How Machine Learning Can Improve Portfolio Allocation of Robo-Advisors

Thierry Roncalli*

*Amundi Asset Management[1], France.

swissQuant Conference, Zurich, May 16th, 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 research works of Thierry Roncalli, Edmond Lezmi, Thibault Bourgeron, Joan Gonzalvez, Jean-Charles Richard and Jiali Xu.

Definition

"In its primary sense, robo-advisory is a term for defining automated portfolio management. This includes automated trading and rebalancing, but also automated portfolio allocation" (Bourgeron et al., 2018, page 1).

Some figures

US market

- The five largest robo-advisors:
  - Vanguard Personal Advisor Services ($115 bn in 2018 vs $47 bn in 2016)
  - Schwab Intelligent Portfolios ($33 bn in 2018 vs $10 bn in 2016)
  - Betterment ($16 bn in 2018 vs $7 bn in 2016)
  - Wealthfront ($11 bn in 2018)
  - Personal Capital ($8 bn in 2018)
- The tree that hides the forest: 22 new robo-advisors in 2014, 44 new robo-advisors in 2015, etc.
- In 2015, Blackrock acquired FutureAdvisor ($0.8 bn) for a value between 150 and 200 millions of dollar

⇒ What is planned?

$1 trillion of assets in 2020 (OECD, 2017)

Some figures

European market

- It is dominated by the UK: Nutmeg (£1.5 bn), Zen Assets (£1 bn), Fidelity (US/UK), etc.
- Vaamo (Germany), True Wealth (Switzerland), OwlHub (Germany), Moneyfarm (UK/Italy), Scalable Capital (Germany/UK), Yomoni (France), Werthstein (Switzerland), WeSave (France), Fundshop (France), Quirion (Switzerland), Ginmon (Germany), Marie Quantier (France), Descartes Finance (Switzerland), etc.
- Less than €1 bn in 2016, €6 bn in 2017 and €14 bn in 2018

Motivation

- Digitalization of financial services in the US (millennials and others)
- Management fee reduction in Europe

What are the objectives of a robo-advisor?

The underlying idea is to offer a customized service

- Better knowing the individual investor
  - What are the goals of the investor? (saving, retirement, housing, education financing, etc.)
  - What is the risk aversion of the investor?
  - MIFID II compliant
- Building more appropriate asset allocation solutions
  - Robo-advisors claim to offer a customized solution
  - Robo-advisors claim to offer a cost-efficient solution
  - Robo-advisors claim to offer a more-transparent solution

What are the objectives of a robo-advisor?

Reality is different

- Closed system with a few number of products
- Small universe of assets classes
- Web/digital application or robo-advisor?

Robo-advisors today

This is the advisor!

This is the robo!

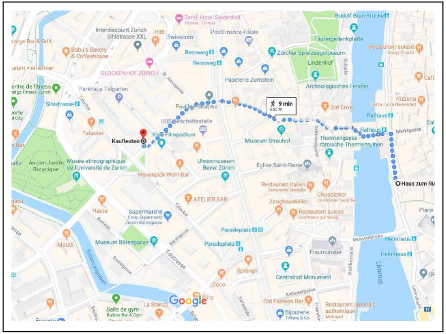
Robo-advisors tomorrow

This is the robo-advisor!

The industry of asset management

This is a map

This is a map



What is different? What remains the same?

The industry of asset management

This is a mobile

This is a smartphone

What is different? What remains the same?

The industry of asset management

You don't buy a product

You buy a service

The industry of asset management

Wealth management Retail investors	Retail distribution	Institutional investors

$\Rightarrow$ Not one robo-advisor but at least 3 robo-advisory concepts:

1. Retail investors (BtoC)
2. Distribution channels (BtoB)
3. Banking networks (BtoB/BtoC)

Retail investors

Mass production versus mass customization

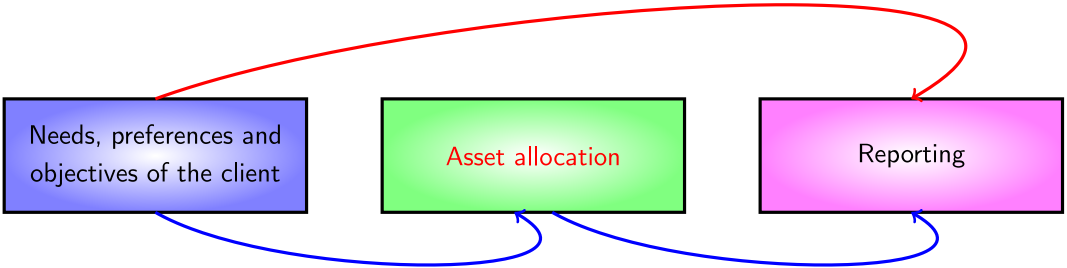
"While mass production has happened a long time ago in investment management through the introduction of mutual funds and more recently exchange traded funds, a new industrial revolution is currently under way, which involves mass customization, a production and distribution technique that will allow individual investors to gain access to scalable and cost-efficient forms of goal-based investing solutions" (Martellini[2], 2016, page 5).

⇒ 3 dimensions:

- KYC
- Bespoke/customized solution
- Client reporting

[2] Martellini, L. (2016), Mass Customization Versus Mass Production – How an Industrial Revolution is About to Take Place in Money Management and Why It Involves a Shift from Investment Products to Investment Solutions, *Journal of Investment Management*, 14(3), pp. 5-13.

Goal-based investing



The main issues are:

- Asset allocation engine
- Producing the reporting according to client needs

Goal-based investing

Portfolio management engine

- ~~Lifestyle~~ ⇒ lifecycle
- Goal setting (client profiling)
- What is the objective function?
  - ~~Mean-variance utility function~~
  - Goal probability
  - Multi-objective function
  - Etc.
- Mass customization (tricky part)
  - Low correlation between income and saving/investment
  - Housing issue and real estate investment
  - Income volatility

Product ⇒ solution

Client reporting

- Return, ~~volatility~~, ~~Sharpe ratio~~, ~~beta~~, ~~Sortino ratio~~, etc.
- Probability to achieve a goal
  - Probability to have a supplementary retirement of \$300 per month, \$500 per month, \$1000 per month, etc.
  - Probability to finance children's education
  - Probability to purchase real estate
- Conflicting goals & trade-off costs between several goals (arbitrage)
- Positive and negative scenarios

The forgotten dimension

Distribution networks

How to manage hundreds of funds/portfolios from an industrial perspective?

25 distributors $\times$ (20 investment portfolios + 20 model portfolios) = 1 000 portfolios $\Rightarrow$ 50 fund managers

A robo-advisor is an integrated system between the asset manager and the distributor:

- Taking into account distributor's active views and/or advisor's active views and/or asset manager's views (tactical asset allocation)
- Open architecture investment platforms $\Rightarrow$ model portfolios $\neq$ investment portfolios
- Plugging thematic funds, sales/marketing campaign, etc.
- Trading ideas testing
- Custom reporting
- Etc.

(Private/retail) banking networks

What is different between robo-advisors for distribution networks and banking networks?

- The allocation is done by the financial advisor
- Private banking $\Rightarrow$ open architecture $\neq$ retail banking $\Rightarrow$ closed architecture
- How to manage sticky positions (direct investment in some emblematic/iconic stocks)

$\Rightarrow$ The goal is to reduce the commercial activity and to increase the advisory activity

$\Rightarrow$ This implies to increase the allocation expertise of financial advisors

The major issue is to build a robo that can be used and understood by financial advisors

Machine learning and robo-advisors

Machine learning

- ~~Big data~~
- ~~Digitalization~~
- Forecasting
- Scoring
- Learning & optimization algorithms

More than a data science

Robo-advisors

- Client profiling
- Expected returns
- Views and active bets
- Self-automated portfolio allocation
- Custom reporting

More than a web application

How machine learning is used in finance?

1. Prediction
   - Trading signals
   - Alternative data
2. Classification
   - Event analysis
   - NLP
3. Optimization & on-line learning
   - Beyond MVO
   - Hyperparameter calibration
   - Optimal control

The issue of portfolio allocation

- Portfolio allocation generally reduces to mean-variance optimization
- Success of MVO portfolios $\Rightarrow$ QP!
- What does diversification mean?
  - Diversification versus hedging
  - Volatility optimization?
- The key parameter: Σ^{-1} and not Σ
- Risk or arbitrage factors?
- Arbitrage factor = hedging portfolios
- Stability of MVO solutions
- The secret sauce of portfolio optimization

Robust asset allocation

- Shrinkage approach (Ledoit-Wolf solution = Tikhonov problem)
- Turnover constraints (L_1 penalization problem)
- How to be sensitive to Σ and not to Σ^{-1}
  - Risk budgeting approach (e.g. equal risk contribution, risk parity, etc.)
  - Logarithmic barrier problems
- Defining the robo-advisor optimization objective function $\Rightarrow$ not a QP problem!
  - Defensive diversified funds (20/80) $\Rightarrow$ RB-based function
  - Dynamic diversified funds (80/20) $\Rightarrow$ MVO-based function
  - Flexible funds $\Rightarrow$ MVO-based between asset classes but RB-based within asset classes

~~one size fits all approach~~

Optimization algorithms for large-scale machine learning problems

- Cyclical coordinate descent (CCD)
- Alternative Direction Method of Multipliers (ADMM)
- Proximal operators (PO)
- Dykstra's algorithm

Solving the robo-advisor asset allocation problem

Two problems:

- Optimization of the portfolio weights
  - Main algorithm: ADMM
  - Sub algorithms: CCD + PO + Dykstra (+ QP)
- Calibration of the hyperparameters and online learning
  - Gaussian processes
  - Bayesian optimization

Machine learning and robo-advisors

And now the mathematics...

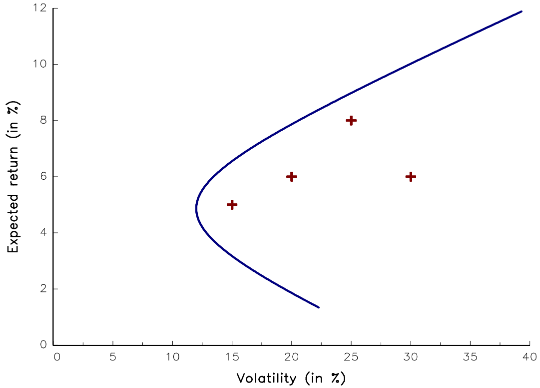
Amundi Quantitative Research on portfolio allocation, machine learning and robo-advisors

Amundi Working Papers

- Alternative Risk Premia: What Do We Know?, WP-61-2017, February 2017.
- Robust Asset Allocation for Robo-Advisors, WP-75-2018, September 2018.
- Constrained Risk Budgeting Portfolios, WP-79-2019, February 2019.
- Financial Applications of Gaussian Processes and Bayesian Optimization, WP-80-2019, March 2019.

`research-center.amundi.com`

Portfolio allocation



Let μ and Σ be 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:

$$\min \sigma(x) \quad \text{s.t.} \quad \mu(x) \geq \mu^\star$$

Portfolio allocation = Mean-variance optimization (MVO)

Why MVO is so popular?

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.

Mean-variance optimization = Quadratic programming

Extension to other asset allocation problems

Definition

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

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

- Portfolio optimization with a benchmark/Tracking-error problems $\Rightarrow$ **Always QP!**
- Active management with views/Black-Litterman model $\Rightarrow$ **Always QP!**
- Index sampling $\Rightarrow$ **Always QP!**
- Turnover management $\Rightarrow$ **Always QP!**
- Linear and quadratic transaction cost models $\Rightarrow$ **Always QP!**

QP everywhere!

Diversification versus hedging

- Diversification: a concept easy to understand?
- Diversification = the search of negative correlations?
- What is the difference between diversification and hedging?
- Diversification = volatility reduction?

What is the issue?

The rule of the game

The mean-variance approach is one of the most aggressive active management models: it concentrates the portfolio on a small number of bets (idiosyncratic factors and arbitrage factors).

⇒ The goal of Markowitz optimization is not to diversify, but to **build active bets** and **leverage** them!

Why MVO portfolios are unstable?

⇒ Because MVO portfolios are sensitive to the precision matrix Σ^{-1} and not directly to the covariance matrix Σ!

Eigendecomposition of the precision matrix

We have $\Sigma = V \Lambda V^\top$ and $\Sigma^{-1} = \left(V \Lambda V^\top\right)^{-1} = V^{\top^{-1}} \Lambda^{-1} V^{-1} = V \Lambda^{-1} V^\top$.

The eigendecomposition of $\mathscr{I}$ is then:

$$V_i(\mathscr{I}) = V_{n-i}(\Sigma) \quad \text{and} \quad \lambda_i(\mathscr{I}) = \frac{1}{\lambda_{n-i}(\Sigma)}$$

⇒ It means that the first factor of the precision matrix corresponds to the last factor of the covariance matrix and that the last factor of the precision matrix corresponds to the first factor.

Illustration of the eigendecomposition of the precision matrix

If we consider the following example: $\sigma_1 = 20\%$, $\sigma_2 = 21\%$, $\sigma_3 = 10\%$ and $\rho_{i,j} = 80\%$, we obtain the following eigendecomposition:

Asset / Factor	Covariance matrix Σ			Precision matrix $\mathcal{I}$		
	1	2	3	1	2	3
1	65.35%	−72.29%	−22.43%	−22.43%	−72.29%	65.35%
2	69.38%	69.06%	−20.43%	−20.43%	69.06%	69.38%
3	30.26%	−2.21%	95.29%	95.29%	−2.21%	30.26%
Eigenvalue	8.31%	0.84%	0.26%	379.97	119.18	12.04
% cumulated	88.29%	97.20%	100.00%	74.33%	97.65%	100.00%

$$12.04 \equiv 1/8.31\%$$

Reverse order of eigenvectors

⇒ **Optimization on the last risk factors: idiosyncratic risk factors and (certainly) noise factors!**

Risk factors versus arbitrage factors

We consider a universe of 6 assets. The volatility is equal respectively to 20%, 21%, 17%, 24%, 20% and 16%. For the correlation matrix, we have:

$$\rho = \begin{pmatrix} 1.00 & & & & & \\ 0.40 & 1.00 & & & & \\ 0.40 & 0.40 & 1.00 & & & \\ 0.50 & 0.50 & 0.50 & 1.00 & & \\ 0.50 & 0.50 & 0.50 & 0.60 & 1.00 & \\ 0.50 & 0.50 & 0.50 & 0.60 & 0.60 & 1.00 \end{pmatrix}$$

$\Rightarrow$ We compute the minimum variance (MV) portfolio with a shortsale constraint

Risk factors versus arbitrage factors

Table: Effect of deleting a PCA factor

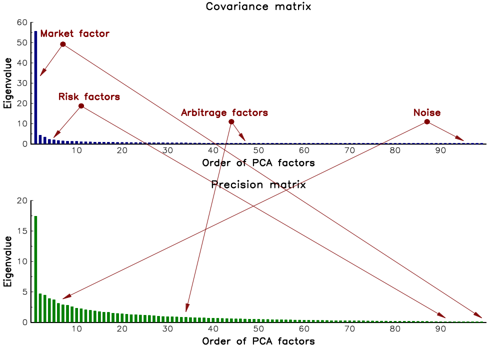
$x^\star$	MV	$\lambda_1 = 0$	$\lambda_2 = 0$	$\lambda_3 = 0$	$\lambda_4 = 0$	$\lambda_5 = 0$	$\lambda_6 = 0$
$x_1^\star$	15.29	15.77	20.79	27.98	0.00	13.40	0.00
$x_2^\star$	10.98	16.92	1.46	12.31	0.00	8.86	0.00
$x_3^\star$	34.40	12.68	35.76	28.24	52.73	53.38	2.58
$x_4^\star$	0.00	22.88	0.00	0.00	0.00	0.00	0.00
$x_5^\star$	1.01	17.99	2.42	0.00	15.93	0.00	0.00
$x_6^\star$	38.32	13.76	39.57	31.48	31.34	24.36	97.42

Source: Roncalli (2013)

$\Rightarrow$ **Deleting the first principal component factor has less impact than deleting the last principal component factor!**

Noise versus arbitrage factors

Figure: Factor decomposition of the FTSE covariance matrix (June 2012)



⇒ Shrinkage is then necessary to eliminate the noise factors, but is not sufficient because it is extremely difficult to filter the arbitrage factors!

Mathematical optimality vs financial optimality

"The indifference of many investment practitioners to mean-variance optimization technology, despite its theoretical appeal, is understandable in many cases. The major problem with MV optimization is its tendency to maximize the effects of errors in the input assumptions. Unconstrained MV optimization can yield results that are inferior to those of simple equal-weighting schemes" (Michaud, 1989).

Are optimized portfolios optimal?

Go to CCD

Arbitrage factors create instability

- We consider a universe of 3 assets.
- The parameters are: $\mu_1 = \mu_2 = 8\%$, $\mu_3 = 5\%$, $\sigma_1 = 20\%$, $\sigma_2 = 21\%$, $\sigma_3 = 10\%$ and $\rho_{i,j} = 80\%$.
- The objective is to maximize the expected return for a 15% volatility target.
- The optimal portfolio is $(38.3\%, 20.2\%, 41.5\%)$.

What is the sensitivity to the input parameters?

ρ		70%	90%		90%	
σ_2				18%	18%	
μ_1						9%
x_1	38.3	38.3	44.6	13.7	−8.0	60.6
x_2	20.2	25.9	8.9	56.1	74.1	−5.4
x_3	41.5	35.8	46.5	30.2	34.0	44.8

What is an arbitrage factor?

The magic formula (Stevens, 1998)

$$x_i^\star = \gamma \frac{\mu_i - \beta_i^\top \mu^{(-i)}}{s_i^2}$$

where:

- β_i is the hedging portfolio of Asset i
- $\beta_i^\top \mu^{(-i)}$ is the expected return of the hedging portfolio
- s_i^2 is the tracking error of the hedging portfolio

$\Rightarrow$ Arbitrage factor = long/short position between an asset and its hedging portfolio

MVO diversification	$\neq$	**Diversification of risk factors**
	$=$	**Concentration on arbitrage factors**

Go to ROBO-ADMM

What is an arbitrage factor?

Table: Hedging portfolios (in %) at the end of 2006

	SPX	SX5E	TPX	RTY	EM	US HY	EMBI	EUR	JPY	GSCI
SPX		58.6	6.0	150.3	-30.8	-0.5	5.0	-7.3	15.3	-25.5
SX5E	9.0		-1.2	-1.3	35.2	0.8	3.2	-4.5	-5.0	-1.5
TPX	0.4	-0.6		-2.4	38.1	1.1	-3.5	-4.9	-0.8	-0.3
RTY	48.6	-2.7	-10.4		26.2	-0.6	1.9	0.2	-6.4	5.6
EM	-4.1	30.9	69.2	10.9		0.9	4.6	9.1	3.9	33.1
US HY	-5.0	53.5	160.0	-18.8	69.5		95.6	48.4	31.4	-211.7
EMBI	10.8	44.2	-102.1	12.3	73.4	19.4		-5.8	40.5	86.2
EUR	-3.6	-14.7	-33.4	0.3	33.8	2.3	-1.4		56.7	48.2
JPY	6.8	-14.5	-4.8	-8.8	12.7	1.3	8.4	50.4		-33.2
GSCI	-1.1	-0.4	-0.2	0.8	10.7	-0.9	1.8	4.2	-3.3	
$\hat{s}_i$	0.3	0.7	0.9	0.5	0.7	0.1	0.2	0.4	0.4	1.2
$\mathcal{R}_i^2$	83.0	47.7	34.9	82.4	60.9	39.8	51.6	42.3	43.7	12.1

Source: Bruder *et al.* (2013)

What is an arbitrage factor?

The second magic formula (Bourgeron *et al.*, 2018)

$$x_i^{\star} = y_i^{\star} + \omega_i \left(y_i^{\star} - z_i^{\star} \right)$$

where:

- $y^{\star}$ is the optimal portfolio by assuming zero correlation
- $z^{\star}$ is the optimal portfolio of the hedging strategies
- ω_i is the leverage defined by:

$$\omega_i = \frac{\sigma_i^2 - s_i^2}{s_i^2} = \frac{\text{idiosyncratic variance}}{\text{tracking error variance}}$$

MVO diversification = Leverage/hedging strategy?

Implementing portfolio optimization in practice

The question

How to be exposed to common risk factors with Markowitz optimization?

The most frequent answer

Quants impose discretionary constraints:

- The secret sauce of portfolio allocation 🙂
- WYWIWYG (what you want is what you get) 🙁

The secret sauce of portfolio optimization

It is based on the iterative process:

$$
\begin{aligned}
x^\star_{(k)} &= \arg\min \frac{1}{2} x^\top \Sigma x - \gamma x^\top \mu \\
& \text{s.t.} \quad \left\{ \begin{array}{l} \mathbf{1}^\top x = 1 \\ \mathbf{0} \leqslant x \leqslant \mathbf{1} \\ x \in \Omega_{(k)} \end{array} \right.
\end{aligned}
$$

where $\Omega_{(0)} = \mathbb{R}^n$ and k is the step.

This iterative process can be represented by the sequence $\boldsymbol{P}$ defined as follows:

$$
\boldsymbol{P} = \left\{ x^\star_{(0)}, \Omega_{(1)}, x^\star_{(1)}, \Omega_{(2)}, x^\star_{(2)}, \Omega_{(3)}, x^\star_{(3)}, \ldots \right\}
$$

An example of strategic asset allocation

Table: Expected returns and risks (in %)

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
μ_i	4.2	3.8	5.3	10.4	9.2	8.6	5.3	11.0	8.8
σ_i	5.0	5.0	7.0	10.0	15.0	15.0	15.0	18.0	30.0

Table: Correlation matrix of asset returns (in %)

		(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
US 10Y Bonds	(1)	100								
Euro 10Y Bonds	(2)	80	100							
IG Bonds	(3)	60	40	100						
HY Bonds	(4)	−20	−20	50	100					
US Equities	(5)	−10	−20	30	60	100				
Euro Equities	(6)	−20	−10	20	60	90	100			
Japan Equities	(7)	−20	−20	20	50	70	60	100		
EM Equities	(8)	−20	−20	30	60	70	70	70	100	
Commodities	(9)	0	0	10	20	20	20	30	30	100

Building a strategic asset allocation

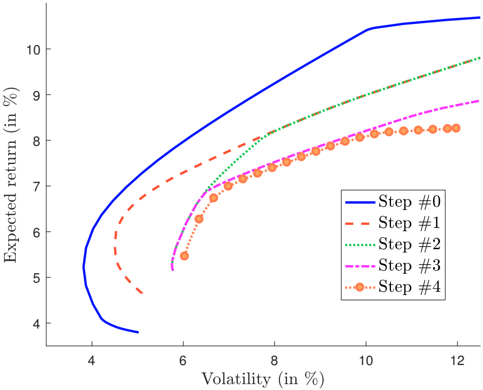
Table: The iterative trial-and-error solutions

Step k		#0	#1	#2	#3	#4	$\cdots$	#K
US 10Y Bonds	(1)	28.39	25.00	24.99	25.00	12.13		10.00
Euro 10Y Bonds	(2)	0.00	15.90	18.60	16.50	22.13		30.00
IG Bonds	(3)	0.00	0.00	0.00	4.86	15.00		10.00
HY Bonds	(4)	69.64	25.00	16.41	10.00	10.00		5.00
US Equities	(5)	0.00	10.70	20.86	25.00	10.00		10.00
Euro Equities	(6)	0.00	0.00	3.16	5.00	20.00		20.00
Japan Equities	(7)	0.00	0.00	0.00	0.00	0.00		5.00
EM Equities	(8)	1.17	21.27	15.98	10.00	10.00		8.00
Commodities	(9)	0.79	2.13	0.00	3.64	0.73		2.00
$\mu(x)$		8.63	7.77	7.41	7.12	6.99		6.57
$\sigma(x)$		7.00	7.00	7.00	7.00	7.00		6.84
$\mathrm{SR}(x \mid r)$		80.49	68.08	63.03	58.93	57.00		52.17

$\Rightarrow$ cap of 25%, then at least 40% of equity, then Euro $>$ US, then JPY $>$ 5%, etc.

Scientific legitimacy of portfolio optimization?

Figure: How does the secret sauce of portfolio optimization work?



How to model portfolio allocation of a robo-advisor?

We must complement Markowitz optimization by:

- introducing smoothness of the solution and/or
- imposing sparsity of the solution and/or
- introducing smoothness of rebalancing and/or
- imposing sparsity of rebalancing and/or
- considering risk factors instead of arbitrage factors and/or
- etc.

$\Rightarrow$ We do not manage a defensive 20/80 diversified fund in the same way than an aggressive 80/20 diversified fund

How to smooth the allocation?

The covariance shrinkage approach

- Let $\hat{\Sigma}$ be the empirical covariance matrix. It is an unbiased estimator, but its convergence is very slow
- Let $\hat{\Phi}$ be another estimator which is biased but converges more quickly

Ledoit and Wolf (2003) propose to combine $\hat{\Sigma}$ and $\hat{\Phi}$:

$$\hat{\Sigma}(\alpha) = \alpha \hat{\Sigma} + (1 - \alpha) \hat{\Phi}$$

The value of α is estimated by minimizing a quadratic loss:

$$\alpha^\star = \arg\min \mathbb{E}\left[\left\| \hat{\Sigma}(\alpha) - \Sigma \right\|^2\right]$$

How to smooth the allocation?

Tikhonov regularization

- The Tikhonov problem can be written as follows:

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

where $\rho_2 > 0$ is a positive number, Γ_2 is a $n \times n \in$ matrix and x_0 is an initial portfolio

- The Ledoit-Wolf covariance shrinkage method is a special case:

$$\left\{ \begin{array}{l} \rho_2 = \dfrac{1-\alpha^{\star}}{\alpha^{\star}} \\ \Gamma_2 = \operatorname{chol} \hat{\Phi} \end{array} \right.$$

- The double shrinkage method is obtained by setting $\Gamma_2 = I_n$ and $x_0 \neq \mathbf{0}$
- The ridge regularization is defined by $\Gamma_2 = I_n$

How to smooth the allocation?

Tikhonov regularization

Figure: Ridge solution with a target portfolio

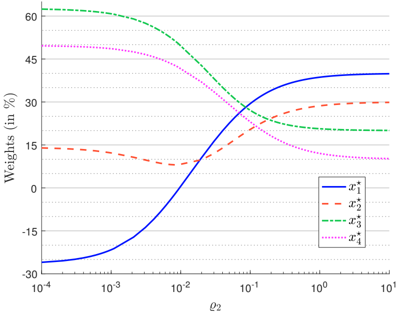
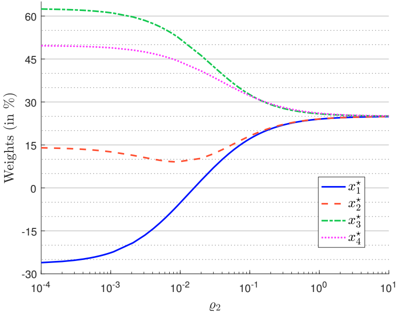


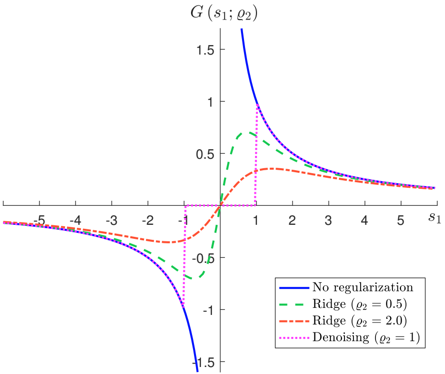
Figure: Ridge solution without a target portfolio



How to smooth the allocation?

Relationship between Tikhonov regularization and spectral filtering

Figure: Inverse of singular values (or eigenvalues)



How to sparsify the allocation?

Rebalancing management

- Turnover constraint: $\sum_{i=1}^{n} \left| x_i - x_i^0 \right| \leq \tau^+$
- Rebalancing costs: $\sum_{i=1}^{n} \left(x_i^- c_i^- + x_i^+ c_i^+ \right)$ where c_i^- and c_i^+ are the bid and ask costs

$\Rightarrow$ Special cases of the lasso problem:

$$\begin{aligned} x^\star & = \arg\min \frac{1}{2} x^\top \Sigma x - \gamma x^\top \mu + \rho_1 \left\| \Gamma_1 \left(x - x_0 \right) \right\|_1 \\ \text{s.t.} & \quad x \in \Omega \end{aligned}$$

where $\rho_1 > 0$ is a positive number, Γ_1 is a $n \times n \in$ matrix and x_0 is an initial portfolio

How to be sensitive to Σ and not to Σ^{-1}?

Let $x = (x_1, \ldots, x_n)$ be the weights of n assets in the portfolio. Let $\mathscr{R}(x_1, \ldots, x_n)$ be a coherent and convex risk measure. We have:

$$\begin{aligned} \mathscr{R}(x_1, \ldots, x_n) &= \sum_{i=1}^{n} x_i \cdot \frac{\partial \mathscr{R}(x_1, \ldots, x_n)}{\partial x_i} \\ &= \sum_{i=1}^{n} \mathrm{RC}_i(x_1, \ldots, x_n) \end{aligned}$$

Let $b = (b_1, \ldots, b_n)$ be a vector of budgets such that $b_i \geq 0$ and $\sum_{i=1}^{n} b_i = 1$. The risk budgeting portfolio is defined by:

$$\mathrm{RC}_i = b_i \cdot \mathscr{R}(x_1, \ldots, x_n)$$

How to be sensitive to Σ and not to Σ^{-1}?

Illustration

- 3 assets
- Volatilities are respectively 30%, 20% and 15%
- Correlations are set to 80% between the 1st asset and the 2nd asset, 50% between the 1st asset and the 3rd asset and 30% between the 2nd asset and the 3rd asset
- Budgets are set to 50%, 20% and 30%
- For the ERC (Equal Risk Contribution) portfolio, all the assets have the same risk budget

Weight budgeting (or traditional) approach

Asset	Weight	Marginal Risk	Risk Contribution Absolute	Risk Contribution Relative
1	50.00%	29.40%	14.70%	70.43%
2	20.00%	16.63%	3.33%	15.93%
3	30.00%	9.49%	2.85%	13.64%
Volatility			20.87%	

Risk budgeting approach

Asset	Weight	Marginal Risk	Risk Contribution Absolute	Risk Contribution Relative
1	31.15%	28.08%	8.74%	50.00%
2	21.90%	15.97%	3.50%	20.00%
3	46.96%	11.17%	5.25%	30.00%
Volatility			17.49%	

ERC approach

Asset	Weight	Marginal Risk	Risk Contribution Absolute	Risk Contribution Relative
1	19.69%	27.31%	5.38%	33.33%
2	32.44%	16.57%	5.38%	33.33%
3	47.87%	11.23%	5.38%	33.33%
Volatility			16.13%	

How to be sensitive to Σ and not to Σ^{-1}?

In the case of the volatility risk measure, risk budgeting is equivalent to solve the logarithmic barrier problem:

$$x^\star = \arg\min \frac{1}{2} x^\top \Sigma x - \lambda \sum_{i=1}^{n} b_i \ln x_i$$

$\Rightarrow$ Extension to Markowitz mean-variance utility function

Defining the optimization problem of the robo-advisor

The optimization problem becomes:

$$
\begin{aligned}
x_{t+1}^{\star} = & \arg\min \frac{1}{2}(x-b)^{\top}\Sigma_t(x-b) - \gamma(x-b)^{\top}\mu_t + \tilde{\rho}_1 \left\|\tilde{\Gamma}_1(x-\tilde{x})\right\|_1 + \\
& \frac{1}{2}\tilde{\rho}_2 \left\|\tilde{\Gamma}_2(x-\tilde{x})\right\|_2^2 + \rho_1 \left\|\Gamma_1(x-x_t)\right\|_1 + \frac{1}{2}\rho_2 \left\|\Gamma_2(x-x_t)\right\|_2^2 - \lambda \sum_{i=1}^{n} b_i \ln x_i \\
\text{s.t.} & \left\{ \begin{array}{l} \mathbf{1}^{\top}x = 1 \\ \mathbf{0} \leqslant x \leqslant \mathbf{1} \\ x \in \Omega \end{array} \right.
\end{aligned}
$$

where b is the benchmark portfolio, $\tilde{x}$ is the reference (or SAA) portfolio and x_t is the current portfolio

This is not a QP problem!

Optimization algorithms for large-scale machine learning problems

Fantastic Four

- Cyclical Coordinate Descent (CCD)
- Alternative Direction Method of Multipliers (ADMM)
- Proximal operators
- Dykstra's algorithm

Coordinate descent methods

The fall and the rise of the 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 (by **observations**): Batch gradient descent (BGD), Stochastic gradient descent (SGD), Mini-batch gradient descent (MGD)
- Gradient descent (by **parameters**): Coordinate descent (CD), cyclical coordinate descent (CCD), Random coordinate descent (RCD)

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):

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

Cyclical coordinate descent (CCD)

Application to the 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)

Cyclical coordinate descent (CCD)

Application to the 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

Cyclical coordinate descent (CCD)

Application to the 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:

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

where $\mathcal{S}_\lambda(v)$ is the soft-thresholding operator: $\mathcal{S}_\lambda(v) = \operatorname{sign}(v) \cdot (|v| - \lambda)_+$

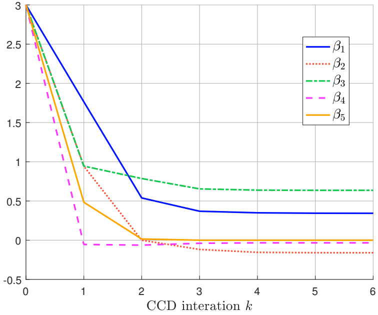
CCD code of the lasso regression

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
```

Convergence of the CCD algorithm applied to the 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)

Application to the risk budgeting 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

$\Rightarrow$ We obtain a logarithmic barrier problem

Solving the risk budgeting problem with the CCD algorithm

In the case of the Markowitz utility function, the objective function is equal to:

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

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

$\Rightarrow$ Extension to many asset allocation problems with a logarithmic barrier or smart beta optimization problems (Richard and Roncalli, 2015).

Go to GP

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 of ADMM

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:

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

An example of ADMM (cont'd)

The ADMM algorithm is then:

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

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

ADMM and the Cholesky trick

We consider the following problem:

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

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 of proximal operator

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 (e.g. the ERC portfolio)

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:

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

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

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

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

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

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

Application to the mean-variance optimization with the mixed $L_1 - L_2$ penalty

The Markowitz portfolio optimization problem becomes:

$$\begin{aligned} x^\star &= \arg\min \frac{1}{2} x^\top \Sigma x - \gamma x^\top \mu + \frac{1}{2}\rho_2 \left\| \Gamma_2 \left(x - x_0\right) \right\|_2^2 + \rho_1 \left\| \Gamma_1 \left(x - x_0\right) \right\|_1 \\ \text{s.t.} &\quad x \in \Omega \end{aligned}$$

Solving the mixed penalty problem with ADMM

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:

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

and:

$$g(z) = \rho_1 \left\| \Gamma_1 (z - x_0) \right\|_1$$

The ADMM algorithm is implemented as follows:

1. the x-step is a QP problem
2. the z-step is the L_1 projection (thresholding operator of the lasso proximal operator)

Solving the mixed penalty problem with the Dykstra's algorithm

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:

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

and:

$$g(z) = \mathbb{1}_\Omega (z) + \rho_1 \left\| \Gamma_1 (z - x_0) \right\|_1$$

The ADMM algorithm is implemented as follows:

1. the x-step is:

$$x^{(k)} = \left(\Sigma + \rho_2 \Gamma_2^\top \Gamma_2 + \frac{\varphi}{2} I_n \right)^{-1} \left(\gamma \mu + \rho_2 \Gamma_2^\top \Gamma_2 x_0 + \varphi \left(z^{(k-1)} - u^{(k-1)} \right) \right)$$

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

Back to optimization problem of the robo-advisor

We have:

$$x_{t+1}^{\star} = \arg\min \varphi_{\mathrm{MV}}(x) + \varphi_{L_2}(x) + \mathbf{1}_{\Omega}(x) + \varphi_{L_1}(x) - \lambda \sum_{i=1}^{n} b_i \ln x_i$$

where:

$$\begin{aligned} \varphi_{\mathrm{MV}}(x) &= \frac{1}{2}(x-b)^{\top}\Sigma_t(x-b) - \gamma(x-b)^{\top}\mu_t \\ \varphi_{L_1}(x) &= \tilde{\rho}_1 \left\| \tilde{\Gamma}_1(x-\tilde{x}) \right\|_1 + \rho_1 \left\| \Gamma_1(x-x_t) \right\|_1 \\ \varphi_{L_2}(x) &= \frac{1}{2}\tilde{\rho}_2 \left\| \tilde{\Gamma}_2(x-\tilde{x}) \right\|_2^2 + \frac{1}{2}\rho_2 \left\| \Gamma_2(x-x_t) \right\|_2^2 \end{aligned}$$

The ADMM algorithm is implemented as follows:

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

Solving the optimization problem of the robo-advisor

1. General case:

$$\text{solved by ADMM} \longleftarrow \left\{ \begin{array}{ll} f(x) = \varphi_{\mathrm{MV}}(x) + \varphi_{L_2}(x) - \lambda \sum_{i=1}^{n} b_i \ln x_i & \longrightarrow \text{solved by CCD} \\ g(z) = \varphi_{L_1}(x) + \mathbf{1}_{\Omega}(x) & \longrightarrow \text{solved by PO + Dykstra} \end{array} \right.$$

or:

$$\text{solved by ADMM} \longleftarrow \left\{ \begin{array}{ll} f(x) = \varphi_{\mathrm{MV}}(x) + \varphi_{L_2}(x) & \longrightarrow \text{analytical solution} \\ g(z) = \varphi_{L_1}(x) + \mathbf{1}_{\Omega}(x) - \lambda \sum_{i=1}^{n} b_i \ln x_i & \longrightarrow \text{solved by ADMM + PO + Dykstra} \end{array} \right.$$

2. Ω is a set of linear constraints:

$$\text{solved by ADMM} \longleftarrow \left\{ \begin{array}{ll} f(x) = \varphi_{\mathrm{MV}}(x) + \varphi_{L_2}(x) + \mathbf{1}_{\Omega}(x) & \longrightarrow \text{solved by QP} \\ g(z) = \varphi_{L_1}(x) - \lambda \sum_{i=1}^{n} b_i \ln x_i & \longrightarrow \text{solved by CCD} \end{array} \right.$$

or:

$$\text{solved by ADMM} \longleftarrow \left\{ \begin{array}{ll} f(x) = \varphi_{\mathrm{MV}}(x) + \varphi_{L_2}(x) + \varphi_{L_1}(x) + \mathbf{1}_{\Omega}(x) & \longrightarrow \text{solved by Augmented QP} \\ g(z) = -\lambda \sum_{i=1}^{n} b_i \ln x_i & \longrightarrow \text{solved by PO} \end{array} \right.$$

3. No risk budgeting:

$$\text{solved by ADMM} \longleftarrow \left\{ \begin{array}{ll} f(x) = \varphi_{\mathrm{MV}}(x) + \varphi_{L_2}(x) + \mathbf{1}_{\Omega}(x) & \longrightarrow \text{solved by QP} \\ g(z) = \varphi_{L_1}(x) & \longrightarrow \text{solved by PO} \end{array} \right.$$

Calibrating the robo-advisor allocation problem

How to calibrate the hyperparameters in a systematic way?

- The covariance matrix Σ_t (e.g. the length of the window)
- The vector of expected returns μ_t (e.g. the length of the moving average, the importance of the views in a Black-Litterman framework, etc.)
- The L_1 shrinkage parameters (e.g. ϱ_1, $\tilde{\varrho}_1$, Γ_1 and $\tilde{\Gamma}_1$)
- The L_2 shrinkage parameters (e.g. ϱ_2, $\tilde{\varrho}_2$, Γ_2 and $\tilde{\Gamma}_2$)
- The turnover parameter
- The logarithmic barrier penalization λ
- Etc.

$\Rightarrow$ Global optimization

Calibrating the robo-advisor allocation problem

What can we do in the case of black-box functions?

1. Grid approach (combinatorial problem)
2. Stochastic optimization (Monte Carlo sampling)
3. Bayesian optimization (Močkus theory)

Gaussian process

Definition

A Gaussian process (GP) is a collection $\{f(x),\ x \in \mathscr{X}\}$ such that for any $n \in \mathbb{N}$ and $x_1, \ldots, x_n \in \mathscr{X}$, the random vector $(f(x_1), \ldots, f(x_n))$ has a joint multivariate Gaussian distribution which is characterized by its mean function:

$$m(x) = \mathbb{E}[f(x)]$$

and its covariance function:

$$\begin{aligned} \mathscr{K}(x, x') &= \operatorname{cov}(f(x), f(x')) \\ &= \mathbb{E}\left[(f(x) - m(x))(f(x') - m(x'))\right] \end{aligned}$$

Why modeling with GPs?

- The Gaussian process is a non-parametric and probabilistic model of a nonlinear function:
  - **Non-parametric** $\Rightarrow$ does not rely on any particular parametric functional form to be postulated
  - **Probabilistic** $\Rightarrow$ takes uncertainty into account in every aspect of the model
- Learn from few data
- Has attractive analytical properties
- Be a natural part of Bayesian framework, making modeling assumptions explicit
- Provide uncertainty quantification

Covariance functions

- The covariance function determines properties of GPs: regularity, lengthscale, periodicity, etc.
- **Usual covariance kernels**
  - SE kernel

$$\mathscr{K}_{\mathrm{SE}}\left(x,x'\right)=\sigma^{2}\exp\left(-\frac{1}{2}\left(x-x'\right)^{\top}\Sigma\left(x-x'\right)\right)$$

  - Brownian motion kernel

$$\mathscr{K}_{\mathrm{BM}}\left(x,x'\right)=\min\left(x,x'\right)$$

  - Linear kernel: $\mathscr{K}\left(x,x'\right)=x^{\top}x'$ (Bayesian linear regression)
  - Matern32, Rational Quadratic, Periodic, etc.
- **Kernel cooking**
  - Space-time mixing

$$\mathscr{K}\left(\left(x,t\right),\left(x',t'\right)\right)=\mathscr{K}_{\mathrm{Time}}\left(t,t'\right)\cdot\mathscr{K}_{\mathrm{Space}}\left(x,x'\right)$$

  - $\mathscr{K}_{\mathrm{SE}}+\mathscr{K}_{\mathrm{Linear}}\cdot\mathscr{K}_{\mathrm{Matern32}}$

Bayesian approach to machine learning

1. Formulate our knowledge about the situation probabilistically
2. Define a model that expresses qualitative aspects of our knowledge (eg, forms of distributions, independence assumptions). The model will have some unknown parameters
3. Specify a prior probability distribution for these unknown parameters that expresses our beliefs about which values are more or less likely, before seeing the data
4. Gather data
5. Compute the posterior probability distribution for the parameters, given the observed data. Use this posterior distribution to:
    - Reach scientific conclusions, properly accounting for uncertainty
    - Make predictions by averaging over the posterior distribution
    - Make decisions so as to minimize posterior expected loss

From Radford Neal.

Bayesian inference

- Place a prior probability distribution $p(\theta)$
- Choose a statistical model $p(x \mid \theta)$ that reflects our beliefs about x given θ
- Observe samples $X = (X_1, ..., X_n)$
- Update probability distribution $p(\theta \mid X_1, \ldots, X_n)$ with Bayes' theorem:

$$\text{Posterior} = \frac{\text{Likelihood} \times \text{Prior}}{\text{Marginal Likelihood}} \Leftrightarrow p(\theta \mid X) = \frac{p(X \mid \theta)\, p(\theta)}{p(X)}$$

Application to Gaussian processes

Gaussian process regression

The posterior of a GP is a GP. The posterior at points $x^\star$ is:

$$f(x^\star \mid x, y) \sim \mathcal{N}\left(m(x^\star \mid x, y), \mathcal{K}(x^\star, x^\star \mid x, y)\right)$$

where $m(x^\star \mid x, y)$ is the mean vector of the posterior distribution:

$$m(x^\star \mid x, y) = m(x^\star) + \mathcal{K}(x^\star, x)\,\mathcal{K}(x, x)^{-1}(y - m(x))$$

and the covariance matrix $\mathcal{K}(x^\star, x^\star \mid x, y)$ is the *Schur's complement* of the prior:

$$\mathcal{K}(x^\star, x^\star \mid x, y) = \mathcal{K}(x^\star, x^\star) - \mathcal{K}(x^\star, x)\,\mathcal{K}(x, x)^{-1}\,\mathcal{K}(x, x^\star)$$

$\Rightarrow$ The prediction is the conditional expectation:

$$\hat{y}^\star = m(x^\star \mid x, y)$$

Practice of Gaussian processes

We generally assume that:

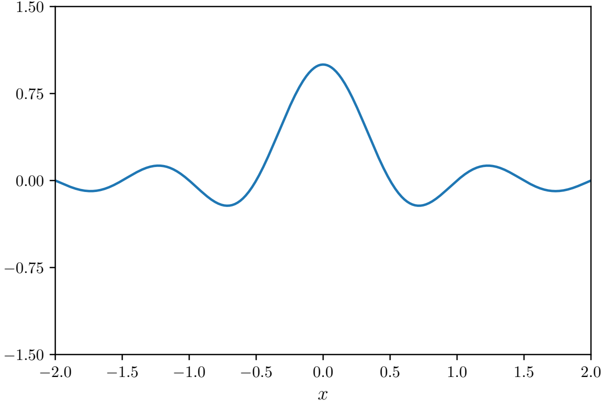
- $m(x) = \mathbf{0}_n$
- $m(x^\star) = \mathbf{0}_{n^\star}$

$\Rightarrow$ The conditional expectation reduces to:

$$m(x^\star \mid x, y) = \mathscr{K}(x^\star, x)\,\mathscr{K}(x, x)^{-1}\,y$$

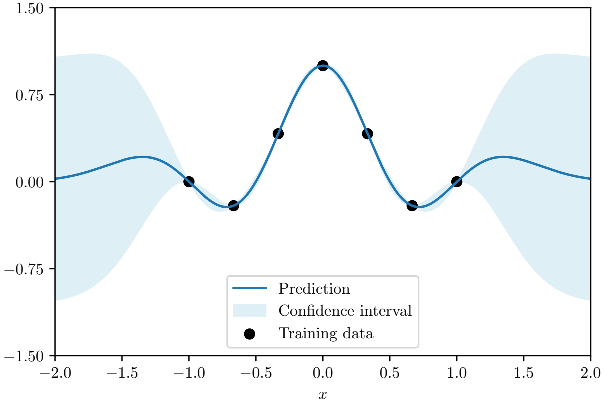
Bayesian inference of GP

Figure: Function $\operatorname{sinc}(2x)$



Bayesian inference of GP

Figure: Posterior distribution of the sample with the $\mathscr{K}_{\mathrm{SE}}$ kernel



Online learning

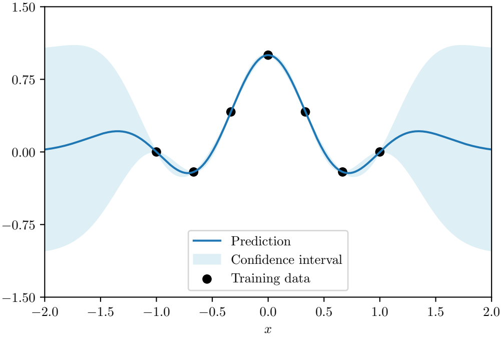


Figure: Prior distribution with the previous training data

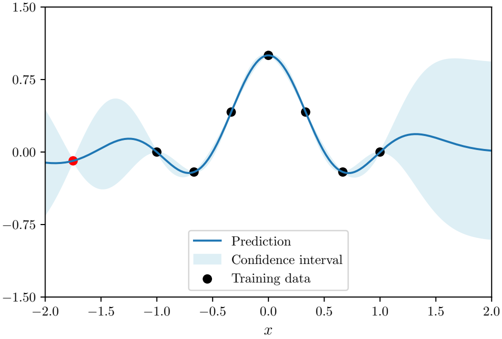
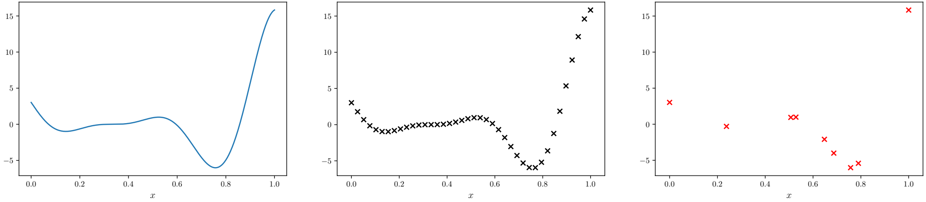


Figure: Posterior distribution with a new observation

General principle of Bayesian optimization

- **Bayesian optimization** = Method for the global optimization of multi-modal, computationally expensive black box functions
- **Goal**: optimize (minimize) a function $f(x)$ on some bounded set $\mathscr{X}$ such that:
  - We don't know the analytical expression of f
  - We don't have access to gradients
  - Computing $f(x)$ for a given x is expensive (time and/or money, for instance deep learning on AWS Servers)
  - The dimension problem is high (more than two hyperparameters) $\Rightarrow$ combinatorial issue

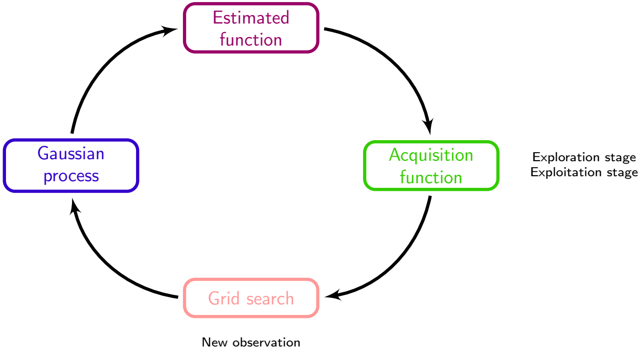


General principle of Bayesian optimization

The underlying idea of Bayesian optimization is to:

1. estimate the unknown objective function and,
2. build the optimal grid search

BO produces an iterative solution:



Probabilistic surrogate and acquisition function

- Bayesian optimization consists of two parts:
  - **Probabilistic surrogate**
    The approximation of the objective function is called a surrogate model[3]
  - **Acquisition function**
    Acquisition functions can be interpreted in the framework of Bayesian decision theory as evaluating an expected maximal gain associated with evaluating f at a point x

[3] Gaussian processes are a popular surrogate model for Bayesian optimization because the GP posterior is still a multivariate normal distribution

Acquisition function

$f(x)$ has a Gaussian process prior and we observe samples of the form $\{(x_i, y_i)\}_{i=1}^n$ where $y_i = f(x_i) + \varepsilon_i$, $\varepsilon_i \sim \mathcal{N}(0, \sigma_\varepsilon^2)$.
For a new observation $x^\star$, the posterior probability distribution is:

$$f(x^\star \mid x, y) \sim \mathcal{N}\left(\hat{m}_n(x^\star), \hat{\mathcal{K}}_n(x^\star, x^\star)\right)$$

We note $\mathcal{D}_n$ the augmented data with the GP:

$$\mathcal{D}_n = \left\{\left(x_i, y_i, \hat{f}_i(x_i)\right)\right\}_{i=1}^n$$

Let $\mathcal{U}_n(x^\star)$ be the **acquisition function** based on $\mathcal{D}_n$. The Bayesian optimization consists then in finding the new optimal point $x_{n+1} \in \mathcal{X}$ such that:

$$x_{n+1} = \arg\max \mathcal{U}_n(x^\star)$$

Improvement-based acquisition function

Improvement

Let $f_n(\varkappa_n^\star)$ be the current optimal value among n samples drawn from $f(x)$:

$$\varkappa_n^\star = \arg\max_{\varkappa \in x} f(\varkappa)$$

Let x_{n+1} be the next point to be evaluated in order to improve this value. We define the improvement $\Delta_n(x^\star)$ as follows:

$$\Delta_n(x^\star) = \max\left(\hat{f}_n(x^\star) - f_n(\varkappa_n^\star), 0\right)$$

$\Rightarrow$ Kushner (1964) proposes to maximize **the probability of a positive improvement**:

$$\Pr\{\Delta_n(x^\star) > 0\} = \Pr\left\{\hat{f}_n(x^\star) > f_n(\varkappa_n^\star)\right\} = \Phi\left(\frac{\hat{m}_n(x^\star) - f_n(\varkappa_n^\star)}{\sqrt{\hat{\mathcal{K}}_n(x^\star, x^\star)}}\right)$$

Improvement-based acquisition function

Močkus (1975) proposes to take into account the **expected value of improvement**:

$$\mathrm{EI}_n(x^\star) = \mathbb{E}[\Delta_n(x^\star)]$$

In the GP framework, we obtain:

$$\begin{aligned}\mathrm{EI}_n(x^\star) &= (\hat{m}_n(x^\star) - f_n(\varkappa_n^\star))\,\Phi\left(\frac{\hat{m}_n(x^\star) - f_n(\varkappa_n^\star)}{\sqrt{\hat{\mathscr{K}}_n(x^\star, x^\star)}}\right) + \\ &\quad \sqrt{\hat{\mathscr{K}}_n(x^\star, x^\star)}\,\phi\left(\frac{\hat{m}_n(x^\star) - f_n(\varkappa_n^\star)}{\sqrt{\hat{\mathscr{K}}_n(x^\star, x^\star)}}\right)\end{aligned}$$

Bayesian Optimization Algorithm

BO algorithm

We initialize the data sample $\mathscr{D}_1$ and the vector θ_1 of hyperparameters

for $n = 1, 2, \ldots$ **do**

Find the optimal value $x_{n+1} \in \mathscr{X}$ of the utility maximization problem:

$$x_{n+1} = \arg\max \mathscr{U}_n(x^\star)$$

Update the data:

$$\mathscr{D}_{n+1} \leftarrow \mathscr{D}_n \cup \left\{ \left(x_{n+1}, y_{n+1}, \hat{f}_{n+1}(x_{n+1}) \right) \right\}$$

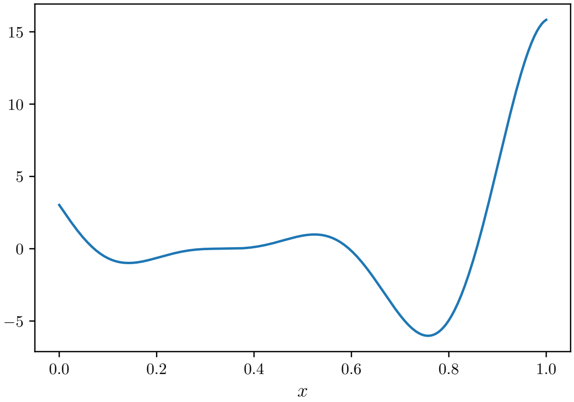
Update the hyperparameter vector θ_{n+1} of the kernel function

end for

return $\mathscr{D}_n$ and θ_n

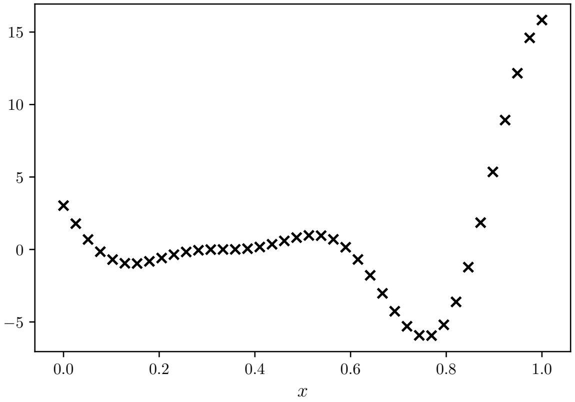
Example of Bayesian optimization

Figure: Objective function of the minimization problem

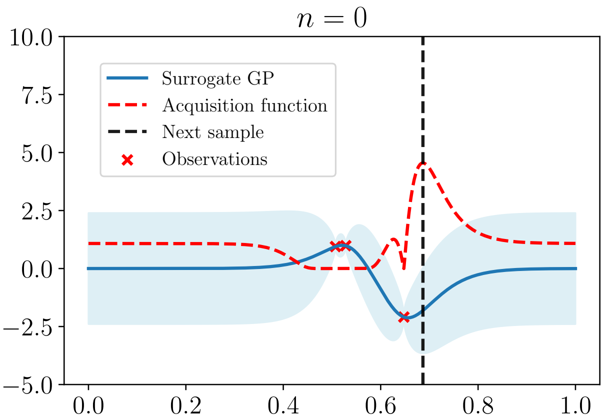


Example of Bayesian optimization

Figure: Grid search



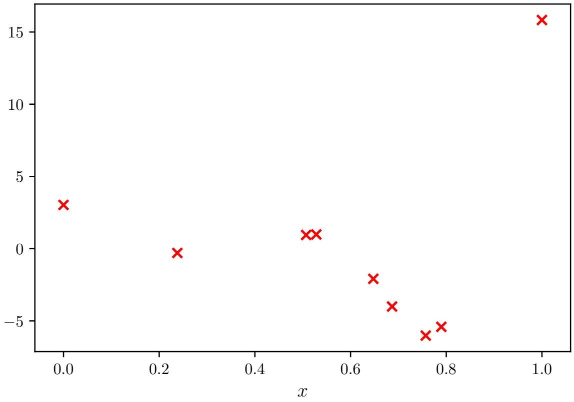
Example of Bayesian optimization



Go to Appendix

Example of Bayesian optimization

Figure: Bayesian grid



Building a self-automated trend-following strategy

We apply Bayesian optimization to the online calibration of the following trend-following strategy:

$$\begin{aligned} x_t^\star &= \arg\min_x -x^\top \hat{\mu}_t + \lambda_t \left\| x - x_{t-1} \right\|_2^2 \\ \text{s.t.} &\quad \sigma_t(x) \leq \bar{\sigma} \end{aligned}$$

where $\hat{\mu}_t$ is the estimated vector of expected returns at time t, $\sigma_t(x) = \sqrt{x^\top \hat{\Sigma}_t x}$ is the portfolio volatility estimated at time t and $\bar{\sigma}$ is the target volatility of the trend-following strategy

Building a self-automated trend-following strategy

Traditional CTA strategy

- λ_t is constant (e.g. 10%)
- Constant moving window length for estimating $\hat{\mu}_t$ (e.g. 3M or 12M)
- Constant moving window length for estimating $\hat{\Sigma}_t$ (e.g. 12M)

$\Rightarrow$ Two problems:

1. The optimization stage ($x_t^\star$), which is solved by using ADMM and the Cholesky trick
2. The calibration stage (λ_t, moving window length of $\hat{\mu}_t$ and $\hat{\Sigma}_t$), which is solved by using BO

Building a self-automated trend-following strategy

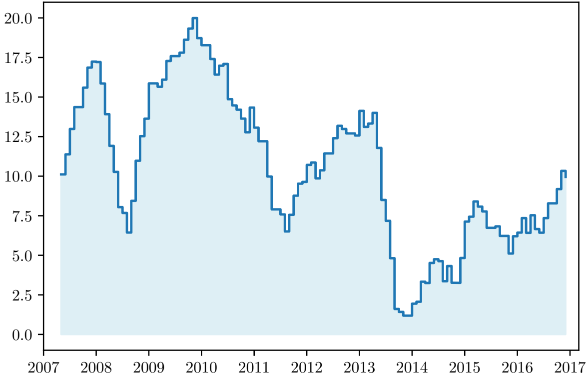
- Setup
  - The trends are computed using a moving-average estimator where $\ell_t(\mu)$ is the window length of the MA estimator
  - The covariance matrix is estimated using the empirical estimator, which window length is denoted by $\ell_t(\Sigma)$
  - The portfolio is rebalanced every week
- Hyperparameters
  - the parameter λ_t that controls the turnover between two rebalancing dates
  - the window length $\ell_t(\mu)$ that controls the estimation of trends
  - the time horizon $\ell_t(\Sigma)$ that measures the risk of the assets
- BO objective function: cumulative return over a 2-year backtest

Results of Bayesian optimization

- Most of the time, the optimal window $\ell_t(\mu)$ is high and equal to 18 months on average
- After the Global Financial Crisis of 2008, its value is dramatically reduced (short-term momentum preferred)
- Regularization hyperparameter λ_t and covariance window $\ell_t(\mu)$ show a positive correlation with VIX_t while the opposite is true for trend window

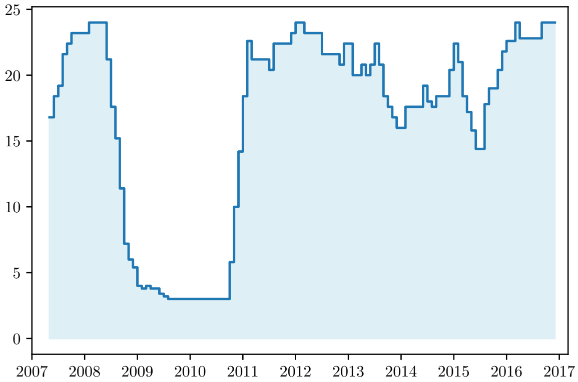
Results of Bayesian optimization

Figure: BO calibrated ridge penalization λ_t (in %)



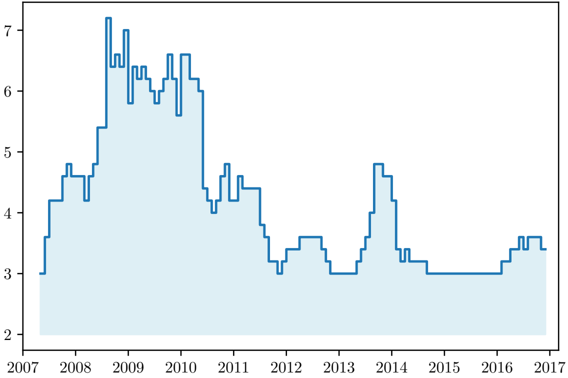
Results of Bayesian optimization

Figure: BO calibrated return window length $\ell_t(\mu)$ (in months)



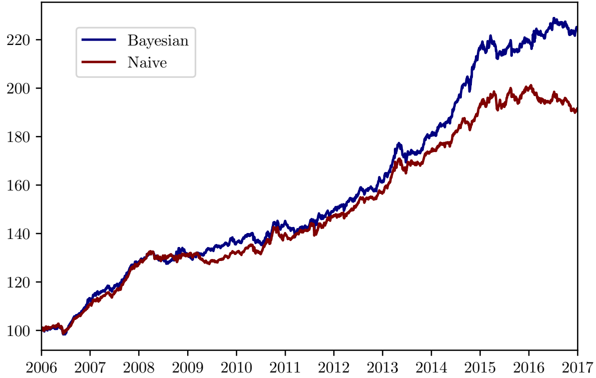
Results of Bayesian optimization

Figure: BO calibrated covariance window length $\ell_t(\Sigma)$ (in months)

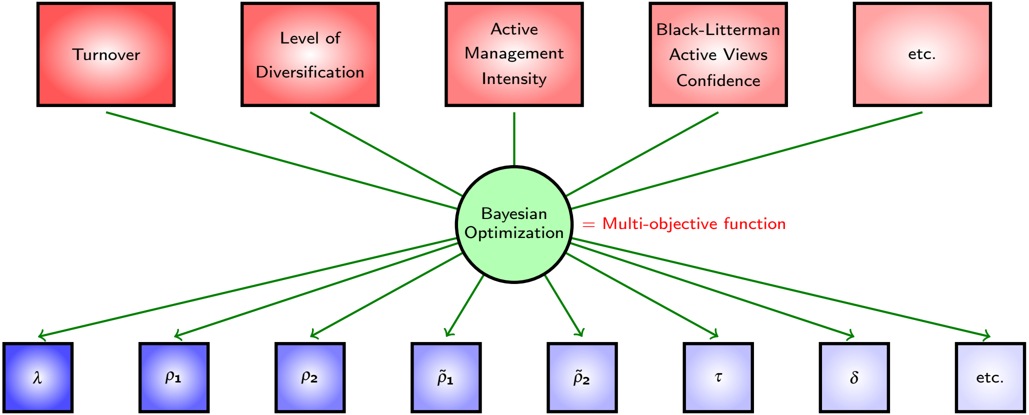


Results of Bayesian optimization

Figure: Cumulated performance



Applying Bayesian optimization to the robo-advisor calibration problem



- QP algorithm = universal algorithm in MVO-type asset allocation problems
- Robo-advisors require to solve more complex asset allocation optimization problems
- The optimization step can be achieved by considering numerical algorithms that have been successful in machine learning
  1. CCD
  2. ADMM
  3. Proximal operators
  4. Dykstra's algorithm
- The calibration step can be achieved by considering Bayesian optimization and Gaussian processes

The second step is the tricky part when building self-automated robo-advisors...

- Next step: learning...

References I

Bourgeron, T., Lezmi, E., and Roncalli, T. (2018)
Robust Asset Allocation for Robo-Advisors, *arXiv*, arxiv.org/abs/1902.07449.

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.

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, *arXiv*, arxiv.org/abs/1903.04841.

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.

References II

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

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. (2015)
Smart Beta: Managing Diversification of Minimum Variance Portfolios, in Jurczenko, E. (Ed.), *Risk-based and Factor Investing*, ISTE Press – Elsevier.

Richard, J-C., and Roncalli, T. (2019)
Constrained Risk Budgeting Portfolios: Theory, Algorithms, Applications & Puzzles, *arXiv*, `arxiv.org/abs/1902.05710`.

References III

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

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

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.

Precision matrix and hedging portfolios

We consider the following regression model:

$$R_{i,t} = \beta_0 + \beta_i^\top R_t^{(-i)} + \varepsilon_{i,t}$$

- $R_t^{(-i)}$ denotes the vector of asset returns R_t excluding the i^{th} asset
- $\varepsilon_{i,t} \sim \mathcal{N}\left(0, s_i^2\right)$
- $\mathscr{R}_i^2$ is the R-squared of the linear regression

Precision matrix

Stevens (1998) shows that the precision matrix is given by:

$$\mathscr{I}_{i,i} = \frac{1}{\hat{\sigma}_i^2 \left(1 - \mathscr{R}_i^2\right)} \text{ and } \mathscr{I}_{i,j} = -\frac{\hat{\beta}_{i,j}}{\hat{\sigma}_i^2 \left(1 - \mathscr{R}_i^2\right)} = -\frac{\hat{\beta}_{j,i}}{\hat{\sigma}_j^2 \left(1 - \mathscr{R}_j^2\right)}$$

Precision matrix and hedging portfolios

We finally obtain:

$$x_i^\star = \gamma \Sigma^{-1} \mu = \gamma \frac{\mu_i - \hat{\beta}_i^\top \mu^{(-i)}}{\hat{\sigma}_i^2 \left(1 - \mathscr{R}_i^2\right)} = \gamma \frac{\mu_i - \hat{\beta}_i^\top \mu^{(-i)}}{\hat{s}_i^2}$$

From this equation, we deduce the following conclusions:

1. The better the hedge, the higher the exposure. This is why highly correlated assets produces unstable MVO portfolios.
2. The long-short position is defined by the sign of $\mu_i - \hat{\beta}_i^\top \mu^{(-i)}$. If the expected return of the asset is lower than the conditional expected return of the hedging portfolio, the weight is negative.

Markowitz diversification	$\neq$	**Diversification of risk factors**
	$=$	**Concentration on arbitrage factors**

Precision matrix and hedging portfolios

Table: Hedging portfolios (in %) at the end of 2006

	SPX	SX5E	TPX	RTY	EM	US HY	EMBI	EUR	JPY	GSCI
SPX		58.6	6.0	150.3	-30.8	-0.5	5.0	-7.3	15.3	-25.5
SX5E	9.0		-1.2	-1.3	35.2	0.8	3.2	-4.5	-5.0	-1.5
TPX	0.4	-0.6		-2.4	38.1	1.1	-3.5	-4.9	-0.8	-0.3
RTY	48.6	-2.7	-10.4		26.2	-0.6	1.9	0.2	-6.4	5.6
EM	-4.1	30.9	69.2	10.9		0.9	4.6	9.1	3.9	33.1
US HY	-5.0	53.5	160.0	-18.8	69.5		95.6	48.4	31.4	-211.7
EMBI	10.8	44.2	-102.1	12.3	73.4	19.4		-5.8	40.5	86.2
EUR	-3.6	-14.7	-33.4	0.3	33.8	2.3	-1.4		56.7	48.2
JPY	6.8	-14.5	-4.8	-8.8	12.7	1.3	8.4	50.4		-33.2
GSCI	-1.1	-0.4	-0.2	0.8	10.7	-0.9	1.8	4.2	-3.3	
$\hat{s}_i$	0.3	0.7	0.9	0.5	0.7	0.1	0.2	0.4	0.4	1.2
$\mathscr{R}_i^2$	83.0	47.7	34.9	82.4	60.9	39.8	51.6	42.3	43.7	12.1

Source: Bruder *et al.* (2013)

The mechanism of Markowitz optimization

We consider two portfolios:

1. The first portfolio $y^\star$ is the optimal portfolio by assuming zero correlation:

$$y_i^\star = \gamma \frac{\mu_i}{\sigma_i^2}$$

2. The second portfolio $z^\star$ is the optimal portfolio of the hedging strategies:

$$z_i^\star = \gamma \frac{\hat{\beta}_i^\top \mu^{(-i)}}{\sigma_i^2 - s_i^2}$$

The Markowitz solution is then:

$$x_i^\star = y_i^\star + \omega_i \left(y_i^\star - z_i^\star \right)$$

where:

$$\omega_i = \frac{\mathfrak{R}_i^2}{1 - \mathfrak{R}_i^2} = \frac{\sigma_i^2 - s_i^2}{s_i^2}$$

Arbitrage factors and hedging portfolios

Example

We consider a universe of four assets. The expected returns are $\hat{\mu}_1 = 7\%$, $\hat{\mu}_2 = 8\%$, $\hat{\mu}_3 = 9\%$ and $\hat{\mu}_4 = 10\%$ whereas the volatilities are equal to $\hat{\sigma}_1 = 15\%$, $\hat{\sigma}_2 = 18\%$, $\hat{\sigma}_3 = 20\%$ and $\hat{\sigma}_4 = 25\%$. All the correlations are equal to 50%, except $\rho_{1,4} = 60\%$ and $\rho_{3,4} = 40\%$.

Table: Linear dependence between the four assets (hedging portfolios)

Asset	α_i	β_i				$\mathfrak{R}_i^2$
1	1.70%		0.139	0.187	0.250	45.83%
2	2.06%	0.230		0.268	0.191	37.77%
3	2.85%	0.409	0.354		0.045	33.52%
4	1.41%	0.750	0.347	0.063		41.50%

Arbitrage factors and hedging portfolios

Table: Risk/return analysis of hedging portfolios

Asset	μ_i	$\hat{\mu}_i$	α_i	σ_i	$\hat{\sigma}_i$	s_i	$\mathfrak{R}_i^2$
1	7.00%	5.30%	1.70%	15.00%	10.16%	11.04%	45.83%
2	8.00%	5.94%	2.06%	18.00%	11.06%	14.20%	37.77%
3	9.00%	6.15%	2.85%	20.00%	11.58%	16.31%	33.52%
4	10.00%	8.59%	1.41%	25.00%	16.11%	19.12%	41.50%

Table: Optimal portfolio

Asset	ω_i	$y_i^\star$	$z_i^\star$	$x_i^\star$
1	84.62%	80.22%	132.48%	36.00%
2	60.68%	63.67%	125.09%	26.39%
3	50.43%	58.02%	118.19%	27.67%
4	70.94%	41.26%	85.40%	9.94%

Arbitrage factors and hedging portfolios

Impact of the correlation

$$\rho_{3,4} = 40\% \Longrightarrow \rho_{3,4} = 95\%$$

Table: Linear dependence between the four assets (hedging portfolios)

Asset	α_i	β_i				$\mathfrak{R}_i^2$
1	3.16%		0.244	−0.595	0.724	47.41%
2	2.23%	0.443		0.470	−0.157	33.70%
3	1.66%	−0.174	0.076		0.795	91.34%
4	−1.61%	0.292	−0.035	1.094		92.37%

Arbitrage factors and hedging portfolios

Table: Risk/return analysis of hedging portfolios ($\rho_{3,4} = 95\%$)

Asset	μ_i	$\hat{\mu}_i$	α_i	σ_i	$\hat{\sigma}_i$	s_i	$\mathfrak{R}_i^2$
1	7.00%	3.84%	3.16%	15.00%	10.33%	10.88%	47.41%
2	8.00%	5.77%	2.23%	18.00%	10.45%	14.66%	33.70%
3	9.00%	7.34%	1.66%	20.00%	19.11%	5.89%	91.34%
4	10.00%	11.61%	−1.61%	25.00%	24.03%	6.90%	92.37%

Table: Optimal portfolio ($\rho_{3,4} = 95\%$)

Asset	ω_i	$y_i^\star$	$z_i^\star$	$x_i^\star$
1	90.16%	60.73%	70.30%	52.10%
2	50.82%	48.20%	103.08%	20.31%
3	1054.10%	43.92%	39.22%	93.44%
4	1211.48%	31.23%	39.25%	−65.85%

Arbitrage factors and hedging portfolios

Impact of the expected return

$$\mu_1 = 7\% \Longrightarrow \mu_1 = 3\%$$

Table: Linear dependence between the four assets (hedging portfolios)

Asset	α_i	β_i				$\mathfrak{R}_i^2$
1	-2.30%		0.139	0.187	0.250	45.83%
2	2.98%	0.230		0.268	0.191	37.77%
3	4.49%	0.409	0.354		0.045	33.52%
4	4.41%	0.750	0.347	0.063		41.50%

Arbitrage factors and hedging portfolios

Table: Risk/return analysis of hedging portfolios ($\mu_1 = 3\%$)

Asset	μ_i	$\hat{\mu}_i$	α_i	σ_i	$\hat{\sigma}_i$	s_i	$\mathfrak{R}_i^2$
1	3.00%	5.30%	−2.30%	15.00%	10.16%	11.04%	45.83%
2	8.00%	5.02%	2.98%	18.00%	11.06%	14.20%	37.77%
3	9.00%	4.51%	4.49%	20.00%	11.58%	16.31%	33.52%
4	10.00%	5.59%	4.41%	25.00%	16.11%	19.12%	41.50%

Table: Optimal portfolio ($\mu_1 = 3\%$)

Asset	ω_i	$y_i^\star$	$z_i^\star$	$x_i^\star$
1	84.62%	53.59%	206.52%	−75.81%
2	60.68%	99.25%	164.80%	59.46%
3	50.43%	90.44%	135.19%	67.87%
4	70.94%	64.31%	86.63%	48.48%

Analytical framework of adding constraints

We specify the optimization problem as follows:

$$\min \frac{1}{2} x^\top \Sigma x \quad \text{s.t.} \left\{ \begin{array}{l} \mathbf{1}^\top x = 1 \\ \mu^\top x \geq \mu^\star \\ x \in \mathscr{C} \end{array} \right.$$

where $\mathscr{C}$ is the set of weights constraints. We define:

- the unconstrained portfolio $x^\star$ or $x^\star(\mu, \Sigma)$:

$$\mathscr{C} = \mathbb{R}^n$$

- the constrained portfolio $\tilde{x}$:

$$\mathscr{C}(x^-, x^+) = \left\{ x \in \mathbb{R}^n : x_i^- \leq x_i \leq x_i^+ \right\}$$

Analytical framework of adding constraints

Theorem

Jagannathan and Ma (2003) show that the constrained portfolio is the solution of the unconstrained problem:

$$\tilde{x} = x^{\star}\left(\tilde{\mu}, \tilde{\Sigma}\right)$$

with:

$$\left\{ \begin{array}{l} \tilde{\mu} = \mu \\ \tilde{\Sigma} = \Sigma + \left(\lambda^{+} - \lambda^{-}\right)\mathbf{1}^{\top} + \mathbf{1}\left(\lambda^{+} - \lambda^{-}\right)^{\top} \end{array} \right.$$

where λ^{-} and λ^{+} are the Lagrange coefficients vectors associated to the lower and upper bounds.

$\Rightarrow$ Introducing weights constraints is equivalent to introduce a shrinkage method or to introduce some relative views (similar to the **Black-Litterman** approach).

Application to the minimum variance portfolio

Table: Specification of the covariance matrix Σ (in %)

σ_i	$\rho_{i,j}$			
15.00	100.00			
20.00	10.00	100.00		
25.00	40.00	70.00	100.00	
30.00	50.00	40.00	80.00	100.00

Given these parameters, the global minimum variance portfolio is equal to:

$$x^{\star} = \begin{pmatrix} 72.74\% \\ 49.46\% \\ -20.45\% \\ -1.75\% \end{pmatrix}$$

Application to the minimum variance portfolio

Table: Minimum variance portfolio when $x_i \geq 10\%$

$\tilde{x}_i$	λ_i^-	λ_i^+	$\tilde{\sigma}_i$	$\tilde{\rho}_{i,j}$			
56.195	0.000	0.000	15.00	100.00			
23.805	0.000	0.000	20.00	10.00	100.00		
10.000	1.190	0.000	19.67	10.50	58.71	100.00	
10.000	1.625	0.000	23.98	17.38	16.16	67.52	100.00

Table: Minimum variance portfolio when $10\% \leq x_i \leq 40\%$

$\tilde{x}_i$	λ_i^-	λ_i^+	$\tilde{\sigma}_i$	$\tilde{\rho}_{i,j}$			
40.000	0.000	0.915	20.20	100.00			
40.000	0.000	0.000	20.00	30.08	100.00		
10.000	0.915	0.000	21.02	35.32	61.48	100.00	
10.000	1.050	0.000	26.27	39.86	25.70	73.06	100.00

Extension to constrained MVO portfolios

Remark

Roncalli (2013) extends the previous result when $\mathscr{C} = \{x \in \mathbb{R}^n : Cx \geq d\}$. *The covariance matrix is shrinked as follows:*

$$\tilde{\Sigma} = \Sigma - \left(C^\top \lambda \mathbf{1}^\top + \mathbf{1} \lambda^\top C \right)$$

where λ *is the vector of Lagrange coefficients associated to the constraints* $Cx \geq d$.

Myopic behavior of portfolio managers

Weight constraints → Shrinkage methods

By using weight constraints, the portfolio manager may change (implicitly):

1. the value and/or the ordering of the volatilities;
2. the value, the sign and/or the ordering of the correlations;
3. the underlying assumption of the theory itself.

The question is then the following:

Is the portfolio manager aware of and in agreement with these changes?

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)

Portfolio optimization with a benchmark

The penalization approach of the QP problem

- Markowitz optimization:

$$x^{\star} = \arg\min \frac{1}{2} x^{\top} \Sigma x - \gamma x^{\top} \mu$$

- Markowitz optimization with a benchmark:

$$\begin{aligned} x^{\star}(b) &= \arg\min \frac{1}{2} x^{\top} \Sigma x - \xi x^{\top} \left(\frac{\mu + \mu_b}{2} \right) \\ \text{s.t.} &\quad x \in \Omega \end{aligned}$$

where $\xi = 2\gamma$ and μ_b is the vector of ~~Black-Litterman~~ implied expected returns[a].

[a] If the benchmark b is the optimal portfolio, we have $b = \gamma \Sigma^{-1} \mu_b$

Portfolio optimization with views

Black and Litterman (1992) state that vector R_t of asset returns follow a Gaussian distribution:

$$R_t \sim \mathcal{N}\left(\tilde{\mu}, \Sigma_m\right)$$

where:

- $\tilde{\mu}$ is the implied expected return associated with the current allocation x_0:

$$\tilde{\mu} = r + \mathrm{SR}\left(x_0 \mid r\right) \frac{\Sigma_m x_0}{\sqrt{x_0^\top \Sigma_m x_0}}$$

- Σ_m is the market covariance matrix of asset returns

The portfolio manager's views are given by:

$$PR_t = Q + \varepsilon$$

where P is a $(k \times n)$ matrix, Q is a $(k \times 1)$ vector and $\varepsilon \sim \mathcal{N}\left(0, \Sigma_\varepsilon\right)$ is a Gaussian vector of dimension k.

A new QP problem

We deduce that:

$$\begin{aligned}\bar{\mu} &= \mathbb{E}\left[R_t \mid PR_t = Q + \varepsilon\right] \\ &= \tilde{\mu} + \Sigma_m P^\top \left(P\Sigma_m P^\top + \Sigma_\varepsilon\right)^{-1} \left(Q - P\tilde{\mu}\right)\end{aligned}$$

and:

$$\begin{aligned}\bar{\Sigma} &= \mathbb{E}\left[\left(R_t - \bar{\mu}\right)\left(R_t - \bar{\mu}\right)^\top \mid PR_t = Q + \varepsilon\right] \\ &= \Sigma_m - \Sigma_m P^\top \left(P\Sigma_m P^\top + \Sigma_\varepsilon\right)^{-1} P\Sigma_m\end{aligned}$$

The case of absolute views

If $P = I_n$ and $Q = \breve{\mu}$, we deduce that:

$$\bar{\mu} = \left(I_n - \Sigma_m \left(\Sigma_m + \Sigma_\varepsilon\right)^{-1}\right)\tilde{\mu} + \Sigma_m \left(\Sigma_m + \Sigma_\varepsilon\right)^{-1} \breve{\mu}$$

The regularized QP problem of the Black-Litterman model

Let $\hat{\Sigma}$ be the empirical covariance matrix.

1. If we assume that $\Sigma_m = \tau\hat{\Sigma}$ and $\Sigma_\varepsilon = \tau\hat{\Sigma}$, we obtain:

$$\bar{\mu} = \frac{\tilde{\mu} + \breve{\mu}}{2}$$

2. If we assume that $\Sigma_m = \hat{\Sigma}$ and $\Sigma_\varepsilon = \tau\hat{\Sigma}$, we obtain:

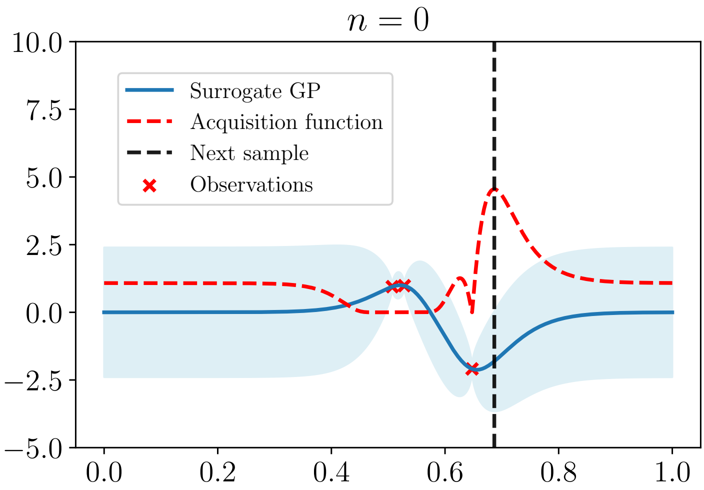
$$\bar{\mu} = \frac{\tau}{1+\tau}\tilde{\mu} + \frac{1}{1+\tau}\breve{\mu}$$

Black-Litterman and regularization

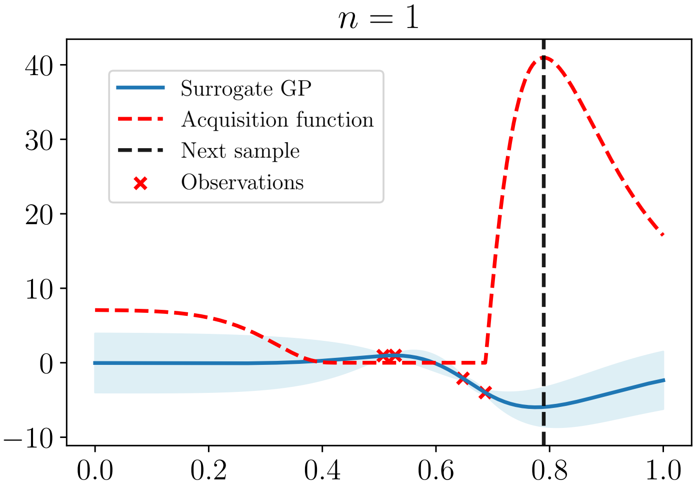
The Black-Litterman model with absolute views is a special case of the tracking-error optimization problem where:

- the current allocation x_0 is the benchmark;
- the uncertainty on the views and the covariance matrix of asset returns are in the same order of magnitude (τ is equal to one).

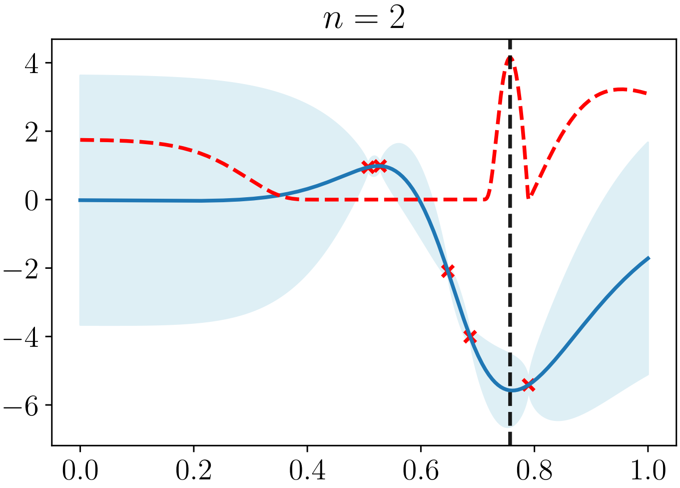
Example of Bayesian optimization



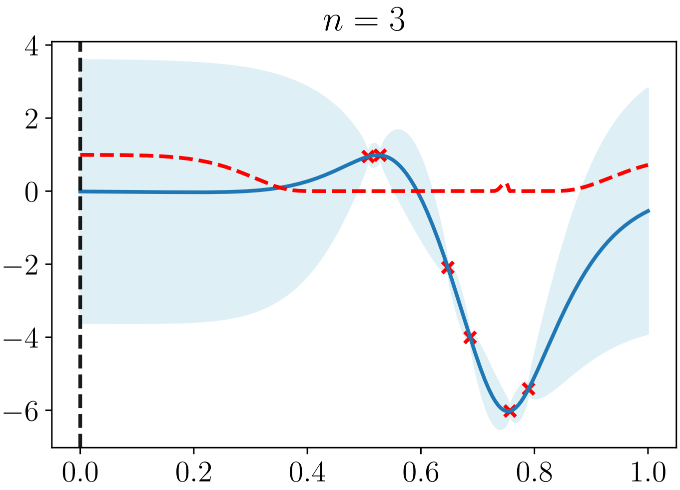
Example of Bayesian optimization



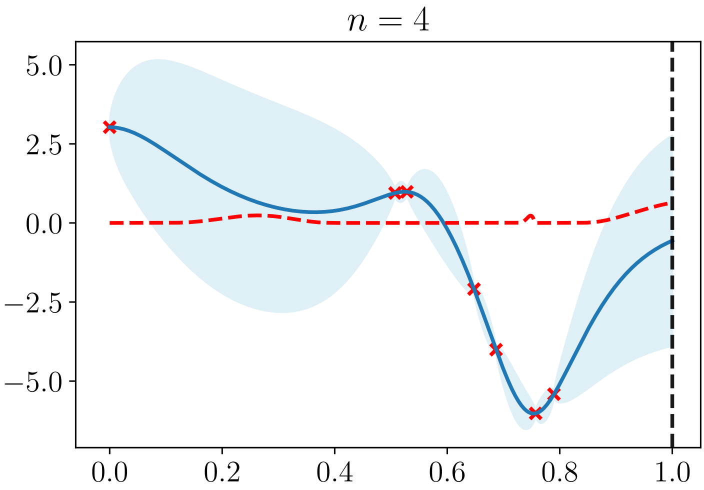
Example of Bayesian optimization



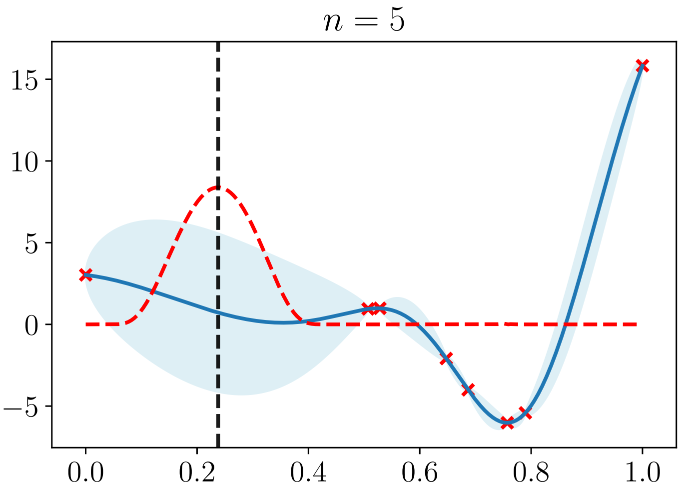
Example of Bayesian optimization



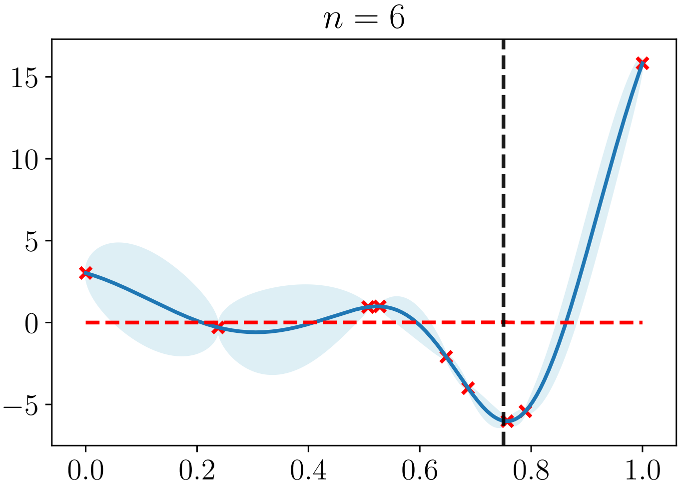
Example of Bayesian optimization



Example of Bayesian optimization



Example of Bayesian optimization



Example of Bayesian optimization

Figure: Objective function of the minimization problem

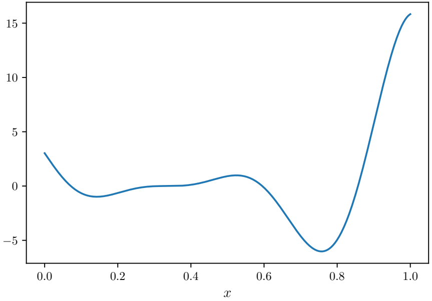
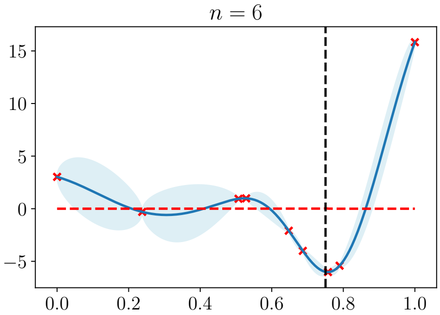


Figure: Solution of the Bayesian optimization



Back to BO

Disclaimer