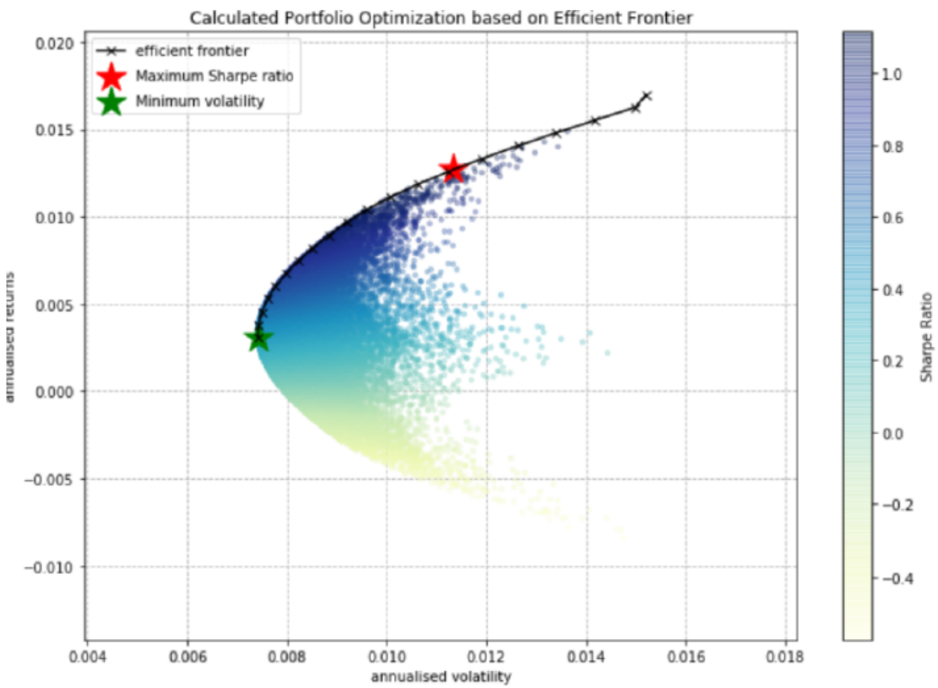
Optimisation de Portefeuille de Markowitz

Irina Kortchemski, CYTECH



Problèmes d'optimisation en finance

- La finance est toujours une optimisation ou un pricing!
  L'optimisation permet de :
  - maximiser les profits
  - minimiser les coûts
  - contrôler les risques.
- Les problèmes d'optimisation auxquels sont confrontés les professionnels de la finance peuvent inclure :
  - Allouer de manière optimale les actifs dans un portefeuille.
  - Construire une courbe de taux d'intérêt basée sur des observations de taux de référence.
  - Adapter un modèle de volatilité implicite aux données du marché des options.
  - Dénouer une position importante de manière à minimiser les coûts de transaction.
  - Concevoir une stratégie de couverture optimale d'un portefeuille d'actifs.

Objectifs du projet

Optimisation de portefeuille de Markowitz

- Théorie d'optimisation avec contraintes
- Conditions de Kuhn-Tucker
- Solution analytique pour le portefeuille de Markowitz
- Algorithmes d'optimisation avec contraintes
  - • Méthode de descente à pas optimal
  - • Méthode de Lagrange - Newton
  - • Algorithmes DUALS
  - ◦ Algorithme de Uzawa
  - ◦ Variantes des Algorithmes de Uzawa (Articles de recherche)
- Nous allons examiner un Toy - Model (un modèle en 3 dim) ensemble!

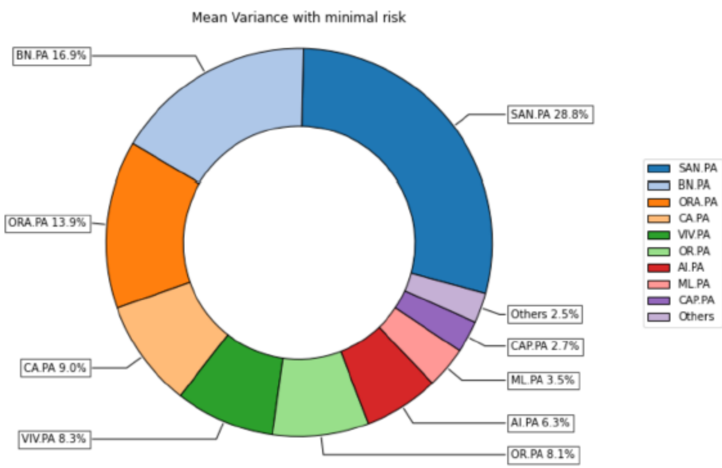
Objectifs du projet

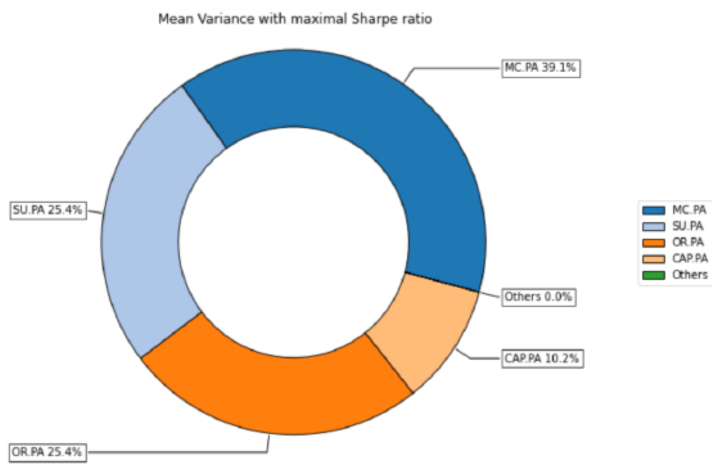
- Application aux bases des données de Bloomberg. Construction des Portfeuilles efficiens et optimaux.
  - Vous programmez tout les algoritmes du zero
  - Vous apprendrez à utilisez les Solvers de Python et de MatLab
- Fonctions d'optimisation de Python
- Calcul de VAR des Portfeuilles
  - VaR historique
  - VaR analytique
  - VaR via Monté Carlo

Portefeuilles optimaux

- Image (CaC 40 + allocation)

	A	B	C	D	E	F	G	H	I
1	Date,AC.PA,ACA.PA,AI.PA,AIR.PA,ATO.PA,BN.PA,BNP.PA,CA.PA,CAP.PA,DG.PA,EN.PA,ENGI.PA,GLE.PA,HO.PA,KER.PA,LR.PA,MC.PA,								
2	2019-01-02,35.53131103515625,8.188713073730469,90.3832778930664,80.3687515258789,52.4089241027832,53.6742134094								
3	2019-01-03,34.948829650878906,8.162334442138672,88.72212982177734,77.54242706298828,49.634422302246094,53.71817								
4	2019-01-04,35.764305114746094,8.585267066955566,91.19255828857422,81.27584838867188,51.45975875854492,54.500782								
5	2019-01-07,35.91963195800781,8.571199417114258,90.68142700195312,79.78630065917969,50.919456481933594,53.498348								
6	2019-01-08,36.00700378417969,8.626593589782715,90.80921173095703,82.73675537109375,52.438133239746094,53.920425								
7	2019-01-09,36.44386672973633,8.616042137145996,89.57400512695312,85.6299057006836,53.65015411376953,54.13146209								
8	2019-01-10,36.550655364990234,8.689022064208984,88.59435272216797,85.56307220458984,54.45330047607422,54.307323								
9	2019-01-11,36.725399017333984,8.692538261413574,87.95545196533203,84.38861846923828,53.63555145263672,54.588718								
10	2019-01-14,36.48270034790039,8.755846977233887,87.23136901855469,85.84952545166016,50.61280059814453,54.2809486								
11	2019-01-15,36.39532470703125,8.754088401794434,88.21100616455078,88.1411361694336,50.802635192871094,54.7469902								
12	2019-01-16,36.76422882080078,9.097006797790527,88.5517578125,88.36075592041016,51.532772064208984,54.0523300170								
13	2019-01-17,36.66714859008789,8.928185462951660,88.5517578125,87.61598205566406,51.284523010253906,54.43043518066								
14	2019-01-18,37.88064956665039,9.077662467956543,89.91474914550781,89.43017578125,53.781578063964844,54.826129913								
15	2019-01-21,37.66707229614258,8.99325180053711,89.57400512695312,90.23223876953125,53.00763702392578,55.34493637								
16	2019-01-22,37.81269454956055,8.854326248168945,89.36103057861328,90.47093963623047,53.139060974121094,55.230625								
17	2019-01-23,37.56028366088867,8.887738227844238,89.48881530761719,89.57339477539062,53.18286895751953,55.1163101								
18	2019-01-24,37.443790435791016,8.908841133117676,89.70177459716797,91.58809661865234,55.679927825927734,55.22183								
19	2019-01-25,37.39524841308594,9.125144004821777,89.61659240722656,92.1514663696289,56.366249084472656,54.9140663								





Théorie d'optimisation

- L'optimisation fait partie des mathématiques appliquées. Cela signifie que les problèmes d'optimisation sont formulés en termes de propriétés de fonctions convenablement définies.
- Un nombre très limité de problèmes d'optimisation peut être résolu en "forme fermée", c'est-à-dire une formule explicite prête à être utilisée. En général, la résolution d'un problème d'optimisation nécessite la conception d'un algorithme et son implémentation dans un code informatique.
- Tous les algorithmes ne sont pas égaux : nous préférons fortement les algorithmes présentant de bonnes caractéristiques de performance.

Portefeuille de Markowitz

- Développée par l'économiste américain Harry Markowitz (l'un des Prix Nobel d'économie 1990), la théorie du portefeuille efficient définit le mécanisme de sélection de titres constituant un portefeuille boursier idéal sur le papier.
- Le modèle de Markowitz, permet donc, à partir d'une série de portefeuilles, de répondre à la question : Comment choisir le meilleur portefeuille, pour un niveau de risque donné ? Autrement dit
  • Comment maximiser mon rendement, pour un niveau de risque donné ?
  • Comment minimiser mon risque, pour un niveau de rendement souhaité.

Deux étapes d'optimisation

- Lorsqu'un investisseur recherche le meilleur portefeuille en termes de retour sur risque parmi la variété de portefeuilles possibles, deux étapes doivent être réalisées.
- La première consiste à déterminer l'ensemble des portefeuilles efficients.
  • On etudie la problème d'optimisation uniquement avec les actifs risqués $S_1, S_2, ...S_N$.
- La seconde consiste à sélectionner le portefeuille final spécifique à partir de l'ensemble efficient, compte tenu du rendement cible de l'investisseur, de son risque cible ou de sa préférence pour un ratio rendement/risque optimal.
  • On inclu dans le problème d'optimisation un actis sans risque S_0.

Portefeuille de Markowitz

- On possède N actifs risquées représentés par des variables aléatoires $S_1, S_2, ...S_N$.
- Vecteur X_i est la proportion de la somme investie dans l'actif i donc $\sum_{i=1}^{N} X_i = 1$
- Le portefeuille total est représenté par la variable aléatoire $P(x) = \sum_{i=1}^{N} X_i S_i$
- Chaque action rapporte en moyenne $E_i = \mathbb{E}[S_i]$ (espérance de S_i)
- Le rendement

$$R_0 = \sum_{i=1}^{N} X_i E_i$$

Portefeuille de Markowitz

- Le risque du portefeuille est mesuré par sa variance:

$\sigma^2(X_1, ...X_N) = \mathbb{E}[(P - \mathbb{E}[P])^2]$

- Matrice de covariance

$\Gamma_{ij} = \mathbb{E}[(S_i - \mathbb{E}[S_i])(S_j - \mathbb{E}[S_j])]$

- Variance du portefeuille s'écrit de la forme:

$\sigma^2(X_1, ...X_N) = \sum_{ij}^N \Gamma_{ij} X_i X_j$

Cherchons l'investissement pour minimiser le risque et assurer un bon rendement,

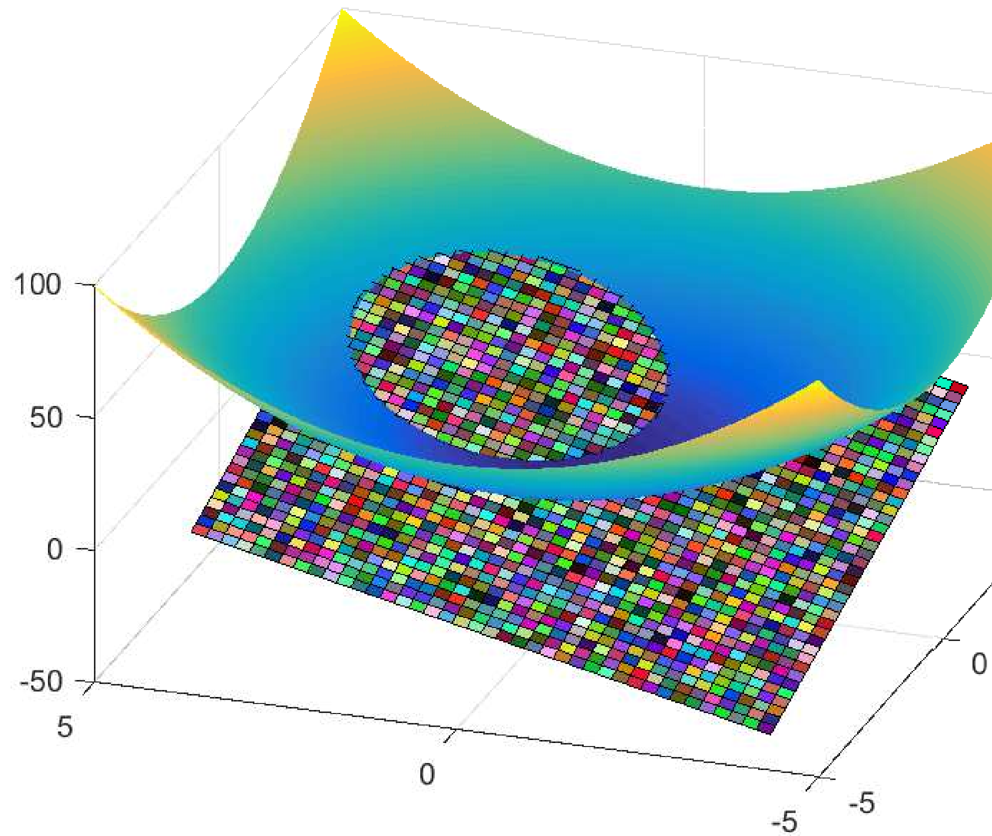
$$\begin{cases} \min_{X_1,...X_N} \sum_{ij}^N \Gamma_{ij} X_i X_j \\ \sum_{i=1}^n X_i E_i = R_0 \\ \sum_{i=1}^N X_i = 1 \end{cases}$$

Optimisation du Portefeuille

- Minimisation le risque sous une seulle contrainte egalité:

$$\begin{cases} \min_{X_1,\dots X_n} & \sum_{ij}^{n} \Gamma_{ij} X_i X_j \\ & \sum_{i=1}^{n} X_i = 1 \end{cases}$$

• Contrainte $X_i \geq 0$ on impose si le potrefeuille est "long". C'est à dire on interdie la vente à la decoverte



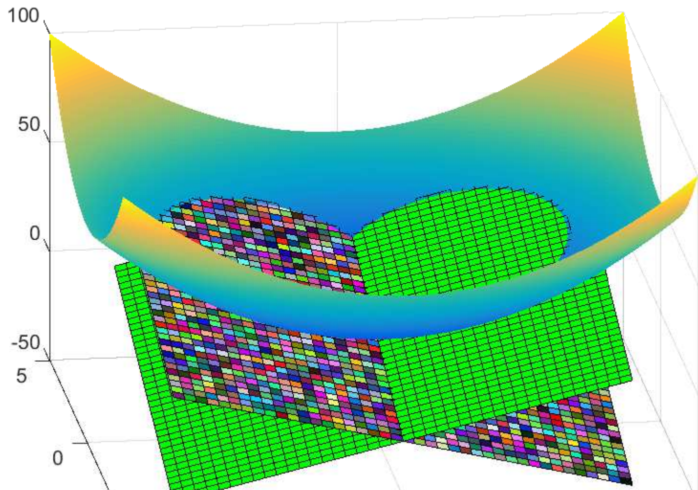
Optimisation du Portefeuille

- Minimisation le risque: sous les contraintes egalités:

$$\begin{cases} \min_{X_1,\dots X_n} \quad \sum_{ij}^n \Gamma_{ij} X_i X_j \\ \sum_{i=1}^n X_i E_i \geq R_0 \\ \sum_{i=1}^n X_i = 1 \end{cases}$$

- Maximiser le return sous les contraintes egalités:

$$\begin{cases} \max_{X_1,\dots X_n} \sum_{i=1}^n X_i E_i \\ \sum_{ij}^n \Gamma_{ij} X_i X_j = \sigma_0^2 \\ \sum_{i=1}^n X_i = 1 \end{cases}$$



Optimisation du Portefeuille

- Minimisation le risque sous deux contraintes:

$$\begin{cases} \min_{X_1,\ldots X_N} \quad \sum_{ij}^{N} \Gamma_{ij} X_i X_j \\ \sum_{i=1}^{N} X_i = 1 \\ X_i \geq 0 \end{cases}$$

- Le problème sans contrainte $X_i \geq 0, \quad i = 1, \ldots N$ signifie qu'on permet les ventes des titres à la decouverte. C'est à dire on permet "short selling", la ventes des actifs qu'on ne possède pas.
  • Le problème se résout numériquement à l'aide de l'algorithme de Lagrange-Newton
- Le problème avec contrainte $X_i \geq 0$ est plus difficile.
- On vous invite à etudier des articles de recherche.
  • FLORIN SPINU " An algorithm for computing risk parity weights" (2013)

Optimisation avec Sharpe Ratio

- Portefeuille contient un actif sans risque avec le taux d'interet r_f.
- Maximisation du Sharpe Ratio sous contraintes egalités:

$$\begin{cases} \max_{X_1,\ldots X_N} \ \frac{\sum X_i E_i - r_f}{\sqrt{\sum_{ij} \Gamma_{ij} X_i X_j}} \\ \sum_{i=1}^N X_i = 1 \end{cases}$$

- Problème equivalente. Il y a avec une partie X_0 investie dans l'actif sans risque

$$\begin{cases} \min \quad \sum_{ij}^N \Gamma_{ij} X_i X_j \\ \sum_{i=1}^n X_i E_i + r_f \cdot X_0 \geq R_0 \\ \sum_{i=1}^N X_i + X_0 = 1 \end{cases}$$

Rendement d'un actif

- N est le nombre des actifs dans le portefeuille, M est le nombre des intervalles temporelles, N_t le nombre des jours de trading.
- Rendement sur l'intervalle $[t_i, t_{i+1}]$ d'un actif $S(t_i, j)$ du type j

$$R(i,j) = \ln(\frac{S(t_{i+1}, j)}{S(t_i, j)})$$

- Rendement E_j moyen annuel de l'actif de type j

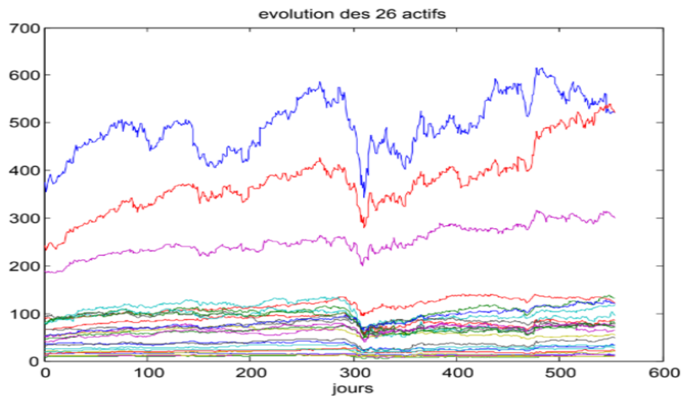
$$E_j = \frac{N_t}{M} \Sigma_{i=1}^{M} R(i,j)$$

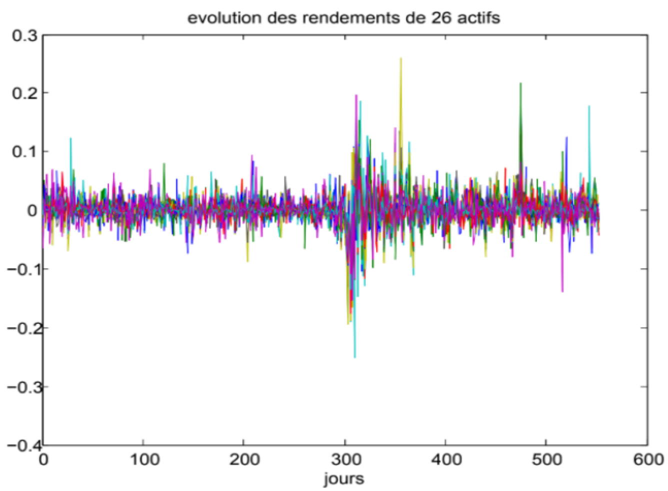
- Matrice de covariances

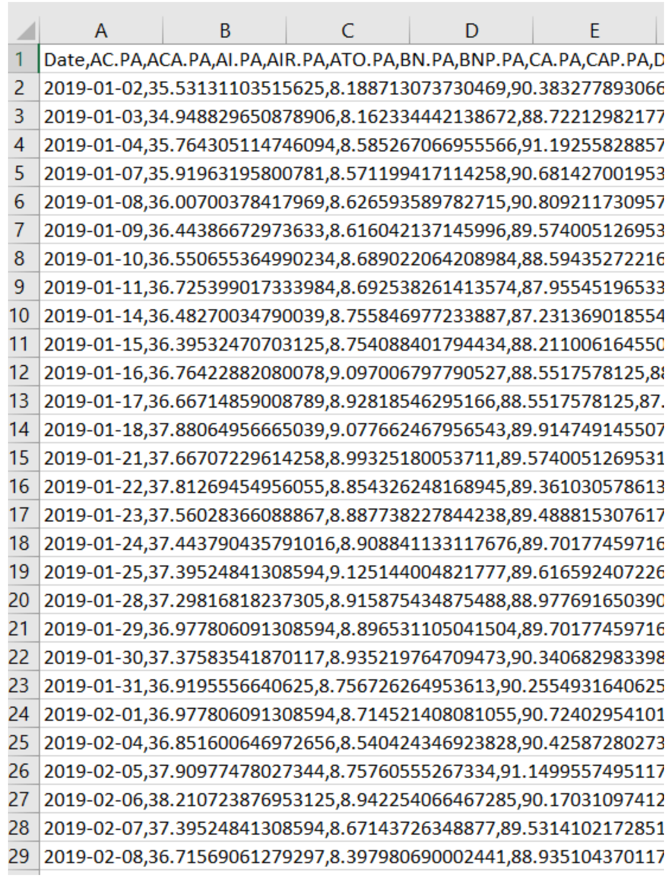
$$\Gamma_{ij} = (\frac{\sum_{k=1}^{M} R_{ik}^T \cdot R_{kj}}{N} - E_i^T \cdot E_j) \cdot N_t$$

Actifs CAC40

- CAC 40
- Evolution des actifs
- Rendments de actif







	A	B	C	D	E
1	Date,AC.PA,ACA.PA,AI.PA,AIR.PA,ATO.PA,BN.PA,BNP.PA,CA.PA,CAP.PA,D				
2	2019-01-02,35.5313110351562 5,8.18871307373046 9,90.383277893066				
3	2019-01-03,34.948829650878906,8.162334442138672,88.72212982177				
4	2019-01-04,35.764305114746094,8.585267066955566,91.19255828857				
5	2019-01-07,35.91963195800781,8.571199417114258,90.681427001953				
6	2019-01-08,36.00700378417969,8.626593589782715,90.809211730957				
7	2019-01-09,36.44386672973633,8.616042137145996,89.574005126953				
8	2019-01-10,36.550655364990234,8.689022064208984,88.59435272216				
9	2019-01-11,36.725399017333984,8.692538261413574,87.95545196533				
10	2019-01-14,36.48270034790039,8.755846977233887,87.231369018554				
11	2019-01-15,36.39532470703125,8.754088401794434,88.21100616455 0				
12	2019-01-16,36.76422882080078,9.097006797790527,88.5517578125,8				
13	2019-01-17,36.66714859008789,8.928185462951 66,88.5517578125,87.				
14	2019-01-18,37.88064956665039,9.077662467956543,89.914749145507				
15	2019-01-21,37.66707229614258,8.993251800537 11,89.5740051269531				
16	2019-01-22,37.81269454956055,8.854326248168945,89.361030578613				
17	2019-01-23,37.56028366088867,8.887738227844238,89.488815307617				
18	2019-01-24,37.443790435791016,8.908841133117676,89.70177459716				
19	2019-01-25,37.39524841308594,9.125144004821777,89.61659240722 6				
20	2019-01-28,37.29816818237305,8.915875434875488,88.97769165039 0				
21	2019-01-29,36.977806091308594,8.896531105041504,89.70177459716				
22	2019-01-30,37.37583541870117,8.935219764709473,90.340682983398				
23	2019-01-31,36.9195556640625,8.756726264953613,90.2554931640625				
24	2019-02-01,36.977806091308594,8.714521408081055,90.72402954101				
25	2019-02-04,36.851600646972656,8.540424346923828,90.42587280273				
26	2019-02-05,37.90977478027344,8.75760555267334,91.1499557495117				
27	2019-02-06,38.210723876953125,8.942254066467285,90.17031097412				
28	2019-02-07,37.39524841308594,8.67143726348877,89.5314102172851				
29	2019-02-08,36.71569061279297,8.397980690002441,88.935104370117				

Solution analytique

- Problème d'optimisation convexe et Lagrangien.

$$\begin{cases} \min \quad \sum_{ij}^{N} \Gamma_{ij} X_i X_j \\ \sum_{i=1}^{n} X_i E_i \geq R_0 \\ \sum_{i=1}^{N} X_i = 1 \end{cases}$$

- $L(X, \lambda, \mu) =$
  $\sum_{ij}^{N} \Gamma_{ij} X^i X^j + \lambda(\sum_{i=1}^{N} X^i E^i - R_0) + \mu(\sum_{i=1}^{N} X^i - 1)$
- Condition de Kutn-Tucker
  - $2 \cdot \sum_{j=1}^{N} \Gamma_{ij} \cdot X_j + \lambda - \mu E_j = 0$
  - $\lambda(\sum_{i=1}^{N} X^i E^i - R_0) = 0$
  - $(\sum_{i=1}^{N} X_i - 1) = 0$

Solution analytique

- Problème d'optimisation convexe

$$\begin{cases} \min & \sum_{ij}^{N} \Gamma_{ij} X_i X_j \\ & \sum_{i=1}^{n} X_i E_i \geq R_0 \\ & \sum_{i=1}^{n} X_i = 1 \end{cases}$$

- Solution analytique • si $\frac{A}{B} > R_0$

$$X_i = \frac{\sum_j \Gamma_{ij}^{-1} U_j}{B}, \qquad R_0 = \frac{A}{B}$$

• si $\frac{A}{B} \leq R_0$

$$X_i = \frac{R_0 B - A}{BC - A^2} \sum_j \Gamma_{ij}^{-1} E_j - \frac{R_0 A - C}{BC - A^2} \sum_j \Gamma_{ij}^{-1} U_j$$

- $A = \sum_{ij} \Gamma_{ij}^{-1} E_j U_i$, $B = \sum_{ij} \Gamma_{ij}^{-1} U_j U_i$, $C = \sum_{ij} \Gamma_{ij}^{-1} E_j E_i$

Solution analytique

- $A = \sum_{ij} \Gamma_{ij}^{-1} E_j U_i,\ B = \sum_{ij} \Gamma_{ij}^{-1} U_j U_i,\ C = \sum_{ij} \Gamma_{ij}^{-1} E_j E_i$
  - U=ones$(1, N)$
  - E est le vecteur de rendement annuel

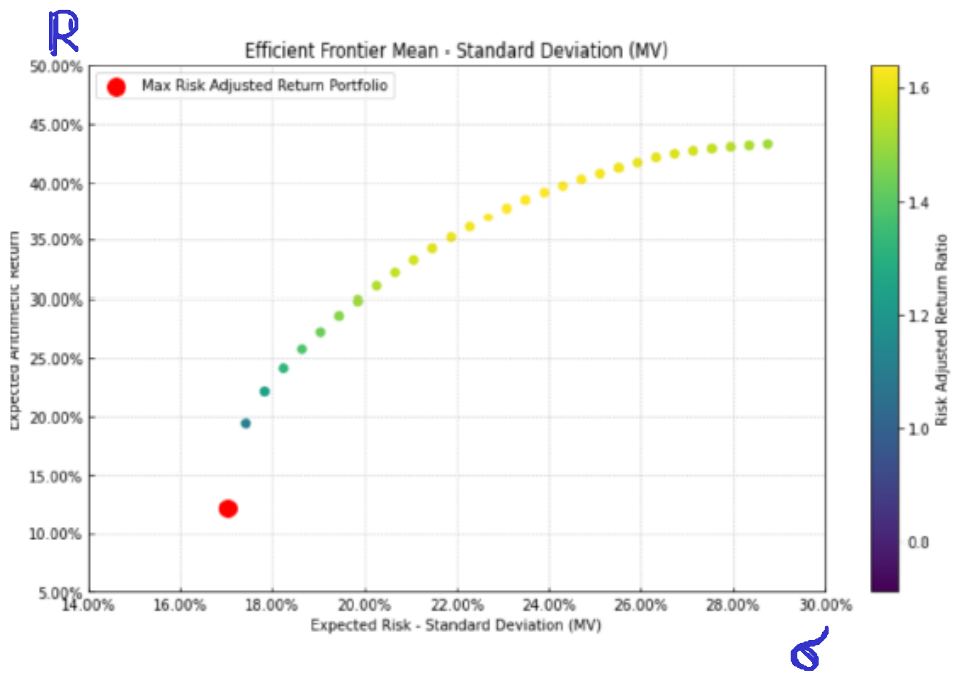
Frontière efficiente de Markowitz

- Chaque actif possède un couple rendement/risque (R, σ), $\quad \sigma^2 = \sum_{ij}^{N} \Gamma_{ij} X_i X_j$ qu'il est possible de représenter graphiquement. Pour chaque rendement (R), il existe un portefeuille qui minimise le risque σ. À l'inverse, pour chaque niveau de risque, on peut trouver un portefeuille maximisant le rendement attendu. L'ensemble de ces portefeuilles est appelé frontière efficiente.
- Pour un Portefeuille efficient aucun autre n'est meilleur au sens des critères que sont la moyenne et la variance.

Frontière efficiente de Markowitz

- Pour un Portefeuille efficient aucun autre n'est meilleur au sens des critères que sont la moyenne (R) et la variance σ.
- Equation de frontière efficiente

$$\frac{\sigma^2}{\left(\frac{1}{\sqrt{2B}}\right)^2} - \frac{\left(R - \frac{A}{B}\right)^2}{\left(\frac{\sqrt{BC - A^2}}{B}\right)^2} = 1$$

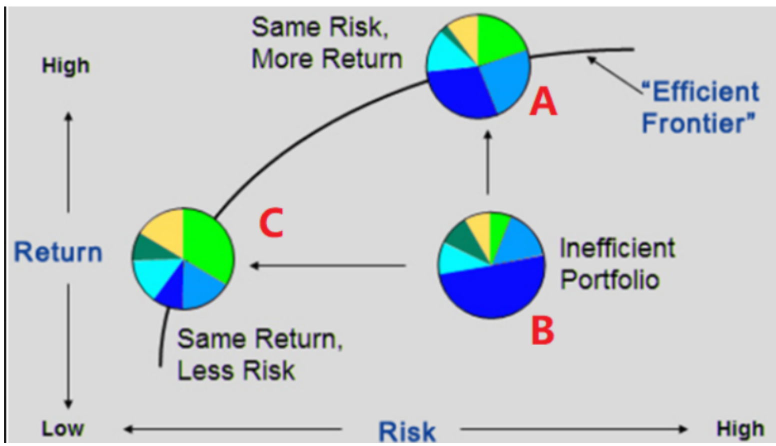


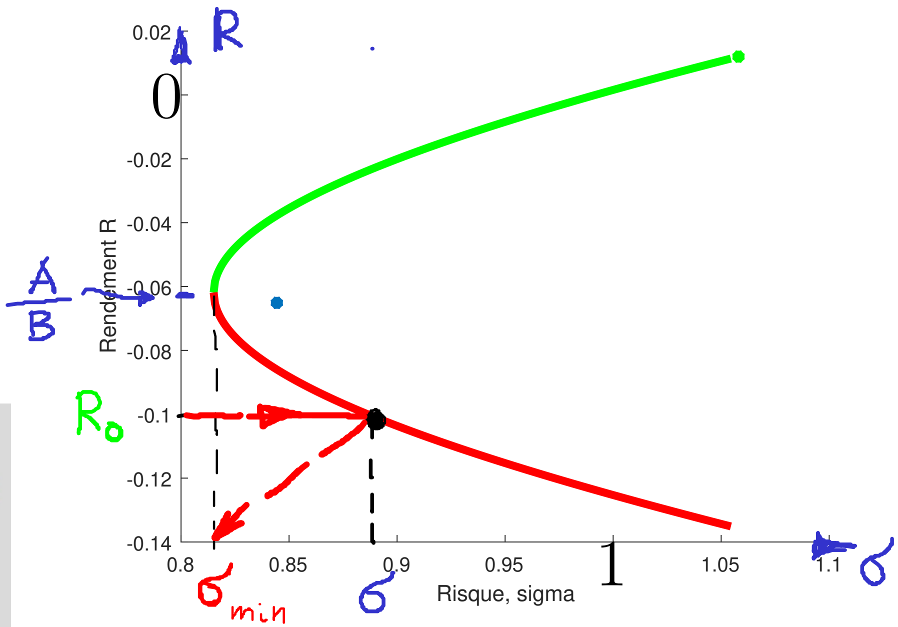
Frontière efficiente de Markowitz

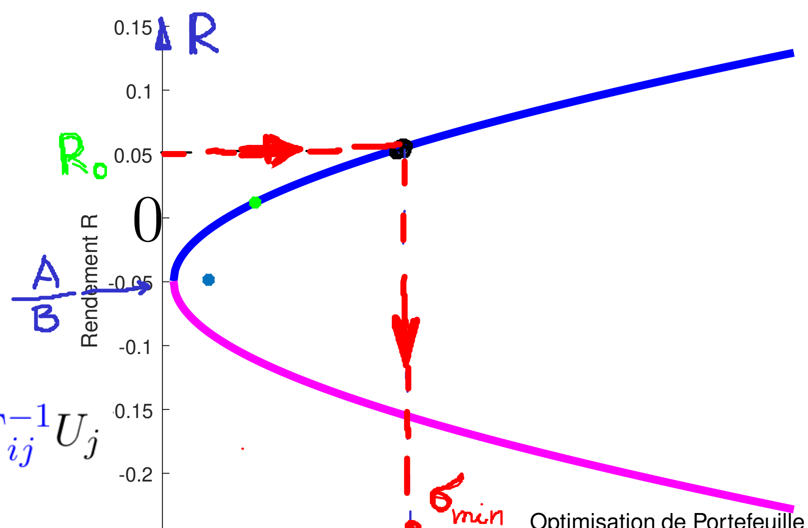
- Absence de l'actif sans risque

$$X_i = \frac{\sum_j \Gamma^{-1}_{ij} U_j}{B}$$

- $\frac{A}{B} > R_0$





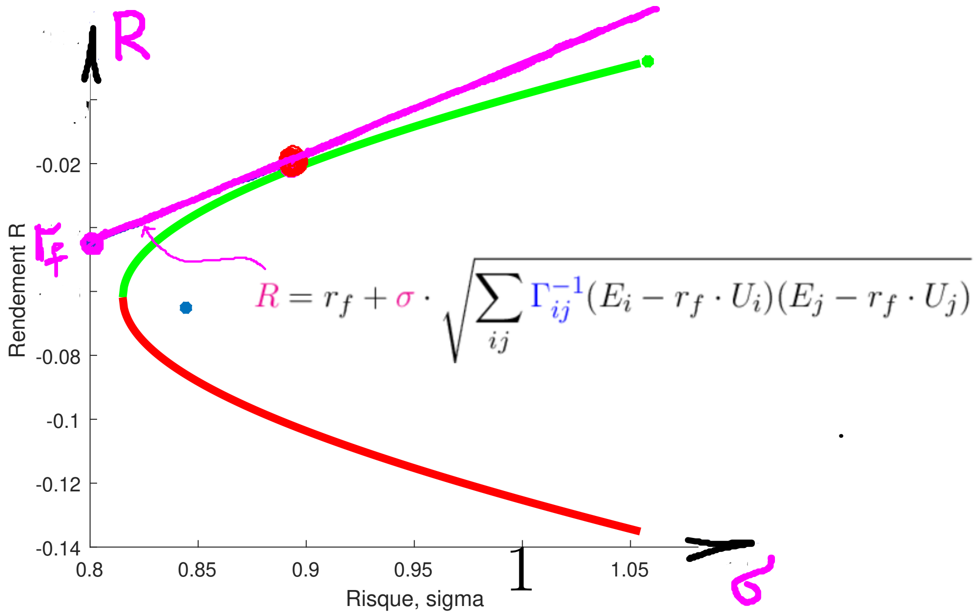


- $\frac{A}{B} < R_0$

$$X_i = \frac{R_0 B - A}{BC - A^2} \sum_j \Gamma^{-1}_{ij} E_j - \frac{R_0 A - C}{BC - A^2} \sum_j \Gamma^{-1}_{ij} U_j$$

Frontière efficiente de Markowitz

- Presence de l'actif sans risque
- Théorème des deux fonds: En présence de l'actif sans risque, tous les investisseurs répartissent leur richesse entre l'actif sans risque un même fund risqué reproduisant l'indice de marché déterminé par répartiion des acifs risqués.



- Le portefeuille de tangence, le point rouge dans l'image ci-dessus, est le portefeuille dit optimal qui réalise le ratio de Sharpe le plus élevé possible. Au fur et à mesure que l'on s'éloigne de ce point l'excès de rendement par rapport au risque, sera plus faible.

Théorie de Black et Scholes

- La valeur d'un actif i est modélisée par un mouvement brownien géométrique

$$S_i(t) = S_i(0)\exp((r - \frac{\sigma^2}{2})t + \sigma W_i(t)), \quad i \leq \mathbb{N}$$

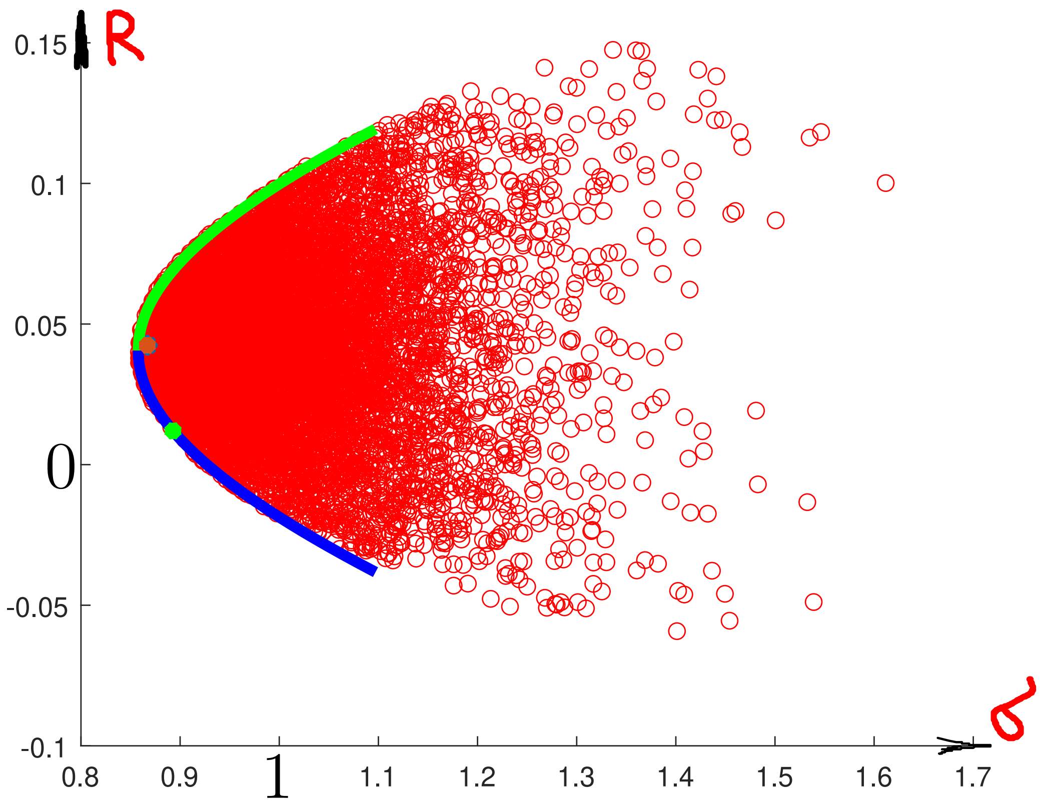
- $W_i(t)$ est un mouvement Brownien,
- σ est la volatilité

Simulation des portefeuilles

- Travail à faire:
- Simulation de l'évolution des actif $S(t_i, j)$
- Calcul de l'espérance E_j de chaque actif et de sa matrice de covariance Γ_{ij} à partir des donnés du marché.
- Simulation des allocations de N actifs:
  - $X_i = U^{(i)} / \sum(U)$, $\quad U^{(i)}$ suit la loi uniforme
  - $\sum X_i = 1.$
- Calcul des rendements aléatoires des portefeuilles $R = \sum X_i E_i$ et des risques $\sigma^2 = \quad \sum_{ij}^{N} \Gamma_{ij} X_i X_j$
- Graphe d'un nuage des porteufeuilles aléatoires (R, σ)
- Graphe d'une la frontière efficiente.

Frontière efficiente

- Tout les portefeiulles simulés sont à l'interieures d'une frontière efficiente



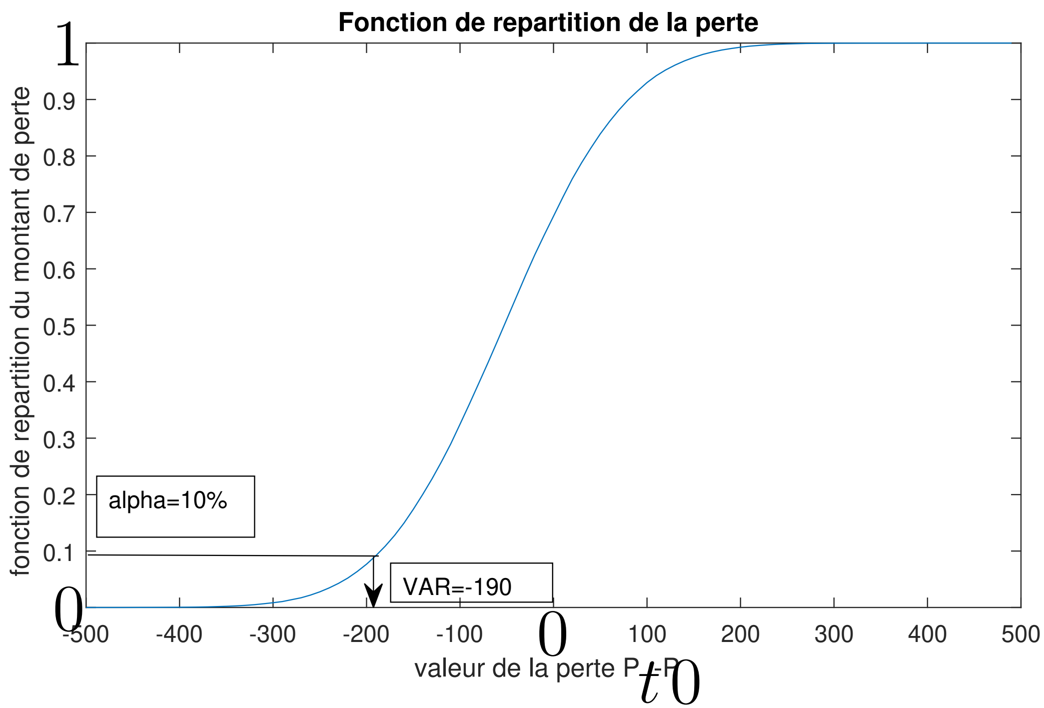
VAR d'un Portefeuille

- La VaR répond à l'affirmation suivante : "Nous sommes certains à la probabilité $1-\alpha = 90\%$ que nous n'allons pas perdre plus de VAR euros sur T prochains jours"

$$\mathbb{P}[R_T - R_0 \leq VaR] = \alpha$$

- Simulation des Rendements aléatoires $R = \sum X_i E_i$ ou les allocations X_i sont aléatoires

- Simulation de la Fonction de repartition $F_R(x) = \mathbb{P}[R_T - R_0 \leq x]$



Contraintes $X_i \geq 0, \quad i = 1, 2, ...N$

- Algorithme de Lagrange-Newton les contraintes

$$X_i \geq 0, \quad i = 1, 2, ...N$$

  ne sont pas imposées. Les ventes à la découvertes sont permises.

- Algorithme de Spinu (et ces variantes) les contraintes $X_i \geq 0, \quad i = 1, 2, ...N$ sont imposées
- Fonction SCO.MINIMIZE de Python admet l'imposition des contraintes $X_i \geq 0, \quad i = 1, 2, ...N$
- Fonction QUADPROG de MatLab admet l'imposition des contraintes $X_i \geq 0, \quad i = 1, 2, ...N$

Algorithme de Lagrange-Newton

- Contrainte

$$\sum_{i=1}^{N} X^i = 1$$

est imposée

- Contraintes

$$X_i \geq 0, \quad i = 1, 2, ...N$$

ne sont pas imposées. Les ventes à la découvertes sont permises.

Algorithme de Lagrange-Newton

- Construction d'une suite des points X_k, proportions de la somme investies dans les actions.
  • On utilise l'algoritme:
  $X_{k+1} = X_k + d_k,$
  où d_k est la direction de descente
  • On cherche aussi une suite des multiplicateurs de Lagrange λ_k
  $\lambda_{k+1} = \lambda_k + \xi_k,$
  où ξ_k est de la direction de descente dans l'espace de multiplicateurs de Lagrange.
- Contraintes: $h_1(X) = \sum_{i=1}^{n}(X^i E^i - R_0)$,
  $h_2(X) = \sum_{i=1}^{n}(X^i - 1)$
- Lagrangien
  $L(X, \lambda, \mu) = \sum_{ij}^{n} \Gamma_{ij} X^i X^j + \lambda \cdot h_1(X) + \mu \cdot h_2(X)$

Algorithme de Lagrange-Newton

- Algorithme de Lagrange-Newton

- Choisir x_0, ε.
- Choisir λ_0
- Tant que le critère d'arret n'est pas satisfait trouver d_k et ξ_k.

$$\begin{pmatrix} d_k \\ \xi_k \end{pmatrix} = - \begin{pmatrix} \nabla_x \nabla_x L(x_k, \lambda_k) & [\nabla h(x_k)]^T \\ \nabla h(x_k) & 0 \end{pmatrix}^{-1} \cdot \begin{pmatrix} [\nabla_x L(x_k, \lambda_k)]^T \\ h(x_k) \end{pmatrix}$$

- Poser $x_{k+1} = x_k + d_k$
- Poser $\lambda_{k+1} = \lambda_k + \xi_k$
- Poser $k = k + 1$
- Fin Tant que

Algorithme de LL. Example

Toy - Problème 1

$$\begin{cases} \min & f_1(x) = \frac{1}{2}(x^2 + y^2) \\ & x + y = 1 \end{cases}$$

Algorithme de Lagrange-Newton

- Considérons le Problème 1.
  On pose (avec les notations habituelles)

$$L(x,y,\lambda) = x^2 + y^2 + \lambda(x+y-1), \qquad h(x,y) = x+y-1$$

Montrer que

$$\begin{pmatrix} \nabla_x \nabla_x L(x_k,\lambda_k) & [\nabla h(x_k)]^T \\ \nabla h(x_k) & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{pmatrix}$$

et

$$\begin{pmatrix} [\nabla_x L(x_k,\lambda_k)]^T \\ h(x_k) \end{pmatrix} = \begin{pmatrix} x+\lambda \\ y+\lambda \\ x+y-1 \end{pmatrix}$$

Expliquer qu'on peut utiliser le critère d'arrêt comme

$$\sqrt{||d_k||^2 + ||\xi_k||^2} < \varepsilon$$

Algorithme d'Uzawa

- Algorithme d'Uzawa pour une fonction quadratique avec des contraintes affines:

$$\begin{cases} min \quad f(x) = \frac{1}{2}\langle Ax, x\rangle + \langle b, x\rangle \\ Qx - c = 0 \end{cases}$$

- Choisir $\lambda_0 \in \mathbb{R}^p$
- Choisir $x_0 \in \mathbb{R}^n$
- Choisir un pas $\rho > 0$
- Tant que le critère d'arret n'est pas satisfait ($||Qx_k - c|| > \varepsilon$):
- Calculer x_{k+1} par la résolution du sous-problème:

$$min_x \ \frac{1}{2}\langle Ax, x\rangle + \langle b, x\rangle + \langle \lambda_k, Qx - c\rangle$$

- Calculer $\lambda_{k+1} = \lambda_k + \rho(Qx_{k+1} - c)$
- Fin Tant que

- Pour calculer explicitement x_{k+1}, il faut utiliser une des méthodes d'optimisation sans contraintes: Gradient à pas optimal, Gradient conjugué, Newton, ...

Variantes de l'algorithme de Usawa

- Contrainte $\sum_{i=1}^{N} X^i = 1$ est imposée
- Contraintes $X_i \geq 0, \quad i = 1, 2, ...N$ sont imposées. Les ventes à la découvertes sont interdites.
- Article : Algorithme de Spinu
- Article :Théophile Griveau-Billion Quantitative Research Lyxor Asset Management, Paris A Fast Algorithm for Computing High-dimensional Risk Parity Portfolios
- Article : Jaehyuk Choi, Rong Chen "Improved iterative methods for solving risk parity portfolio" Preprint submitted to Journal of Derivatives and Quantitative Studies, arXiv:2203.00148v1 [q-fin.PM] 28 Feb 2022

- Choose x_0, $\lambda_0 = 1$, $k = 0$, and iterate:

$$
\begin{aligned}
&\text{while } \lambda_k > \varepsilon \text{ and } k < max\_iter\\
&u_k = Cx_k - b \circ x_k^{-1}\\
&H_k = C + \mathrm{diag}(b \circ x_k^{-2})\\
&h_k = -H_k^{-1} u_k\\
&\delta_k = \|h_k \circ x_k^{-1}\|_\infty\\
&\lambda_k = \sqrt{-u_k^{\mathrm{T}} h_k}\\
&\text{if } \lambda_k > \lambda_*\\
&\quad x_{k+1} = x_k + \tfrac{1}{1+\delta_k} h_k\\
&\text{else}\\
&\quad x_{k+1} = x_k + h_k
\end{aligned}
$$

- Once optimal x^* is found, set $w = x^* / \sum_{i=1}^n x_i^*$.

AN ALGORITHM FOR COMPUTING RISK PARITY WEIGHTS

FLORIN SPINU

ABSTRACT. Given a portfolio of assets (or return streams), the risk budget allocation problem seeks the long-only portfolio, fully invested in those assets, with the following property: the contribution of each asset to the risk of the portfolio equals a predetermined weight. When the predetermined weights arel equal the solution is the risk parity portfolio. Mathematically this reduces to to solving the nonlinear equation $Cx = b/x$, where C is a positive definite (covariance) matrix and x and b are column vectors with positive entries. We prove that this equation has unique solution and we give an efficient algorithm to compute it based on Newton's method.

A Fast Algorithm for Computing High-dimensional Risk Parity Portfolios*

Théophile Griveau-Billion
Quantitative Research
Lyxor Asset Management, Paris
theophile.griveau-billion@lyxor.com

Jean-Charles Richard
Quantitative Research
Lyxor Asset Management, Paris
jean-charles.richard@lyxor.com

Thierry Roncalli
Quantitative Research
Lyxor Asset Management, Paris
thierry.roncalli@lyxor.com

September 2013

Abstract

In this paper we propose a cyclical coordinate descent (CCD) algorithm for solving high dimensional risk parity problems. We show that this algorithm converges and is very fast even with large covariance matrices ($n > 500$). Comparison with existing algorithms also shows that it is one of the most efficient algorithms.

Keywords: Risk parity, risk budgeting, ERC portfolio, cyclical coordinate descent algorithm, SQP algorithm, Jacobi algorithm, Newton algorithm, Nesterov algorithm.

JEL classification: G11, C60.

1 Introduction

In this paper, we focus on risk parity (or risk budgeting) portfolios. The underlying idea is to do allocation by risk, not by capital. In this case, the portfolio manager defines a set of risk budgets and then compute the weights of the portfolio such that the risk contributions match the risk budgets.

From a mathematical point of view, a risk budgeting (or RB) portfolio is defined as follows (Roncalli, 2013):

$$\begin{cases} \mathcal{RC}_i(x) = b_i \mathcal{R}(x) \\ b_i > 0 \\ x_i > 0 \\ \sum_{i=1}^n b_i = 1 \\ \sum_{i=1}^n x_i = 1 \end{cases}$$

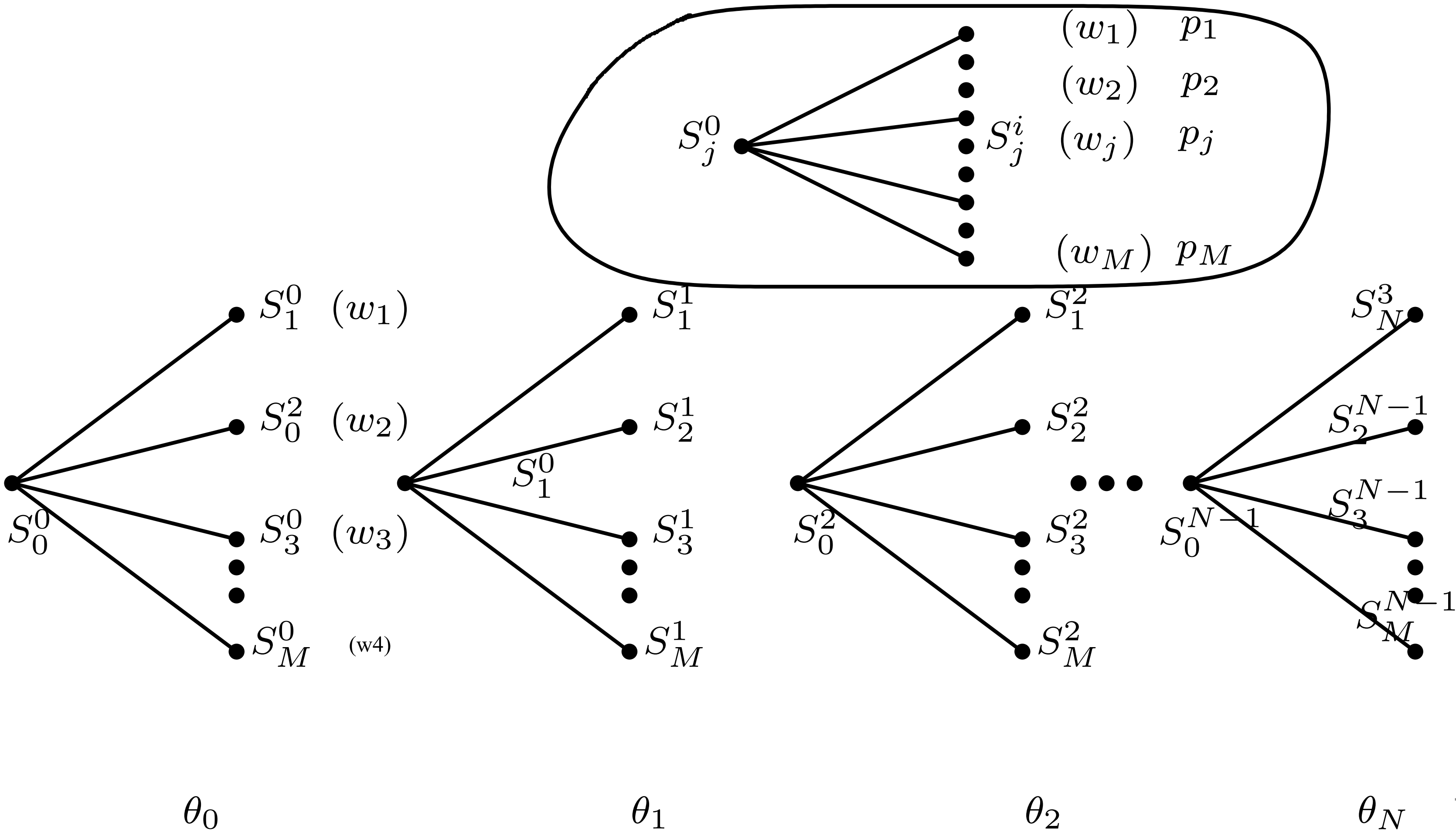
*We are grateful to Florin Spinu for providing us his code on the Newton-Nesterov algorithm and for stimulating discussions on risk parity optimization.

Ouverture

- Portefeuille dans l'espace $\mathbb{P}$
- Black-Litterman model

Portefeuille dans l'espace $\mathbb{P}$

- On applique des méthodes d'optimisation sans contraintes. On utilise "Méthode directe" (cours de Pr. Aktar "Gestion de portefeuille")) et on travaille dans l'espace de probabilité $\mathbb{P}$.



Portefeuille et Fonction d'utilité

- Notations:

• U est la fonction d'utilité.
• N est la quantité des actifs.
• M est le nombre des différentes états pour chaque actif.
• $S_i(\omega_j) = S_j^i$ est la prix de l'actif , i est le type de l'actif, j est l'état de l'actif.
• Explicitons le problème pour $M = 3$ et $N = 3$.
• La valeur du portefeuille composé d'un actif sans risque et 2 actifs risqué à $t = 0$

$$V_0 = \theta_0 S_0^0 + \theta_1 S_0^1 + \theta_2 S_0^2$$

Supposons que $S_0^0 = S_0^1 = S_0^2 = 1$ donc on a une contrainte:

$$K = \theta_0 + \theta_1 + \theta_2$$

ptimisation par la Méthode de Descente

- Problème d'optimisation avec contraintes

$$\begin{cases} \max_{\theta_0,\theta_1,\theta_2} F = \sum_{j=1}^{M} p_j \, U[(1+r)\theta_0 + \sum_{i=1}^{N-1} \theta_i S_j^i] \\ \theta_0 + \sum_{i=1}^{N-1} \theta_i = K \end{cases}$$

• On a utilisé l'espérance du portefeuille $E[V_T] = p_1 \cdot V_T(\omega_1) + p_2 \cdot V_T(\omega_2) + p_3 \cdot V_T(\omega_3)$
• On injecte $\theta_0 = K - \theta_1 - \theta_2$ dans la fonction F et on obtient le problème sans contraintes:

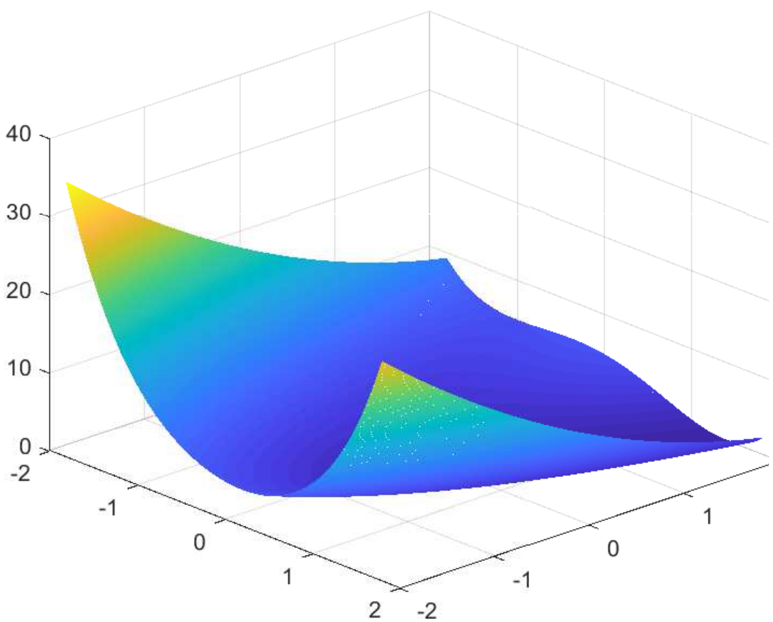
- Problème d'optimisation sans contraintes.

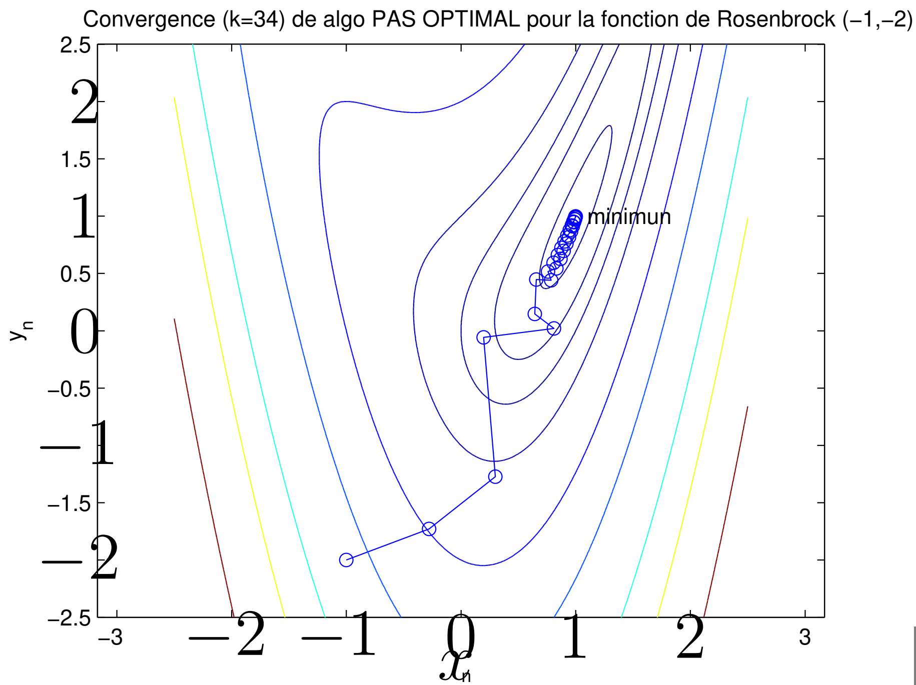
$$\left\{ \min_{\theta_1,\theta_2} F = -\sum_{j=1}^{M} p_j \, U[\sum_{i=1}^{N-1} \theta_i((S_j^i - (1+r)) + K(1+r))] \right.$$

• Utilisons pour la fonction-utilité une fonction logarithmique $U = \log(\theta)$.

Méthode de Descente. Visualisation.

- Suite de deschente X_{k+1} : $\quad X_{k+1} = X_k + t_k d_k$
- Fonction de Rosenbrock
  $f(x,y) = (x^2 - y)^2 + (x-1)^2, \quad X_0 = (-1,-2)$
- Fonctions MatLab: mesh, surf, contour





Descente du Gradient à pas optimal

- On approxime en chaque point X_k une fonction $f(x)$ par une fonction quadratique

$$f(X_k + t\, d_k) = f(X_k) + t \sum_i \frac{\partial f}{\partial x^i}(X_k) d_k^i + \frac{t^2}{2} \sum_{i,j} \frac{\partial^2 f}{\partial x^i \partial x^j}(X_k) d_k^i d_k^j + o|d|^2$$

La matrice $H^{ij}(X_k) = \frac{\partial^2 f}{\partial x^i \partial x^j}(X_k)$ s'appelle la matrice hessienne en point X_k.

- Direction de decente $d_k(X_k) = -\nabla f(X_k)$ est une fonction de X_k.
- Algorithme pour une fonction quelconque

- Choisir X_0 et ε
- Pour $k = 0...N$ faire
- Poser $d_k(X_k) = -\nabla f(X_k)$
- Si $||d_k(X_k)|| < \varepsilon$ stop
- Sinon poser $t_k = \frac{||d_k(X_k)||^2}{\langle H(X_k) d_k, d_k \rangle}$
- Fin Si
- Poser $X_{k+1} = X_k + t_k d_k$
- Fin Pour