
AlgoInvest & Trade

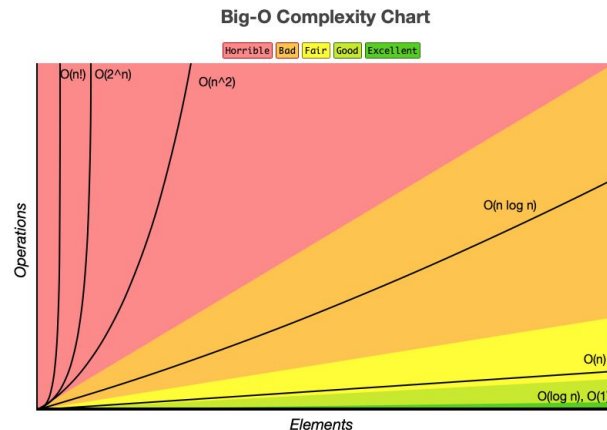
Projet OC n°7

Vernet Audric - Septembre 2021

2 algorithmes différents

Nous allons commenter deux algorithmes visant à déterminer la meilleure combinaison d'actions possible en termes de gains, avec une limite de portefeuille fixée à 500€

Nous disposons de deux jeux de données brutes contenant respectivement et approximativement 1000 actions





1. Description et analyse

Choisir le bon algorithme est indispensable dans un monde où les ressources sont limitées tant d'un point de vue technique que d'un point de vue économique

- **Analyse de l'algorithme de force brute**
Une solution parfaite mais trop complexe
- **Analyse de l'algorithme optimisé**
Une solution imparfaite mais réaliste et économe en temps et en ressources
- **Conclusion**
Avant la comparaison avec Sienna



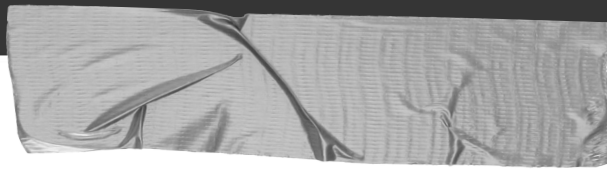
Description de l'algorithme de force brute

Cet algorithme va générer toutes les combinaisons possibles et toutes les calculer pour trouver la solution **parfaite**.

Nous avons une liste de 20 actions.

Une fonction sert à calculer le prix d'une combinaison, une autre sert à calculer le bénéfice. A l'aide de la fonction combinations du module itertools, nous stockons toutes les combinaisons possibles à condition que le prix ne dépasse pas 500€.

Nous parcourons ensuite cette liste pour trouver la combinaison au bénéfice le plus haut.



Avantages et inconvénients de l'algorithme de force brute

Etant donné que l'on teste toutes les combinaisons possibles, cet algorithme génère de nombreuses opérations, un volume de données conséquent et ne s'exécute dans un temps raisonnable que pour une toute petite quantité de données.

Avantage: trouve la solution parfaite

Inconvénient: inadapté à des plus gros volumes de données (grosse complexité temporelle et spatiale)

Notation Big-O: $O(n!)$



Description de l'algorithme optimisé

Cet algorithme opère un premier tri afin de ne parcourir la liste qu'une seule fois. Il utilise un algorithme de tri natif en python, de type timsort; il ne s'agit pas d'une opération linéaire néanmoins il est très efficace.

La liste est donc nettoyée et triée en amont, plaçant les actions **les plus rentables** au sommet de la liste.

Une fonction permet de parcourir la liste et d'acheter chaque action possible dans la limite de 500€



Avantages et inconvénients de l'algorithme optimisé

Cette fois, nous ne testons pas toutes les combinaisons possibles, cela réduit considérablement le nombre d'opérations, et par conséquent la complexité spatiale et temporelle. A l'aide d'un tri et d'une itération conditionnelle nous parvenons à une solution réaliste et raisonnable.

Avantage: solution quasi-instantanée, faible complexité spatiale et temporelle

Inconvénient: n'essaye pas toutes les combinaisons, solution imparfaite qui a ses limites; voir exemple de cas limite et conclusion

Notation Big-O: $O(\ln(n))$

Un exemple simple de cas limite avec l'algorithme optimisé

4 actions triées et 10€ de portefeuille maximum:
(5€/25%, 3€/24%, 3€/23%, 3€/22%)

Algorithme optimisé: 1,97€ de gains avec 8€ investis
Solution parfaite: 2,07€ de gains avec 9€ investis

La solution optimisée viendra choisir la première et la seconde action car une fois 8€ dépensés, le portefeuille ne contient plus assez d'argent pour acheter une autre action, alors que la solution des 3 dernières actions est la meilleure en termes de gains.

On note que si les gains sont supérieurs, le rendement est inférieur (24,6% de rendement pour l'algorithme contre 23% pour la solution qui génère le plus de gains)



Cependant...

...les écarts de bénéfice entre actions étant beaucoup moins importants que dans cet exemple, l'algorithme optimisé trouve une solution avec des rendements et des gains qui **s'approche de près de la "meilleure" solution**



Conclusion

Prendre une décision d'investissement relève autant de l'art que de la science, le but étant de prendre la meilleure décision possible. Nous avons développé deux solutions aux stratégies différentes afin de nous aider à y parvenir. Ces deux solutions sont fonctionnelles dans leur portée, c'est à dire très peu de données pour la force brute, et beaucoup plus pour la solution optimisée. On peut sans hésiter qualifier la solution optimisée de meilleure solution à bien des égards, dans les contraintes de la réalité technologique et économique.



2. Comparaison avec Sienna

Nous allons ici comparer les résultats obtenus à l'aide de l'algorithme avec les choix de Sienna

- **Dataset 1**
Mieux que Sienna
- **Dataset 2**
Toujours mieux que Sienna

Sienna

Prix: 498.76
Bénéfice: 196.61
Nombres d'actions achetées: 1

Algorithme

Prix: 499.94
Bénéfice: 198.50
Nombres d'actions achetées: 25

Dataset 1

1,18€ de plus investis,
1,89€ de gains
supplémentaires pour
l'algorithme

Sienna

Prix: 489.24
Bénéfice: 193.78
Nombres d'actions achetées: 18

Algorithme

Prix: 499.98
Bénéfice: 197.76
Nombres d'actions achetées: 22

Dataset 2

10,74€ de plus investis,
3,98€ de gains
supplémentaires pour
l'algorithme