

# Cryptofolio :

## Plate-forme d'optimisation de portfolio de cryptomonnaies

Julien Denes, Michaël Trazzi

Mai 2018

## Introduction

### 1 Revue de la littérature existante

#### 1.1 Online Portfolio Selection : A Survey

Dans cet article, les auteurs proposent un panorama des techniques d'optimisation de portfolio à l'heure actuelle (mai 2013), et proposent une analyse des principes mathématiques sur lesquelles elles reposent. Ils commencent par proposer une distinction entre la “Mean Variance Theory” et la “Capital Growth Theory”. La première, basée sur la théorie de Markovitz, ne s'intéresse qu'à une seule période de temps : on sélectionne un portfolio fixé en cherchant le meilleur compromis entre “return” (mean) et risque (variance). La seconde théorie se focalise sur une sélection de portfolio multi-périodiques (ou séquentielle), c'est à dire en découpant une période en sous-séquences et en autorisant une modification du portfolio à la fin de chacune de ces périodes. On s'intéresse alors à sélectionner la meilleure rentabilité (ou taux de croissance). Les techniques de la “Capital Growth Theory” sont regroupées sous le nom de “Online Portfolio Selection”, sur lesquelles cet article se focalise.

Les auteurs découpent ces techniques en 5 catégories : Benchmarks, Follow-the-Winner, Follow-the-Loser, Pattern-Matching Approaches, et Meta-Learning Algorithms.

Les Benchmarks sont des algorithmes quasi-triviaux qui servent de repères : Buy and Hold (aucune modification du portfolio à chaque période), Best Stock (idem mais portfolio composé d'un unique stock, le meilleur), Constant Rebalanced Portfolios (le portfolio de la période suivante est toujours un portfolio fixé).

Les algorithmes Follow-the-Winner sont basés sur le principe de l'augmentation du poids des stocks avec la meilleure croissance à la période  $t$  pour former le portfolio du temps  $t + 1$ . Parmi eux, on pourra citer les Universal Portfolios (construit des portfolio composés d'un unique stock puis mixe les plus performants), Exponential Gradient (cherche à suivre le meilleur portfolio au temps  $t$  mais en restant proche du portfolio adopté précédemment), le très similaire Follow the Leader (qui ne tient pas compte de la proximité), ou Follow the Regularized Leader (idem mais avec un terme de régularisation, qui typiquement minimise la taille du vecteur de portfolio).

Les approches Follow-the-Loser se basent quant à elles sur la théorie de la mean reversion, qui affirme en sorte que les stocks ayant eu de mauvaises performances à une période  $t$  seront ceux qui auront de bonnes performances au temps  $t + 1$  (et vice versa). On citera l'algorithme Anticor (transfert le capital des bons performeurs au mauvais en proportion de la corrélation entre eux), ou encore Passive Aggressive Mean Reversion qui résout un programme linéaire de minimisation de la distance entre le portfolio de  $t$  et de  $t + 1$  sous contrainte de respecter la “mean reversion property”. Online Moving Average Reversion fonctionne sensiblement sur le même principe mais en se basant sur plus d'une période en amont. On pourra noter que les algorithmes sont, d'après les tests des auteurs, les plus performants.

Les approches de Pattern-Matching cherchent quant à eux d'abord à trouver un set  $C$  de prix similaires dans l'historique, puis à partir de celui-ci compose le portfolio le plus performant basé sur ce que ce set  $C$  laisse à prévoir.

Enfin, les algorithmes de Meta-Learning vont chercher à combiner plusieurs portfolios performants à une période  $t$  pour former celui de la période  $t + 1$ , et ce selon différentes techniques.

## 1.2 A Deep Reinforcement Learning Framework for the Financial Portfolio Management Problem

### 1.2.1 Introduction

Cet article s'intéresse à l'application des techniques de Machine Learning au problème de gestion du portfolio. Il existe des approches de type Deep Learning appliquées au marchés financier, mais celles-ci se focalisent sur la prédiction du prix, ce qui s'avère particulièrement difficile pour le cas des cryptomonnaies.

Les travaux d'apprentissage concernant l'algorithmic trading n'essayant pas de prédire le prix, et n'utilisant pas de modèle pré-existant, utilisent des techniques de Reinforcement Learning (RL). Cependant, ces travaux se contentent d'échanger seulement un stock, ce qui ne nous intéresse pas pour le cas du portfolio. Les techniques de Deep RL récentes consistant à entraîner deux politiques en même temps peuvent s'avérer trop difficiles et instables. C'est pourquoi dans ce papier est adoptée une approche de RL adaptée au cas du portfolio.

Cette approche utilise des réseaux de neurones (RNN, CNN et LSTM) prenant en entrée l'historique de stocks, pour prédire l'action adaptée à adopter à la fin de la période (e.g. acheter, vendre, etc.). Les résultats théoriques sont testés en pratique en effectuant un trading permanent sur Polonix.com, à intervalles de 30 minutes. Contrairement au marché habituel, le marché des cryptomonnaies est ouvert 24h/24 7j/7.

### 1.2.2 Définition du problème

Le trading continu est divisé sur des périodes de 30 minutes. On suppose qu'au début et à la fin de chaque période on peut acheter et vendre directement, sans influencer sur le marché global. Le Bitcoin est considéré être notre "cash", ou cryptomonnaie de référence.

A la fin de chaque période, en achetant/vendant on change notre vecteur de portefeuille, donné par la proportion de notre capital dans chacune des cryptomonnaies. Le but de l'algorithme est de générer une séquence de vecteurs de portefeuille permettant de maximiser le capital accumulé, en prenant en compte un coût constant de transaction de 0.25%.

### 1.2.3 Données utilisées

Les 12 cryptomonnaies avec la plus grande coinmarketcap ont été utilisées. Cela permet d'assurer la liquidité totale à la fin de chaque période de trading, ce que nous avons supposé. En outre, cela garantit que nous ne traitons pas le cas de cryptomonnaies récentes qui pourraient avoir un comportement instable. Pour le cas où les données n'existent pas (la cryptomonnaie n'existe pas encore) on donne des (faux) prix décroissants pour inciter notre algorithme à ne pas la considérer.

### 1.2.4 Apprentissage par Renforcement

Pour définir notre problème de Reinforcement Learning il nous faut définir l'agent et l'environnement. Ici l'agent est notre algorithme, et l'environnement est l'ensemble des prix (accessibles) des stocks. Le flux de l'ensemble des prix étant trop difficile à traiter, on considère que tout ce dont dispose notre agent sont les prix les plus haut, les plus bas et le cours de clôture des actions, pour

chaque période. Les états sont alors la donnée de cet environnement et du portfolio à la précédente période.

Afin de déterminer une politique optimale on utilise une montée de gradient avec mini-batch et du online learning (les nouvelles données du marché arrivent en continu durant l'apprentissage).

### **1.2.5 Résultats**

Cette approche surpasse toutes les méthodes traditionnelles. Des architectures utilisées pour les réseaux de neurones, la meilleure est CNN, suivie de près par un RNN basique, et bien meilleure qu'un LSTM.

Les hypothèses de liquidité totale et de non influence sur le marché sont cependant fortes. De plus, l'algorithme n'a pas été testé sur un marché réel plus traditionnel. Il faudrait dans ce cas étudier les corrélations entre le trading et la réaction du marché.