (2013)
Introduction to Risk Parity and Budgeting, Chapman & Hall/CRC Financial Mathematics Series.

Roncalli, T. (2019)
Handbook of Risk Management, forthcoming.

Scherer B. (2007)
Portfolio Construction & Risk Budgeting, Third edition, Risk Books.

References IV

Saha, A., and Tewari, A. (2013)
On the Nonasymptotic Convergence of Cyclic Coordinate Descent Methods, *SIAM Journal on Optimization*, 23(1), pp. 576-601.

Tibshirani, R. (1996)
Regression Shrinkage and Selection via the Lasso, *Journal of the Royal Statistical Society B*, 58(1), pp. 267-288.

Tibshirani, R.J. (2017)
Dykstra's Algorithm, ADMM, and Coordinate Descent: Connections, Insights, and Extensions, in Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (Eds), *Advances in Neural Information Processing Systems*, 30, pp. 517-528.

Tseng, P. (2001)
Convergence of a Block Coordinate Descent Method for Nondifferentiable Minimization, *Journal of Optimization Theory and Applications*, 109(3), pp. 475-494.

Disclaimer