



**Charles Kubudi Cordeiro e Silva**

**Algoritmos para o problema de seleção online  
de portfolios**

**Dissertação de Mestrado**

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Programa de Pós-graduação em Informática do Departamento de Informática do Centro Técnico Científico da PUC-Rio.

Orientador: Prof. Marco Serpa Molinaro

Rio de Janeiro  
Fevereiro de 2019

Todos os direitos reservados. É proibida a reprodução total ou parcial do trabalho sem autorização da universidade, do autor e do orientador.

### **Charles Kubudi Cordeiro e Silva**

Graduou-se em Engenharia da Computação pela Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio). Em 2016 ingressou no programa de mestrado em Informática na Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio). Desde 2014, atua como pesquisador de estratégias quantitativas para o mercado financeiro.

#### Ficha Catalográfica

Kubudi Cordeiro, Charles

Algoritmos para o problema de seleção online de portfólios / Charles Kubudi Cordeiro e Silva; orientador: Marco Serpa Molinaro. – 2019.

v., 50 f: il. color. ; 30 cm

Dissertação (mestrado) - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática.

Inclui bibliografia

1. Informática – Teses. 2. Otimização de portfólios;. 3. Atualização multiplicativa de pesos;. 4. Mercado ações brasileiro.. I. Serpa Molinaro, Marco. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. III. Título.

CDD: 004

## Agradecimentos

Ao meu orientador Marco Molinaro, pela dedicação e paciência ao longo do processo de pesquisa.

Aos meus colegas de trabalho da Kadima Asset Management, por me ajudarem em várias etapas do trabalho.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

## Resumo

Kubudi Cordeiro, Charles; Serpa Molinaro, Marco. **Algoritmos para o problema de seleção online de portfólios**. Rio de Janeiro, 2019. 50p. Dissertação de Mestrado – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

A otimização online de portfólios é um problema de engenharia financeira que consiste na escolha sequencial de alocação de capital entre um conjunto de ativos, com o objetivo de maximizar o retorno acumulado no longo prazo. Com o avanço dos estudos de modelos de *machine learning*, diversos algoritmos estão sendo utilizados para resolver esse problema. Uma série de algoritmos seguem a metodologia *Follow-the-winner (FTW)*, onde o peso de ações com boa performance é aumentado baseado na hipótese de que a tendência de alta será mantida; outros seguem a metodologia inversa *Follow-the-loser (FTL)*, em que ações com má performance tem seu peso aumentado apostando em uma reversão dos preços. Algoritmos estado-da-arte do tipo *FTW* possuem garantia teórica de se aproximar da performance da melhor ação escolhida de antemão, entretanto, algoritmos do tipo *FTL* tem performance superior observada empiricamente. Nosso trabalho busca explorar a ideia de aprender quando utilizar cada uma das duas categorias. Os mecanismos utilizados são algoritmos de *online learning* com flexibilidade para assumir ambos comportamentos. Foi realizado um estudo da literatura sobre indicadores de memória em séries financeiras e sua possível utilização de forma explícita para escolha entre *FTL* e *FTW*. Posteriormente, propomos um método de se realizar o aprendizado entre essas duas categorias de forma online e de forma dinâmica para utilização em algoritmos de *online learning*. Em nossos experimentos, o método proposto supera o *benchmark* estabelecido UCRP com excesso de retorno de 36.76%.

## Palavras-chave

Otimização de portfólios; Atualização multiplicativa de pesos; Mercado ações brasileiro.

## Abstract

Kubudi Cordeiro, Charles; Serpa Molinaro, Marco (Advisor). **Algorithms for online portfolio selection problem**. Rio de Janeiro, 2019. 50p. Dissertação de Mestrado – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Online portfolio selection is a financial engineering problem which aims to sequentially allocate capital among a set of assets in order to maximize long-term return. With the recent advances in the field of machine learning, several models have been proposed to address this problem. Some algorithms approach the problem with a "Follow-the-winner" (FTW) methodology, which increases the weights of more successful stocks based on their historical performance. Contrarily, a second approach, "Follow-the-loser" (FTL), increases the weights of less successful stocks, betting on the reversal of their prices. Some state-of-the-art FTW type algorithms have the guarantee to asymptotically approach the same performance as the best stock chosen in hindsight, while FTL algorithms have empirical evidence of overperforming the previous. Our goal is to explore the idea of learning when to use each of those two algorithm categories. We do this by using online learning algorithms that are capable of switching between the described regimes. We review the literature for existing measures of time series memory and predictability, and explicitly use this information for choosing between FTW and FTL. Later, we propose a method for choosing between this two types of algorithms in an online and dynamic manner for usage together with online learning algorithms. The method outperforms the chosen benchmark UCRP in our experiments with 36.76% excess returns.

## Keywords

Portfolio optimization; Multiplicative weights update method; Brazilian stock market.

# Sumário

|         |  |    |
|---------|--|----|
| 1       | Introdução   | 11 |
| 1.1     | Métodos clássicos de seleção de portfólios   | 12 |
| 1.2     | Online learning para seleção de portfólios, e regimes Trend Following e Mean Reverting | 12 |
| 1.3     | Contribuições da tese  | 14 |
| 1.4     | Organização  | 15 |
| 2       | Conceitos Preliminares   | 16 |
| 2.1     | Aprendizado online com experts   | 16 |
| 2.2     | Multiplicative Weights Update (MWU)  | 17 |
| 2.3     | Utilizando online learning para seleção de portfólios                                  | 18 |
| 2.4     | Benchmark  | 21 |
| 2.5     | Modificações ao algoritmo MWU  | 22 |
| 2.5.1   | Fixed Share  | 22 |
| 2.5.2   | Sleeping Experts   | 23 |
| 2.5.3   | Stop loss  | 24 |
| 2.6     | Algoritmo Base Final   | 25 |
| 3       | Aprendizado com indicadores  | 27 |
| 3.1     | Indicadores de tendencia em séries financeiras   | 27 |
| 3.1.1   | Expoente de Hurst  | 27 |
| 3.1.2   | Autocorrelação   | 29 |
| 3.2     | Aprendizado offline com indicadores  | 30 |
| 3.2.1   | Setup Experimental   | 31 |
| 3.2.1.1 | Seleção do módulo do hiperparâmetro de aprendizado                                     | 31 |
| 3.2.1.2 | Benchmark  | 32 |
| 3.2.2   | Algoritmo com $\eta$ estático e distinto para cada ação                                | 32 |
| 3.2.3   | Algoritmo com $\eta$ dinâmico e distinto para cada ação                                | 33 |
| 3.2.3.1 | Divisão de série em janelas TF e MR  | 33 |
| 3.2.3.2 | Experimento com $\eta$ dinâmico  | 34 |
| 3.2.4   | Algoritmo com $\eta$ dinâmico para todo universo                                       | 35 |
| 3.3     | Conclusão  | 35 |
| 4       | Aprendizado online com meta-algoritmos   | 38 |
| 4.1     | Benchmark  | 38 |
| 4.2     | Aprendizado com parâmetros distintos para cada ação                                    | 39 |
| 4.2.1   | Um meta-algoritmo para partição de regimes $TF$ e $MR$ distintos por ativo             | 39 |
| 4.2.2   | Experimento  | 40 |
| 4.3     | Aprendizado com parâmetro único para todo universo                                     | 42 |
| 4.3.1   | Meta-algoritmo para aprendizado de parâmetro comum a todos os ativos                   | 42 |
| 4.3.2   | Experimento  | 43 |

|   |                            |    |
|---|----------------------------|----|
| 5 | Conclusão                  | 46 |
|   | Referências bibliográficas | 48 |

## Lista de figuras

|     |   |    |
|-----|---|----|
| 2.1 | Exemplo de execução do algoritmo <i>MWU</i> em todas ações que fazem parte do índice IBOVESPA no período de 2006 até 2010, utilizando dados diários de preço de fechamento do mercado. O eixo $y$ representa o retorno bruto da estratégia, que começa com $r^0 = 1.0$ . O <i>benchmark</i> UCRP é definido na Seção 2.4. Diferentes valores do parâmetro de aprendizado tem performances variadas. | 20 |
| 2.2 | Exemplo de execução do algoritmo <i>MWU</i> em todas ações que fazem parte do índice IBOVESPA no período de 2011 até 2018, utilizando dados diários de preço de fechamento do mercado. O <i>benchmark</i> UCRP é definido na Seção 2.4  | 21 |
| 2.3 | Comparação entre o <i>benchmark</i> utilizado e o índice IBOVESPA, no período de 01/2011 até 03/2018. O UCRP tem um retorno de 46.13%, enquanto o índice tem o retorno de 21.33%. A maior diversificação empregada pelo UCRP pode ser a justificativa da performance superior.  | 22 |
| 3.1 | Histograma da distribuição do expoente de Hurst das ações negociadas na bolsa brasileira.   | 29 |
| 3.2 | Distribuição da autocorrelação de grau $k = 1$ das ações negociadas na bolsa brasileira.  | 30 |
| 3.3 | Resultados da estratégia utilizando um $\eta$ distinto para cada ação, de acordo com a análise dos indicadores calculados de antemão. O eixo $y$ representa o retorno bruto da estratégia, com $r^0 = 1.0$ .  | 33 |
| 3.4 | Resultados da estratégia utilizando um $\eta$ distinto e dinâmico para cada ação, de acordo com a análise dos indicadores calculados de antemão.  | 35 |
| 3.5 | Divisão de regimes encontrada pelo algoritmo guloso para a série de preços do UCRP, de acordo com o indicador H.  | 36 |
| 3.6 | Divisão de regimes encontrada pelo algoritmo guloso para a série de preços do UCRP, de acordo com o indicador AC.   | 36 |
| 3.7 | Resultados da estratégia utilizando um $\eta$ dinâmico para todo mercado, de acordo com a análise dos indicadores calculados de antemão sob a série de retornos do UCRP.  | 37 |
| 4.1 | Exemplo de fluxo de atualização multiplicativa do meta-algoritmo com $n = 3$ .  | 41 |
| 4.2 | Histograma da distribuição do excedente retorno. O eixo $x$ representa o retorno excedente, em valores absolutos.   | 42 |



|     |   |    |
|-----|---|----|
| 4.3 | Exemplo de retorno acumulado do algoritmo proposto para os ativos CMIG4, ELET3, ITUB4 e USIM5. Para este conjunto de ativos a otimização feita resultou na escolha dos parâmetros de $\eta_{base} = 13.41$ e $\eta_{master} = 18.97$ . O algoritmo em questão está representado pela curva azul. Pode ser observado que a técnica supera os resultados dos algoritmos com $\eta$ estático em ambos casos - com hiperparâmetro de aprendizado positivo e negativos - e também o <i>benchmark</i> UCRP. | 43 |
| 4.4 | Exemplo de fluxo de atualização multiplicativa do meta-algoritmo de regime único.   | 44 |
| 4.5 | Histograma da distribuição do excedente retorno para o algoritmo com parâmetro de aprendizado comum a todos os ativos.  | 44 |

## Lista de tabelas

|     |  |    |
|-----|--|----|
| 2.1 | Setup para algoritmos usando aprendizado com experts | 16 |
|-----|--|----|

# 1

## Introdução

O uso de técnicas sistemáticas é uma tendência global na gestão de carteiras de ativos. Em comparação com a abordagem discricionária tradicional, a alocação utilizando modelos matemáticos possui diversas vantagens, como a capacidade de monitorar grandes volumes de dados, ser menos suscetível à erros operacionais e não ter influência de emoções no processo de tomada de decisões.

A seleção de portfólios é um dos problemas mais fundamentais de engenharia financeira, que contempla a alocação do patrimônio em um dado universo de ativos disponíveis. A montagem de portfólios ótimos pode ser dividida em duas etapas. A primeira etapa diz respeito à escolha do universo de ativos desejados para compor a carteira. Já a segunda etapa contempla a escolha ideal das alocações entre os ativos disponíveis. Esse trabalho é focado nesta segunda fase. Mais formalmente, um portfólio construído no universo de  $n$  ações é descrito por um vetor de alocações  $b = (b_1, b_2, \dots, b_n)$ , tal que  $b_i \geq 0$  e  $\sum_i b_i = 1$ .

As ações possuem uma variação constante em seus preços. Em um dado instante  $t$ , a performance de uma ação  $i$  pode ser medida através do retorno  $x_i^t$  do seu preço atual  $p_i^t$  em relação ao seu preço anterior  $p_i^{t-1}$  através da fórmula

$$x_i^t = \frac{p_i^t}{p_i^{t-1}} - 1.$$

A performance de todas as ações consideradas, portanto, pode ser representada por um vetor de retornos  $x^t = (x_1^t, x_2^t, \dots, x_n^t)$ . Em um dado instante  $t$ , em posse de um vetor de pesos (alocações)  $b^t$  e um vetor de retornos  $x^t$ , podemos calcular o retorno do portfólio  $r^t$  como

$$r^t = \langle b^t, x^t \rangle = \sum_{i=1}^n b_i^t x_i^t.$$

O objetivo do problema pode ser maximizar o somatório dos retornos do portfólio  $\sum_i^t r^i$ , ou alternativamente, maximizar o produto  $(\prod_i^t (r^i + 1)) - 1$ . O segundo caso considera reinvestimento dos juros (juros compostos), enquanto o primeiro não (juros simples).

## 1.1

### Métodos clássicos de seleção de portfólios

Uma das escolas mais importantes no estudo de otimização de portfólios é a análise de Média-Variância desenvolvida por Markowitz (1952) em seu aclamado trabalho. Markowitz discorre sobre a importância de se levar em conta não somente o retorno esperado (média) de cada ativo, mas também seu risco (variância) e suas dependências com outros componentes do portfólio.

O trabalho de Markowitz é bastante empregado para fins teóricos, porém na prática sua utilização é restrita por conta de suas limitações. O trabalho de DeMiguel et al. (2007), por exemplo, sugere que a estratégia de alocação uniforme é capaz de vencer o modelo de Média-Variância de Markowitz. Uma das limitações do modelo proposto é a sensibilidade às estimativas de retorno, utilizadas como entrada. Essa questão é discutida no trabalho de Best e Grauer (1991). O trabalho sugere que ao utilizar a média de retornos em uma janela como estimativa, o modelo se torna bastante instável e pouco confiável.

Após o trabalho de Markowitz, diversos outros trabalhos se dedicaram à tarefa de seleção de portfólios, propondo alterações na teoria tradicional. O artigo de Rather et al. (2017) faz um levantamento dos trabalhos existentes nessa área. Em geral, os modelos de otimização propostos tem em comum o objetivo de maximizar o retorno do portfólio, e ao mesmo tempo minimizar o seu risco. Estes podem ser divididos em modelos de otimização com função objetivo única, que pode ser a maximização do lucro ou minimização das perdas, e modelos de otimização com múltiplos objetivos.

Recentemente, novos modelos baseados em algoritmos de inteligência artificial vem sendo propostos para essa tarefa como descrito por Rather et al. (2017). Técnicas como redes neurais e algoritmos genéticos estão sendo utilizados.

## 1.2

### Online learning para seleção de portfólios, e regimes Trend Following e Mean Reverting

Em contraste com o trabalho de Markowitz, uma outra linha estudos de algoritmos para seleção de portfólios é através do modelo de *online learning*. A grande vantagem desse modelo é a de não assumir hipótese estatística, como por exemplo assumir que os retornos seguem uma distribuição. Além disso, a maioria dos algoritmos nesse modelo são bastante eficientes, sendo adequado para o crescente fluxo de dados atuais. O presente trabalho foca nesse modelo de *online learning*

Esse é um modelo de tomada de decisão sequencial onde as informações de performances passadas guiam a alocação do portfólio no próximo instante de tempo. Em particular, não utiliza nenhum modelo preditivo explícito sobre ganhos futuros dos ativos. Nesse modelo, a medida padrão de qualidade de um algoritmo é o chamado *regret*, definido como a diferença entre a perda/ganho do algoritmo e a da melhor ação em todo período. Veja Seção 2.1 para definição mais formal do modelo de online learning e conceitos relacionados.

*Online learning* vem sendo proposto como uma metodologia para a seleção de portfólios em diversos trabalhos. No trabalho de Li e Hoi (2014), é feito um levantamento do que foi produzido. Uma das categorias de algoritmos estudados são os do tipo *Follow-the-winner*, onde mais capital é alocado nas ações que estão com melhor performance. Essa classe de algoritmos parte da hipótese *Trend Following (TF)*, que acredita que ações costumam repetir a mesma direção dos retornos recentes. O Algoritmo de *Universal Portfolios (UP)* proposto por Cover (1991) é um exemplo, e o primeiro a atingir uma garantia de *regret* teórica. O autor prova que o *regret* do algoritmo proposto é garantia  $O(n \log T)$ , sendo  $n$  o número de ações e  $T$  o número de períodos. O problema do algoritmo de Cover é requerer um tempo de computação exponencial, tornando seu uso prático inviável.

Diversas melhorias ao algoritmo de Cover foram propostas, muitas vezes para considerar hipóteses mais realistas do mercado. Em (1996), Cover e Ordentlich estudaram a inclusão de *side information*, como por exemplo dados fundamentalistas sobre as empresas, tais como o lucro e valor contábil. Posteriormente, Cover e Ordentlich (1998) adicionaram ao algoritmo a possibilidade de lidar com *short selling* e margem. Blum e Kalai (1999) propuseram incorporar os custos de transação ao algoritmo.

Uma segunda categoria de algoritmos são os do tipo *Follow-the-loser*, que transferem a alocação para ações que estão sendo perdedoras. A hipótese é que o mercado possui um comportamento de *Mean Reversal (MR)*, onde ações que performaram mal recentemente devem se recuperar, e vice-versa. Esse é o caso do Anticor, algoritmo publicado por Borodin et al. (2004). Por conta de sua natureza de reversão a média, este algoritmo não possui garantia assintótica de *regret*. Por outro lado, empiricamente o algoritmo supera todos os algoritmos citados no trabalho de Li e Hoi (2014).

Uma outra indicação nessa direção é a seguinte. Vários algoritmos de *online learning* possuem um parâmetro de aprendizado  $\eta$  (o algoritmo *WMU*, utilizado nesta tese, é um deles). Tradicionalmente, um valor positivo é atribuído à esse parâmetro, fazendo o algoritmo assumir um comportamento seguidor de tendências (*TF*) onde o peso das ações com melhor performance

é gradualmente aumentado. Todavia, na literatura existem diversos relatos de melhora na performance com uso de valores negativos, resultando em um algoritmo com comportamento de contra-tendencia. No trabalho de Even-Dar et al. (2006) é proposta uma modificação do *MWU* tradicional para incorporar o risco associado ao retorno de cada ativo. Nos resultados apresentados, a melhor performance é observada em algoritmos com parâmetro de aprendizado negativo.

Apesar das garantias teóricas para algoritmos Follow-the-winner de um lado, e o melhor comportamento de algoritmos contraintuitivos Follow-the-loser, nenhum estudo foi realizado para tentar melhor entender essa situação, e potencialmente utilizar o fenômeno causante para se obter ainda melhores resultados. Nesse trabalho, buscamos maneiras de aprender quando utilizar FTW e FTL para maximizar a performance dos algoritmos de *online learning*.

### 1.3

#### Contribuições da tese

Nossa contribuição principal é tentar entender melhor a questão acima e tentar incorporar este entendimento em algoritmos baseados em *online learning*. Mais concretamente:

- Propomos tornar o hiperparâmetro de aprendizado dos algoritmos de *online learning* em um parâmetro que pode ser otimizado e ter seu valor modificado dinamicamente de acordo com o aprendizado.
- Foi realizado um estudo da literatura sobre indicadores de tendencia Trend Following e Mean Reversal em séries financeiras e sua possível incorporação de forma explícita para escolha do parâmetro de aprendizado. Fazemos isso de forma offline, como um teste para verificar a pertinência dos indicadores estudados.
- Propomos um método de se realizar o aprendizado de forma online e de forma dinâmica do melhor regime (TF ou MR) para utilização em algoritmos de *online learning*. Este processo também é feito utilizando *online learning*. Em nossos experimentos, o método proposto supera o *benchmark* estabelecido UCRP com excesso de retorno de 36.76%, sendo portanto a contribuição mais relevante apresentada no trabalho.

Ressaltamos que o objetivo aqui não é propor um algoritmo prático a ser usado em produção. Tentamos isolar apenas a questão do relacionamento dos regimes TF/MR em online learning, e portanto omitimos alguns fatores importantes da execução prática sistemática, por exemplo custos de transação.

## 1.4

### Organização

O trabalho está organizado da seguinte forma: No Capítulo 2 definimos o problema de aprendizado online com *experts* e sua aplicação à seleção de portfólio online, bem como o algoritmo base que utilizaremos. No Capítulo ?? são introduzidos indicadores conhecidos na literatura para verificar a presença de regimes TF/MR em séries temporais. Em seguida, no Capítulo 3, são feitos experimentos utilizando os indicadores anteriormente explicitados para alteração do parâmetro de aprendizado do algoritmo base, com objetivo de capturar os diferentes regimes de mercado TF e MR. No Capítulo 4 são propostos dois novos algoritmos para resolver o problema de seleção de portfólio baseados na hipótese de TF e MR, porém sem necessidade do uso de indicadores externos. Experimentos são conduzidos. Por fim, no Capítulo 5 apresentam-se as considerações finais desta dissertação e possíveis direções para trabalhos futuros.

## 2

## Conceitos Preliminares

### 2.1

#### Aprendizado online com experts

Nessa seção apresentamos a versão mais simples, e importante, do modelo *online learning*, chamado de *aprendizado online com experts*. Em alto-nível, esse modelo captura a seguinte situação: Um agente deve tomar decisões sequenciais e para isso tem acesso a conselhos de  $m$  experts. Após cada escolha feita, o valor das sugestões dos experts é apresentado e isso pode ser levado em consideração para a escolha de quais experts utilizar no futuro.

Mais formalmente, o setup para previsão com o auxílio de experts é definido da seguinte forma.

- Consideramos a existência de  $m$  possíveis ações (experts).
- Os ganhos de uma ação  $i$  no tempo  $t$  são definidos pelo valor  $g_i^t$ .
- Para cada instante  $t$ , o algoritmo faz uma escolha representada pela distribuição de probabilidade  $b^t$  sobre as ações. A escolha pode usar a informação dos ganhos  $g^1, \dots, g^{t-1}$  nos períodos anteriores.
- Após a decisão, o vetor de ganhos  $g^t$  do tempo  $t$  é revelado.
- O objetivo do algoritmo é maximizar o ganho total  $\sum_t \langle b^t, g^t \rangle$ .

Tabela 2.1: Setup para algoritmos usando aprendizado com experts

O vetor  $b^t$  representa uma seleção fracional das  $m$  ações ou experts. Equivalentemente, podemos pensar que uma só ação ou expert é selecionado de forma aleatória usando a distribuição  $b^t$ , e a expressão  $\langle b^t, g^t \rangle$  representa o valor esperado do ganho no instante  $t$ .

Segundo Arora et al. (2012), uma das primeiras utilizações desse setup foi feita no algoritmo de Winnow proposto por Littlestone (1988). Nele, a técnica é utilizada para resolver um problema de programação linear para geração de classificadores lineares.

Um exemplo de uso do modelo de *online learning* é a técnica de boosting, descrita no artigo de Freund e Schapire (1999). Nele, o algoritmo proposto recebe um conjunto de classificadores simples e cria uma combinação com acurácia superior aos classificadores base. Outros exemplos incluem o uso para



otimização robusta proposto por Ben-Tal et al. em (2015) e o uso por Bärman et. al. em (2017) para otimização inversa.

Os algoritmos discutidos no contexto desse setup em geral são avaliados através da aproximação ao longo do tempo em relação à algum *benchmark* estabelecido. Infelizmente não é possível obter garantias não-triviais contra o algoritmo omnisciente que conheça todos os dados de antemão e que seja totalmente adaptativo. Portanto, o benchmark padrão é comparar o algoritmo contra o melhor algoritmo omnisciente mas que utiliza uma ação *fixa* durante todo o horizonte. A diferença entre o ganho desse benchmark e o ganho do algoritmo é denominado de *regret*:

$$\text{regret} = \max_{b \in \Delta^m} \sum_{t=1}^T \langle b, g^t \rangle - \sum_{t=1}^T \langle b^t, g^t \rangle,$$

onde  $\Delta^m$  é o conjunto de distribuições de probabilidades disponíveis para escolha em cada instante  $t$ .

O resultado mais clássico nessa área é que é possível se obter um regret de ordem  $\sqrt{T}$  (Helmbold et al. (1998)) (veja Teorema 2.1). Note que isso implica que a perda média (isto é, por instante de tempo) com relação ao benchmark é de ordem  $1/\sqrt{T}$ , ou seja, vai a zero conforme o horizonte aumenta. Ou seja, efetivamente tais algoritmos conseguem competir com a melhor ação fixa no horizonte.

A versão do modelo apresentada é um caso particular do modelo de *Online Linear Optimization* (OLO), introduzido por Kalai e Vempala (2003). Nele, o conjunto de escolhas do algoritmo é um conjunto convexo genérico  $K$ , ao invés do conjunto de distribuições de probabilidades em  $m$  objetos. O modelo de OLO, por sua vez, é um caso particular do framework mais geral *Online Convex Optimization* (OCO), descrito por Hazan et al. (2016). Nele, a função de ganho no tempo  $t$  é uma função concava  $f_t$ , ao invés da função linear  $\langle b^t, \cdot \rangle$ .

## 2.2

### Multiplicative Weights Update (MWU)

O *Multiplicative Weights Update* (MWU) é o algoritmo mais clássico no contexto de aprendizado com experts.

A ideia principal do MWU é a seguinte. O algoritmo inicia sem conhecimento prévio sobre os experts, então seleciona a distribuição uniforme sobre eles. Ao aprender quais experts estão com boa performance, o algoritmo aumenta o peso colocado nos bons experts de acordo com uma atualização multiplicativa. É importante ressaltar que a boa performance é medida com-

parativamente em relação aos outros experts. Então, mesmo que um expert tenha um ganho positivo, pode ter seu peso reduzido desde que outros experts tenham obtido melhor performance. Mais formalmente, o algoritmo *MWU* é descrito em Algorithm 1.

---

**Algorithm 1** Algoritmo MWU
 

---

- 1: **Inicialização:** Uma taxa de aprendizado  $\eta$  é fixada. Os pesos iniciais de cada expert são inicializados uniformemente  $\forall w, w_i^1 = 1$
- 2: **for**  $t = 1, 2, \dots, T$  **do**
- 3:   Os pesos nos experts são normalizados para se obter uma distribuição de probabilidade:

$$b_i^t = \frac{w_i^t}{\sum_j w_j^t}$$

- 4:   Para cada expert  $i$ , o peso é modificado de acordo com seus ganhos  $g_i^t$ :
- 5:

$$w_i^{t+1} = w_i^t e^{\eta \cdot g_i^t}.$$

- 6: **end for**
- 

Como provado no artigo de Helmbold et al. (1998), esse algoritmo possui garantia teórica de aproximar-se da performance do melhor expert *offline*, ou seja, observando toda a série de retornos. Apresentamos aqui uma garantia um pouco mais geral obtida por Arora et al. (2012).

**Teorema 2.1 (Arora et al. (2012))** *Assuma que  $g_i^t \in [-1, 1]$  e  $\eta \in [0, \frac{1}{2}]$ . O algoritmo MWU garante que depois de  $T$  períodos, para qualquer expert  $i$ , temos:*

$$\sum_{t=1}^T \langle b^t, g^t \rangle \geq \sum_t g_i^t - \eta T - \frac{\ln n}{\eta}.$$

Setando  $\eta = \frac{\sqrt{\ln n}}{\sqrt{T}}$  isso mostra que MWU tem regret  $2\sqrt{T \ln n}$ .

## 2.3

### Utilizando online learning para seleção de portfólios

Antes da publicação deste algoritmo, Cover (1991) descreveu um algoritmo específico para o problema de portfólio com garantia de performance semelhante ao Algoritmo 1, porém com complexidade de tempo exponencial no número de *experts*. O MWU, por outro lado, possui complexidade de tempo linear no número de ações.

Relembramos agora o problema de seleção de portfólios online. Lembre que um portfólio construído no universo de  $n$  ações é descrito por um vetor de alocações  $b = (b_1, b_2, \dots, b_n)$ , tal que  $b_i \geq 0$  e  $\sum_i b_i = 1$ . A performance de

ação  $i$  no tempo  $t$  pode ser medida através do retorno do seu preço atual  $p_i^t$  em relação ao seu preço anterior  $p_i^{t-1}$  através da fórmula:

$$x_i^t = \frac{p_i^t}{p_i^{t-1}} - 1$$

A performance de todas as ações consideradas, portanto, pode ser representada por um vetor de retornos  $x^t = (x_1^t, x_2^t, \dots, x_n^t)$ . Em um dado instante  $t$ , em posse de um vetor de pesos (alocações)  $b^t$  e um vetor de retornos  $x^t$ , podemos calcular o retorno do portfólio  $r^t$ .

$$r^t = \langle b^t, x^t \rangle = \sum_{i=1}^n b_i x_i$$

No problema de seleção de portfólios online, no início de cada período  $t$  a estratégia possui a sequência de retornos das ações  $x^1, x^2, \dots, x^{t-1}$ . A partir dessa informação, o portfólio para o tempo  $t$  é escolhido, os retornos de cada ação  $x^t$  são revelados, e o retorno do algoritmo  $r^t$  pode ser calculado. O objetivo do problema é maximizar o somatório dos retornos  $\sum_t r^t$ .

O problema é naturalmente adequado ao *setup* de aprendizado online com *experts*, e também à utilização do *MWU* como algoritmo. Para isso, basta atribuir cada ação  $i$  como um *expert*, utilizando seus retornos como vetor de ganhos:

$$\forall t \quad g^t = r^t$$

**Sinal do parâmetro de aprendizado e comportamento TF/MR.** O parâmetro de aprendizado  $\eta$  do *MWU* comumente assume um valor positivo. Nesse caso o algoritmo aumenta o peso de ações que tiveram um desempenho superior em comparação com o resto do universo, e reduz caso contrário. O modulo do parâmetro representa a intensidade da atualização feita: quanto maior, mais rápido é o aprendizado do algoritmo em relação aos retornos recentes. Pode-se dizer que quando  $\lim_{\eta \rightarrow \infty}$  o vetor de pesos  $b^t$  possui alocação concentrada unicamente no ativo que teve o maior retorno  $\max(x^{t-1})$  do período anterior. A estratégia de aumentar as alocações nas ações que tiveram melhor performance no passado é amplamente utilizada por administradores de carteiras no mundo, e é frequentemente chamada de *Trend Following (TF)*, pois assume implicitamente que ações com bons retornos continuaram a ter bons retornos no futuro.

Porém, caso o parâmetro de aprendizado  $\eta$  seja setado como um número negativo, o algoritmo passa a ter comportamento inverso e *reduzir* o peso das

ações que apresentaram bom retorno no período. A hipótese nesse caso é que o mercado tem um comportamento de reversão à média (*Mean Reverting (MR)*). Como mencionado na introdução, apesar de contraintuitivo, tal configuração de parâmetro de aprendizado negativo geralmente obtém performances superiores. Em nossos testes preliminares, de fato, observamos que existem períodos em que tal performances superiores ocorrem, como observado na Figura 2.1.

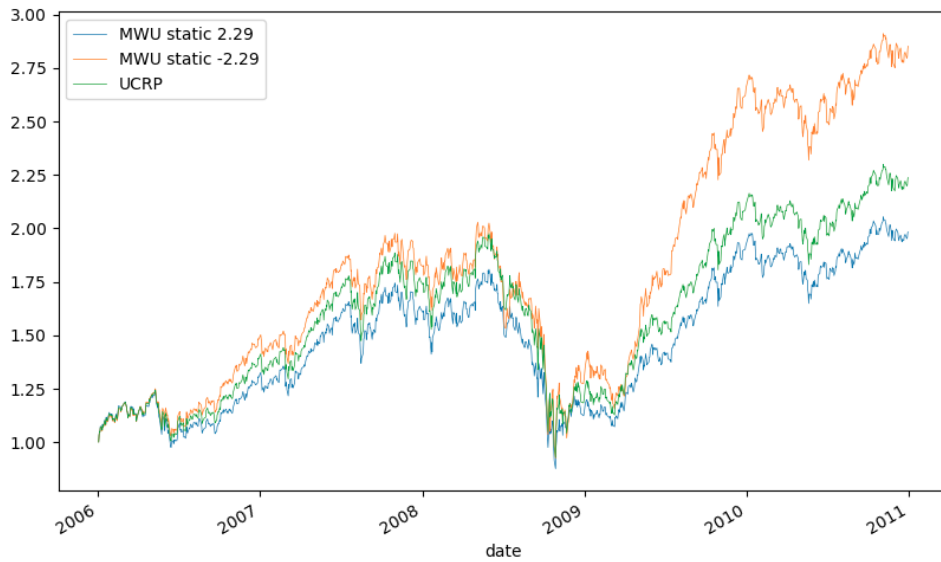


Figura 2.1: Exemplo de execução do algoritmo *MWU* em todas ações que fazem parte do índice IBOVESPA no período de 2006 até 2010, utilizando dados diários de preço de fechamento do mercado. O eixo  $y$  representa o retorno bruto da estratégia, que começa com  $r^0 = 1.0$ . O *benchmark* UCRP é definido na Seção 2.4. Diferentes valores do parâmetro de aprendizado tem performances variadas.

Da mesma maneira, na Figura 2.2 observamos que em algumas janelas a versão *TF* do algoritmo tem o resultado melhor.

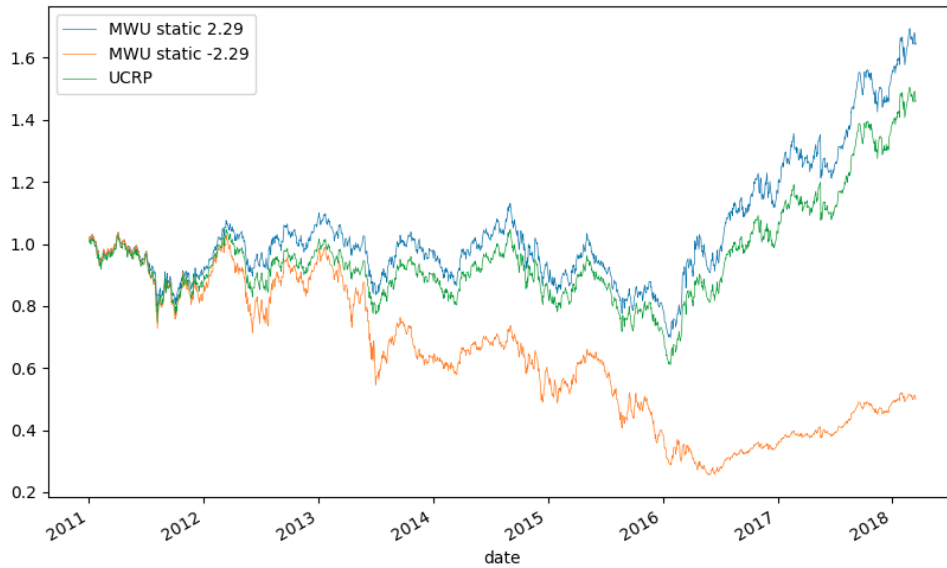


Figura 2.2: Exemplo de execução do algoritmo *MWU* em todas ações que fazem parte do índice IBOVESPA no período de 2011 até 2018, utilizando dados diários de preço de fechamento do mercado. O *benchmark* UCRP é definido na Seção 2.4

Entender melhor esse fenômeno é o foco deste trabalho.

## 2.4 Benchmark

Para avaliar a performance dos algoritmos propostos nesse trabalho, será considerado o retorno total composto da estratégia:

$$\prod_{i=1}^T (r^i + 1) \quad (2-1)$$

O valor inicial de riqueza utilizado será unitário, ou seja, caso nenhum capital seja alocado espera-se que ao final do tempo  $T$  o retorno total seja  $r^T = 1$ .

O *benchmark* escolhido como referência de comparação é o **Portfólio Uniformemente e Constantemente Rebalanceado (UCRP)**. Nele, a distribuição é uniforme entre todos os ativos, e os pesos são rebalanceados em cada período para manter as proporções. Dessa forma, os ganhos de uma ação são distribuídos entre todo o portfólio a cada período, assim como as perdas. Para um universo de  $n$  ações e  $T$  períodos, com vetor de pesos  $b_t$ , o

retorno total do UCRP pode ser expresso como

$$\text{retorno}(\text{UCRP}) = \prod_{t=1}^T \langle b^t, x^t \rangle$$

$$\forall t \quad b^t = \left( \frac{1}{m}, \frac{1}{m}, \dots, \frac{1}{m} \right)$$

O *UCRP* é um *benchmark* comumente utilizado em artigos da área de seleção de portfólios. Em geral, observa-se que o *UCRP* supera a composição original do índice na maioria dos casos, tornando-o um *baseline* desafiador. No artigo de Borodin et. al. (2004), por exemplo, este mesmo benchmark é utilizado (com o nome de *U-CBAL*). Borodin conclui que muitos algoritmos da literatura não são capazes de superar o UCRP, incluindo os portfólios universais de Cover (1991).

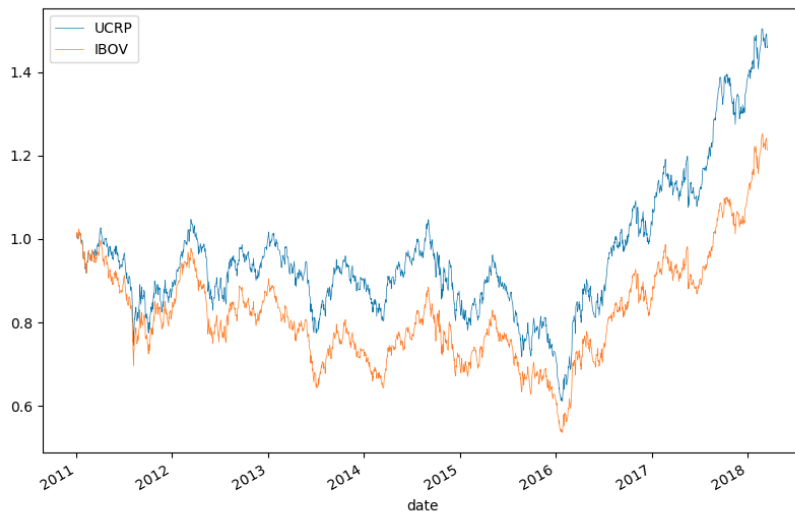


Figura 2.3: Comparação entre o *benchmark* utilizado e o índice IBOVESPA, no período de 01/2011 até 03/2018. O UCRP tem um retorno de 46.13%, enquanto o índice tem o retorno de 21.33%. A maior diversificação empregada pelo UCRP pode ser a justificativa da performance superior.

## 2.5

### Modificações ao algoritmo MWU

Nesse seção descrevemos algumas modificações que faremos ao algoritmo *MWU*.

#### 2.5.1

##### Fixed Share

O algoritmo MWU com a atualização tradicional descrito em Algorithm 1 possui a característica de tentar se aproximar da melhor ação em todo

horizonte. De certa forma, espera-se que exista um *expert* que se destaca em relação aos outros. Assim, com a evolução do tempo o peso será transferido em quase sua totalidade para esse melhor *expert*.

No caso de portfólios de ações, o que se observa na prática é uma mudança de regimes performance ao longo do tempo. Nesse caso, não queremos simplesmente prever o melhor ativo, mas sim a melhor sequência de ativos. Infelizmente o MWU tradicional não consegue tal garantia, como ilustrado no exemplo seguinte. Considere o caso em que temos 2 ativos e dividimos o tempo  $T$  em dois segmentos. No primeiro segmento o ativo #2 tem grandes perdas, enquanto no segundo segmento a situação se inverte e o ativo #1. Ao final do primeiro segmento, o peso do ativo #2 estará próximo de 0. Seria necessária uma recuperação rápida em seu peso para que suas previsões sejam consideradas no segundo segmento, quando sua performance é muito superior ao ativo #1. Porém, como a atualização do MWU é multiplicativa, pesos muito próximos a 0 levam muito tempo para serem incrementados a níveis razoáveis.

Para evitar tal comportamento, a ideia é introduzir uma parcela fixa de peso  $\beta$  para cada expert, como descrito por Herbster e Warmuth (1998) e chamado de algoritmo *Fixed Share*. A etapa de atualização multiplicativa dos pesos é feita de acordo com a equação à seguir:

$$p_i^{t+1} = \left(\frac{\beta}{n}\right) + (1 - \beta) \frac{p_i^t e^{\eta g_i^t}}{\sum_j p_j^t e^{\eta g_j^t}}$$

$$\beta = \frac{1}{T}.$$

Note que isso garante que os pesos  $p_i^t$  nunca vão a 0.

Além de ser uma heurística natural, Fixed Share possui fortes garantias: consegue competir com a melhor sequência offline de experts que tenha  $k$  trocas de experts (com garantia dependendo de  $k$ ), e também consegue competir com o melhor expert em cada um dos intervalos do jogo, ao invés de competir somente no horizonte inteiro; veja (Adamskiy2016) para mais detalhes.

## 2.5.2

### Sleeping Experts

Os experimentos realizados neste trabalho foram inicialmente conduzidos utilizando uma base de dados com todas as ações existentes na bolsa de valores brasileira (B3). Foram identificados alguns problemas com essa metodologia, descritos à seguir.

A negociação de muitas destas ações consideradas é feita com baixa frequência, representando um problema prático de liquidez para o rebalan-

ceamento de carteiras que contém esses ativos. Adicionalmente, diversas ações de empresas pequenas possuem base de dados com pouca consistência, pois muitos negócios são feitos por fora do mercado e não são considerados na evolução dos preços.

Por esses motivos, decidimos utilizar a composição do índice IBOVESPA como referência do universo de ações operáveis em um dado instante. O índice é muito utilizado na indústria como *benchmark* e via de regra possui as ações com maior liquidez do mercado.

Como consequência dessa escolha, surge a necessidade de mudar o universo do algoritmo dinamicamente, uma vez que a composição do índice é modificada ao longo do tempo.

Existe uma versão do algoritmo *MWU* descrita por Freund et al. (1997), chamada de *Sleeping Experts*, que possui a característica de ter diferentes universos de escolha ao longo do tempo. Em linhas gerais, o algoritmo mantém a atualização dos pesos para todos os *experts*, porém utiliza para o portfólio apenas aqueles permitidos em um instante  $t$ , renormalizando neste universo. É conhecido que tal modificação mantém a garantia de aproximação do algoritmo utilizado como base, veja Gaillard (2014) para mais detalhes.

### 2.5.3 Stop loss

Essa modificação é específica para a aplicação de otimização de portfólio considerada. O termo *stop loss* é comumente utilizado na literatura financeira para indicar ordens colocadas no mercado com o objetivo de limitar perdas. Em geral, esse valor de perda é expresso como um percentual do preço de entrada da posição. Consideramos agora o uso de um mecanismo de *stop loss* em nosso algoritmo.

Como exemplo, podemos considerar o caso da ação de uma empresa que teve uma perda de valor intensa durante um curto período de tempo, em comparação com as outras ações do universo considerado. O algoritmo *MWU* com um parâmetro de aprendizado positivo possui uma proteção natural à esse tipo de movimento, pois diminuiria gradualmente o peso dessa ação em seu portfólio. Por outro lado, esse algoritmo com um parâmetro de aprendizado negativo, que também será utilizado neste trabalho, aumentaria indefinidamente o peso dessa ação, sem nenhum mecanismo de parada.

Uma possível solução seria incluir na base apenas ações que não tiveram esses movimentos bruscos de queda ou de empresas que foram à falência. Entretanto, isto incluiria um viés de sobrevivência na base.

A solução empregada foi simular uma ordem de *stop loss* integrada ao



algoritmo. Antes de cada atualização dos pesos, o algoritmo verifica o retorno acumulado em uma janela de  $d$  dias, e, caso seja menor que um valor *stop* escolhido a ação é removida do universo e seu peso distribuído entre os outros componentes do portfólio. Tal ação é desconsiderada para a montagem dos portfólios subsequentes.

Para condução dos experimentos, os parâmetros foram escolhidos com os valores de  $d = 22$  e  $stop = 60\%$ . Não achamos válido desperdiçar um pedaço da base para otimização desses parâmetros, portanto escolhemos valores considerados razoáveis baseados na observação do mercado.

## 2.6

### Algoritmo Base Final

O código base que utilizaremos nos experimentos consiste do *MWU* com as modificações descritas acima; chamamos esse algoritmo de *MFSS* (Multiplicative Fixed Share Sleeping Experts Stop loss).

---

**Algorithm 2** Algoritmo MFSS

---

- 1: **Inicialização:** Uma taxa de aprendizado  $\eta$  é fixada. Os pesos iniciais de cada expert são inicializados uniformemente  $\forall w, w_i^1 = 1$
- 2: **Inicialização:** Um parcela fixa  $\beta$  é fixada.
- 3: **Inicialização:** Um valor de *stop* e uma janela *stop\_window* são fixados.
- 4: **for**  $t = 1, 2, \dots, T$  **do**
- 5:   Verifica stop:

$\forall i$  **Se**  $\prod_{j=(t-stop\_window)}^{t-1} (1 + r_i^j) \leq stop$ , **então** adiciona expert  $i$  ao conjunto *stop\_set*

- 6:    $IBOV =$  Obtém conjunto de ações do índice IBOVSPA no instante  $t$
- 7:   Copia o vetor de pesos para vetor de alocações  $b_i^t = w_i^t$
- 8:   Filtra ações

$\forall i$  **Se**  $i \notin IBOV$  **ou**  $i \in stop\_set$ , **então**  $b_i^t = 0.0$

- 9:   Obtém o vetor de alocações utilizados no instante  $t$  através de normalização

$$b_i^t = \frac{b_i^t}{\sum_j b_j^t}$$

- 10:   Ao fim do período  $t$ , o retorno do portfólio no instante  $t$  é calculado

$$r^t = \langle b^t, g^t \rangle = \sum_{i=1}^n b_i g_i$$

- 11:   Para cada *expert*  $i$ , o peso é modificado de acordo com seus ganhos  $g_i^t$

$$w_i^{t+1} = \left( \frac{\beta}{n} \right) + (1 - \beta) \frac{w_i^t e^{\eta \cdot g_i^t}}{\sum_j w_j^t e^{\eta \cdot g_j^t}}$$

- 12: **end for**
-

## 3

### Aprendizado com indicadores

Para estudar a possível classificação de uma série financeira em *Trend Following (TF)* ou *Mean Reverting (MR)* revisamos a literatura existente em busca de indicadores. Posteriormente, fazemos uma tentativa de aprendizado do hiperparâmetro com suporte da classificação de regimes obtida através das medidas estudadas.

#### 3.1

#### Indicadores de tendencia em séries financeiras

##### 3.1.1

##### Expoente de Hurst

A estatística tradicional falha ao analisar o mercado de ações pois assume que o sistema é totalmente aleatório e portanto somente probabilidades podem auxiliar no entendimento do processo. Por outro lado, Edgar E. Peters (1994) em seu trabalho argumenta que o mercado é um sistema que pode ser descrito através de uma combinação de aleatoriedade e determinismo. Diversos outros trabalhos sugerem que o mercado possui de fato um componente determinístico, contrariando a hipótese do mercado eficiente. Peters sugere utilizar a teoria dos fractais ao analisar o comportamento do mercado de ações.

O **Expoente de Hurst** ( $H$ ) é uma metodologia não paramétrica para análise de estrutura fractal em series temporais, sugerida originalmente por H. E. Hurst (1951) em seu trabalho . Inicialmente, sua teoria foi aplicada ao estudo da modelagem de reservatórios de água. Com o tempo, foi estendida ao estudo da memória em series financeiras.

Hurst estudou o trabalho de Einstein sobre o movimento browniano. Nele, o autor propõe que a distancia coberta por uma partícula aleatória  $R$  aumenta proporcionalmente a raiz quadrada do tempo transcorrido  $T$ , ou seja:

$$R = \Theta(T^{0.5}) \quad (3-1)$$

A ideia de Hurst foi fazer uma análise da série temporal objeto para verificar com qual razão a distância  $R$  é percorrida em relação ao tempo transcorrido. Dessa forma, seria possível comparar essa razão encontrada com o coeficiente do movimento browniano e **classificar** a natureza da série em

**persistente, não persistente, ou aleatória.** A análise de reescala (análise R/S), utilizada para encontrar o expoente, é descrita no Algorithm 3.

---

**Algorithm 3** Análise R/S
 

---

- 1: Uma serie  $\mathbf{x} = [x_1, \dots, x_n]$  é recebida.
  - 2:  $\mathbf{y} = \mathbf{x} - \text{media}(\mathbf{x})$  ▷ A série é normalizada
  - 3:  $y[] = \text{cumsum}(\mathbf{y})$  ▷ A soma acumulada é feita
  - 4:  $R = \max(\mathbf{y}) - \min(\mathbf{y})$
  - 5:  $S = \text{std}(\mathbf{y})$
  - 6: return  $\frac{R}{S}$
- 

O algoritmo para encontrar  $H$ , de acordo com Peters e Peters (1994), utiliza a análise R/S em múltiplos fragmentos da série para maior precisão. Para isso, recebe um vetor de tamanhos de fragmentação **tamanhos** =  $[t_1, \dots, t_n]$ . Para cada elemento desse vetor, o algoritmo divide a série original em múltiplas séries de tamanho  $t_i$  e faz a análise R/S para cada uma delas. O coeficiente associado à um dado tamanho  $t_i$  é a média do resultado da análise das séries individuais.

Ao final dessa iteração, o algoritmo reúne o vetor de coeficientes encontrado e faz uma regressão linear simples com o vetor de tamanhos. O resultado  $H$  é a inclinação da curva encontrada.

---

**Algorithm 4** Expoente de Hurst
 

---

- 1: Uma serie  $x[] = [x_1, \dots, x_n]$  é recebida.
  - 2: Vetor de tamanhos é inicializado: **tamanhos** =  $[t_1, \dots, t_n]$
  - 3: **for**  $t_i$  em **tamanhos** **do**
  - 4:   Um vetor de vetores **x\_split** é criado com a série original partida em  $\frac{n}{t_i}$  séries.
  - 5:   Faz análise R/S para cada uma das series em **x\_split** e armazena em um vetor **temp**.
  - 6:    $\text{coeficientes}[t_i] = \text{media}(\text{temp})$
  - 7: **end for**
  - 8: Faz regressão linear com  $x = \text{tamanhos}$  e  $y = \text{coeficientes}$ . Retorna a inclinação da curva encontrada como  $H$ .
- 

Os tamanhos de fragmentação podem variar, porém é esperado que o coeficiente  $H$  encontrado não seja muito sensível à isso. Como regra geral, escolhe-se normalmente valores  $t_i > 10$  e, naturalmente,  $t_i \leq \frac{n}{2}$ .

O conceito de persistência é diretamente ligado ao comportamento de  $TF$  descrito na seção anterior, enquanto a anti-persistência está ligada ao

comportamento  $MR$  de uma série. Portanto, o coeficiente  $H$  encontrado pode ser interpretado da seguinte forma:

$$\begin{cases} H > 0.5, & \text{Trend Following} \\ H < 0.5, & \text{Mean Reverting} \\ H = 0.5, & \text{sem memória} \end{cases} \quad (3-2)$$

Calculamos o coeficiente  $H$  em toda série disponível, que está ilustrado no Figura 3.1. Utilizamos os retornos semanais de cada ação para evitar capturar ruídos de curto prazo existentes na série. As ações possuem bases com data de início variado, onde as mais antigas começam em 2006. Exceto as ações que não existem mais, todas terminam em 2018. Nota-se que a distribuição dos coeficientes se assemelha a uma curva normal, com ligeiramente mais ações com característica MR que TF.

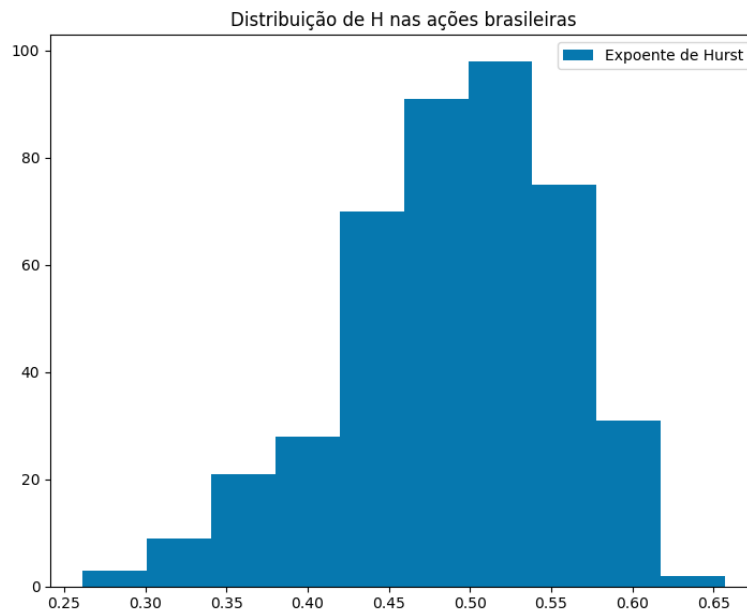


Figura 3.1: Histograma da distribuição do expoente de Hurst das ações negociadas na bolsa brasileira.

### 3.1.2 Autocorrelação

Autocorrelação é a correlação de uma série com sua cópia atrasada de  $k$  períodos. Frequentemente, é a medida escolhida em trabalhos de finanças para verificar auto dependência nos retornos de um ativo.

De acordo com o livro de Tsay (2005), para uma série  $R = r_1, r_2, \dots, r_T$ , com média amostral  $\bar{r}$ , a autocorrelação amostral de grau  $k$  pode ser calculada como:

$$AC(R) = \frac{\sum_{t=k+1}^T (r_t - \bar{r})(r_{t-k} - \bar{r})}{\sum_{t=1}^T (r_t - \bar{r})^2}.$$

Assim como a correlação, seu valor  $AC$  é definido no intervalo  $-1 \leq AC \leq 1$ . Quanto mais negativo, mais inversamente dependentes são os elementos da série. Quanto mais positivo, mais diretamente dependentes são. Essa dependência encontrada pode ser traduzida para nosso método de classificação da seguinte forma.

$$\begin{cases} AC > 0.0, & \text{Trend Following} \\ AC < 0.0, & \text{Mean Reverting} \\ AC = 0.0, & \text{sem memória} \end{cases} \quad (3-3)$$

Em nosso trabalho, escolhemos utilizar a autocorrelação de grau  $k = 1$  nos retornos semanais. O grau 1 foi escolhido por capturar simultaneamente dependências de curto e longo prazo.

Calculamos tal autocorrelação em toda série disponível, que está ilustrado no Figura 3.2. A distribuição dos coeficientes é bastante semelhante a observada na figura 3.1, com um número ligeiramente maior de ações com comportamento MR.

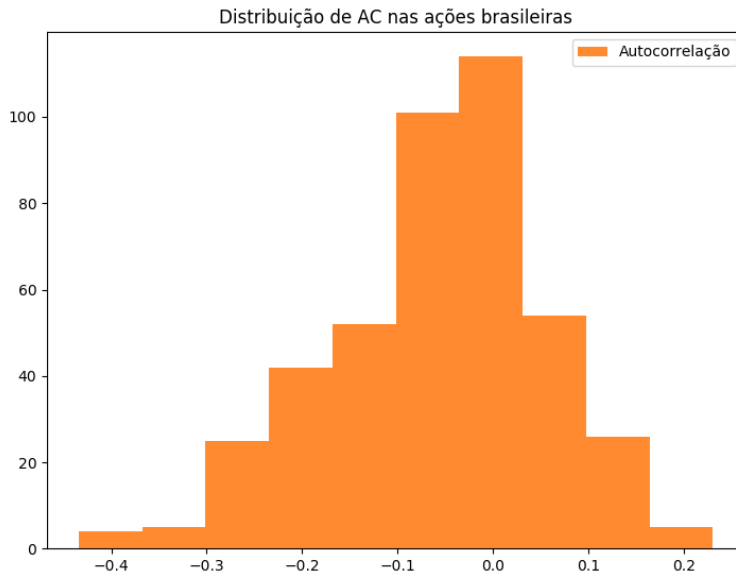


Figura 3.2: Distribuição da autocorrelação de grau  $k = 1$  das ações negociadas na bolsa brasileira.

### 3.2

#### Aprendizado offline com indicadores

Nessa seção procuramos avaliar se a utilização de informação sobre regimes TF e MR pode melhorar o desempenho de algoritmos de *online learning*.

para construção de portfólios. Mais precisamente, utilizamos os indicadores de tendencia Hurst (H) e autocorrelação (AC) estudados na Seção 3.1 para escolher o sinal da taxa de aprendizado  $\eta$  no algoritmo MFSS descrito na Seção 2.6. Lembre da discussão da Seção 2.3 que quando o parâmetro de aprendizado  $\eta$  é setado com valor positivo, o algoritmo aumenta o peso de ações que tiveram boa performance (recomendado em mercados TF), e quando o parâmetro de aprendizado  $\eta$  é setado com valor negativo o algoritmo reduz o peso de ações que tiveram boa performance (recomendado em mercados MR).

Portanto, exploramos a idéia de transformar o hiperparâmetro de aprendizado  $\eta$ , que tradicionalmente é fixo, em um parâmetro otimizável e dinâmico. Fazemos algumas tentativas diferentes de escolha desse parâmetro baseadas nos indicadores calculados sobre as séries financeiras.

Note que todos os testes do presente capítulo possuem um forte viés de *lookahead*, uma vez que os indicadores são calculados considerando toda a série financeira. Para replicar o uso dessa ideia na prática, os cálculos devem ser feitos de forma online (veja o fim do capítulo para discussão).

### 3.2.1

#### Setup Experimental

A base utilizada é proprietária e contempla todas as ações disponíveis historicamente na bolsa de valores do Brasil, B3 (anteriormente BOVESPA). Utilizamos granularidade diária para tomada de decisão. Os dados foram divididos em parte *In-sample* (de 01/2006 até 12/2010), utilizado para otimização de parâmetros, e *Out-of-sample* (de 01/2011 até 03/2018) para validação experimental.

#### 3.2.1.1

##### Seleção do módulo do hiperparâmetro de aprendizado

Os algoritmos propostos nesse capítulo utilizam os indicadores apenas para modificar o sinal dos parâmetros de aprendizado no algoritmo MFSS. O módulo de  $\eta$ , portanto, deve ser escolhido independente dos indicadores.

Para isso, utilizamos um período *In-Sample* de 01/2006 até 12/2010 para otimizar o módulo de  $\eta$ . Executamos o algoritmo MFSS com  $\eta \in (\pm 0.05, \pm 0.5, \pm 1.0, \pm 2.0, \pm 5.0, \pm 10.0)$ . Após, calculamos a média do retornos diários entre os pares de estratégia com módulo de  $\eta$  iguais. O  $\eta$  escolhido é aquele cujo o par de estratégias MFSS obteve o melhor retorno acumulado em todo período considerado. Em nossos testes,  $\eta = \pm 2$  alcançou a melhor performance.

### 3.2.1.2

#### Benchmark

Para avaliar a performance dos algoritmos propostos nesse capítulo, será utilizado o UCRP, como descrito na seção 2.4.

### 3.2.2

#### Algoritmo com eta estático e distinto para cada ação

Nesse experimento consideramos o algoritmo *MFSS* com parâmetro diferente  $\eta_i$  para cada ativo, cujo sinal será determinado utilizando os indicadores  $H$  e  $AC$ , e será mantido fixo durante todo o horizonte. Mais precisamente, para a ação  $i$  escolhemos o parâmetro de aprendizado  $\eta_i$  do algoritmo da seguinte forma (dependendo do indicador  $H$  ou  $AC$  escolhido):

$$\begin{cases} \eta_i = 2.0, & AC_i \geq 0.0 \\ \eta_i = -2.0, & AC_i < 0.0 \end{cases} \quad (3-4)$$

$$\begin{cases} \eta_i = 2.0, & H_i \geq 0.5 \\ \eta_i = -2.0, & H_i < 0.5 \end{cases} \quad (3-5)$$

Ter hiperparâmetros de aprendizados diferentes para os ativos permite que ativos com características diferentes tenham seus pesos atualizados de forma distinta.

Realizamos experimentos utilizando esses hiperparâmetros de aprendizados no período de 01/2011 até 03/2018. Os retornos totais compostos do algoritmo utilizando os indicadores  $H$  e  $AC$  e do *baseline UCRP* são apresentados na Figura 3.3.



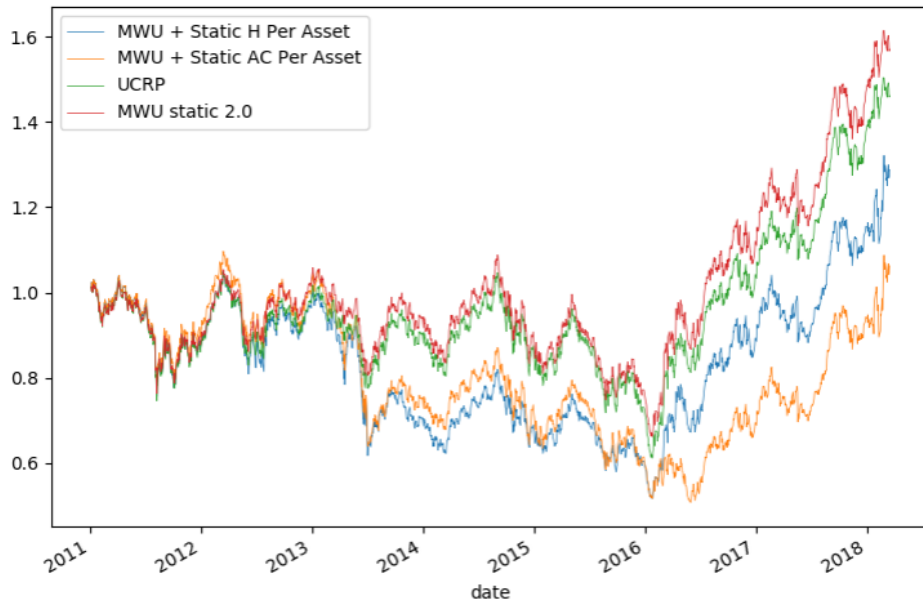


Figura 3.3: Resultados da estratégia utilizando um  $\eta$  distinto para cada ação, de acordo com a análise dos indicadores calculados de antemão. O eixo  $y$  representa o retorno bruto da estratégia, com  $r^0 = 1.0$ .

O portfólio *UCRP* acumulou um retorno de 46.13%, e o MWU com  $\eta = 2.0$  estático teve um retorno de 57.02%. A estratégia utilizando  $H$  retornou 28.81%, e utilizando  $AC$  entregou 6.23%. A partir destes resultados, podemos concluir que as medidas utilizadas dessa forma não agregam valor à estratégia.

### 3.2.3

#### Algoritmo com $\eta$ dinâmico e distinto para cada ação

Analizando os resultados da Seção 3.2.2, verificamos que os indicadores não foram eficazes em conjunto com o algoritmo *MFSS*. Uma das explicações possíveis é de que existem mudanças no regime das ações individuais, alternando entre períodos TF e MR.

Para tentar capturar essas mudanças de regime, testamos a segmentação da janela de testes em períodos TF e MR, novamente utilizando os indicadores estudados.

#### 3.2.3.1

##### Divisão de série em janelas TF e MR

Para particionar as séries individuais em janelas TF e MR, desenvolvemos um algoritmo guloso descrito no Algoritmo 5.

**Algorithm 5** Segmentação ótima TF/MR

---

```

1: Uma serie  $x[] = [x_1, \dots, x_n]$  é recebida para classificação.
2: Um vetor de tamanhos possíveis de janela é inicializado:  $tamanhos[] = [t_1, \dots, t_n]$ 
3: for  $t_i$  em  $tamanhos[]$  do
4:   Divide a série  $x$  em subséries de tamanho  $\frac{x}{t_i}$  e armazena em  $x\_split[]$ .
5:   for  $x\_split_j$  em  $x\_split[]$  do
6:      $split\_sig[j] = \text{calcula\_significancia}(x\_split_j)$ 
7:   end for
8:    $significancia[i] = \text{média}(split\_sig[j])$ 
9: end for
10:  $idxmax = \text{idxmax}(significancia[])$ 
11: return  $tamanhos[idxmax]$ 

```

---

A função *calcula\_significancia* itera sobre as subséries e calcula o indicador escolhido ( $H$  ou  $AC$ ). A partir deste vetor de indicadores, calcula o absoluto da diferença entre o valor e o *baseline* para cada medida ( $H = 0.5$  e  $AC = 0.0$ ). Este valor chamamos de significância, pois representa a força do sinal da medida.

A ideia do algoritmo descrito é verificar qual tamanho de janela divide melhor a série de retornos de uma ação em períodos  $TF$  ou  $MR$ , maximizando a significância definida. A saída do algoritmo pode entregar não somente o tamanho de janela ideal (como descrito em Algorithm 5), mas também o indicador calculado para cada período.

**3.2.3.2****Experimento com eta dinâmico**

O algoritmo em questão utiliza a mesma regra definida nas equações (3-4) e (3-5) para escolha do parâmetro de aprendizado. Entretanto, nesse experimento o valor de  $\eta_i$  é alterado a cada mudança de regime, de acordo com a saída do Algorithm 5.

O portfólio utilizando o indicador  $H$  dinâmico por ativo acumulou um retorno de 25.62%, enquanto a versão utilizando  $AC$  retornou 57.83%. O benchmark *UCRP* acumula um retorno de 46.13% e o MWU com  $\eta = 2.0$  estático teve retorno de 57.02%.

Para um teste com um forte viés de *look-ahead*, essa melhora da versão utilizando a autocorrelação não parece ser suficiente para concluir que a medida tem pertinência neste caso. Outros testes devem ser conduzidos para verificar a hipótese.

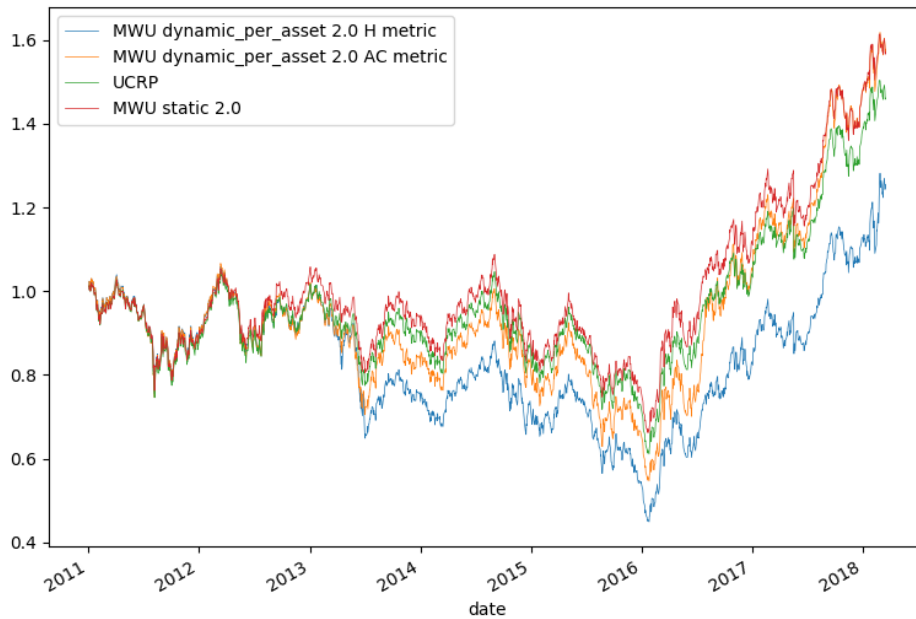


Figura 3.4: Resultados da estratégia utilizando um  $\eta$  distinto e dinâmico para cada ação, de acordo com a análise dos indicadores calculados de antemão.

### 3.2.4

#### Algoritmo com $\eta$ dinâmico para todo universo

Nesta seção, consideramos a possibilidade do mercado apresentar mudanças de regimes globais. Pode-se argumentar que existem ruídos ao observar as séries individuais, mas ao analisar o comportamento como um todo, conseguimos capturar o padrão geral.

Dessa forma, podemos analisar o comportamento da série de preços do mercado em conjunto, e definir em cada instante qual o regime atual —  $TF$  ou  $MR$ . Para executar tal tarefa, escolhemos utilizar o mesmo algoritmo descrito na Seção 3.2.2 aplicado a série de retornos do  $UCRP$ . Ou seja, estamos usando os resultados do  $UCRP$  como um índice que captura a performance do mercado como um todo em um único retorno por instante de tempo.

Novamente, fazemos dois testes - um para cada indicador estudado da literatura. O portfólio  $UCRP$  acumulou um retorno de 46.13% no período. A estratégia utilizando  $H$  retornou 30.45% e utilizando  $AC$  entregou 45.25%. A partir destes resultados, podemos concluir que as medidas utilizadas dessa forma não agregam valor à estratégia.

## 3.3

### Conclusão

Os resultados dos algoritmos propostos neste capítulo não foram capazes de superar claramente o *benchmark* estabelecido. Uma possível razão para isso

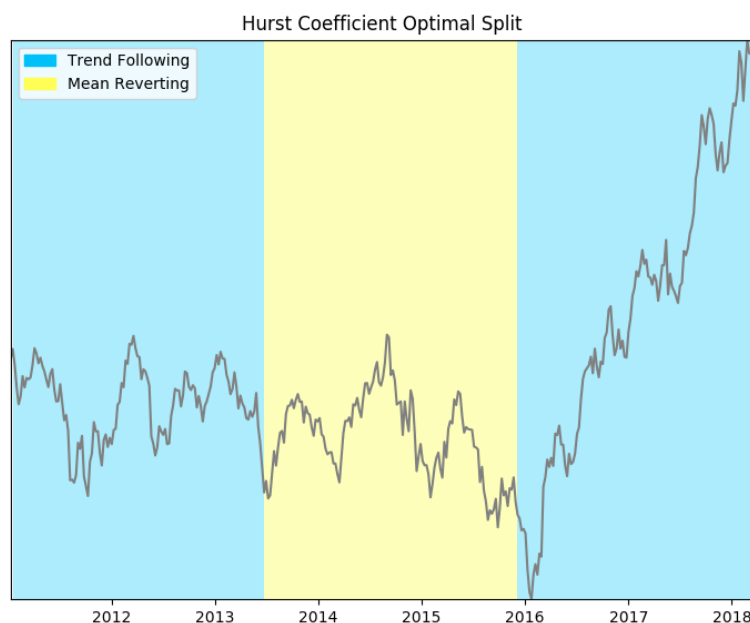


Figura 3.5: Divisão de regimes encontrada pelo algoritmo guloso para a série de preços do UCRP, de acordo com o indicador H.

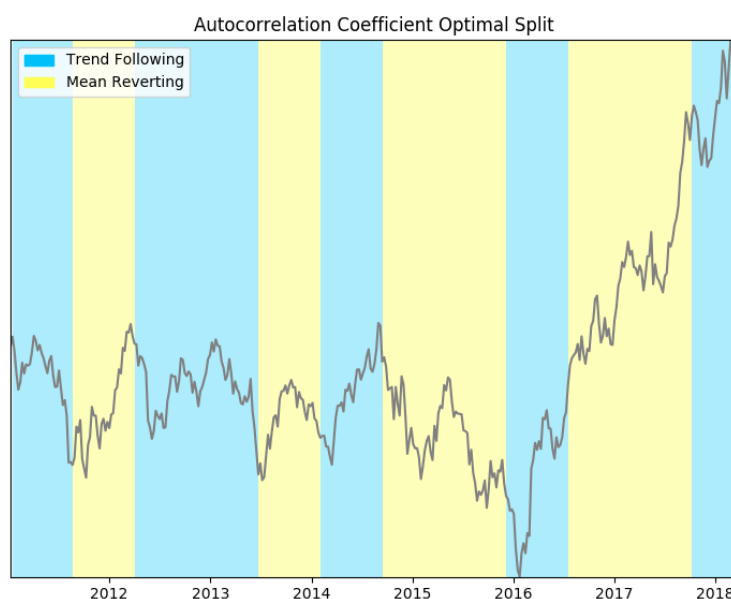


Figura 3.6: Divisão de regimes encontrada pelo algoritmo guloso para a série de preços do UCRP, de acordo com o indicador AC.

é de que os coeficientes não sejam capazes de capturar os comportamentos TF e MR, uma vez que a própria definição desses regimes não é clara. Por exemplo, é possível notar que as partições obtidas nas Figuras 3.5 e 3.6 são muito distintas, isto é, não parece existir um consenso do que é definido como TF e MR.

Outra possível justificativa é o fato de não utilizarmos a força dos

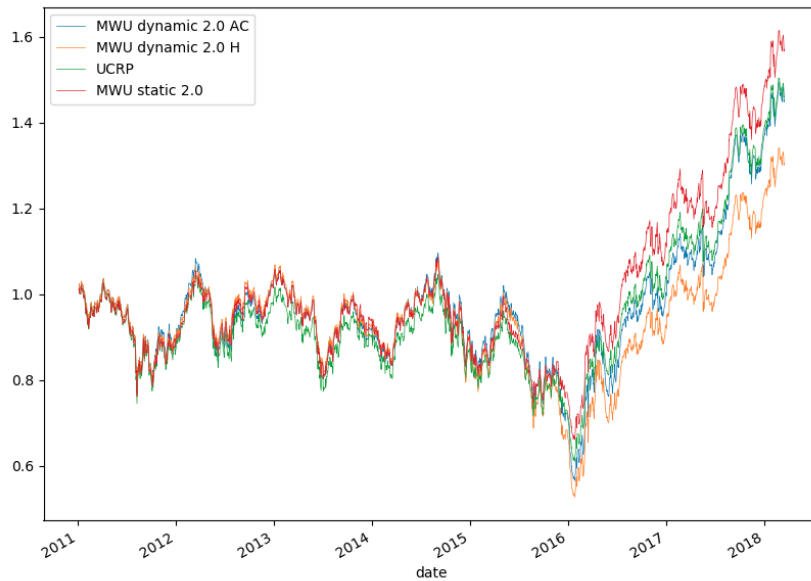


Figura 3.7: Resultados da estratégia utilizando um  $\eta$  dinâmico para todo mercado, de acordo com a análise dos indicadores calculados de antemão sob a série de retornos do UCRP.

indicadores para calibrar o algoritmo. Pode ser que um dado ativo passe por momentos com regime muito ou pouco seguidor de tendência ou de reversão à média, e isso não é levado em consideração.

Por conta dos resultados desencorajantes mesmo com o cálculo *offline* dos indicadores  $H$  e  $AC$ , decidimos não tentar estima-los e fazer o particionamento de janelas de forma *online*.

## 4

### Aprendizado online com meta-algoritmos

Utilizar  $H$  e  $AC$  diretamente não entregou bons resultados, possivelmente, como discutimos, pois não existe consenso de como medir exatamente regimes TF e MR. Entretanto, o mais importante é notar que só precisamos decidir se um período é TF ou MR para fins do algoritmo de seleção de portfólios. Isto é, podemos tentar aprender diretamente o sinal da taxa de aprendizado que resulta na melhor performance para o algoritmo.

Um tópico de pesquisa existente no estudo do problema de seleção de portfólios online é o chamado *meta-learning* ou meta-algoritmos (MA). Um MA recebe um conjunto de algoritmos base e tem como objetivo combiná-los da melhor forma em cada instante de tempo. De certa forma, o  $MWU$  aplicado ao mercado de ações descrito na Seção 2.3 pode ser considerado um MA atuando sobre um conjunto de estratégias base simples - cada uma delas sendo um *buy-and-hold* de um única ação.

Entretanto, podemos utilizar o próprio  $MWU$  para atuar sobre estratégias base do tipo  $MFSS$ , porém que se diferenciam através do parâmetro de aprendizado  $\eta$  escolhido em cada uma delas. A ideia é, portanto, utilizar duas camadas de aprendizado multiplicativo: a camada superior, chamado de  $MA$  ou master, que escolhe qual algoritmo da camada inferior utilizar, e a camada inferior contendo instâncias do algoritmo  $MFSS$  com diferentes parâmetros de aprendizado. Note que isso elimina a necessidade de análise prévia do comportamento  $TF$  ou  $MR$  dos ativos estudados, e nenhum indicador, como  $H$  ou  $AC$ , é utilizado no processo.

Dessa forma, novamente propomos a tentativa de aprendizado do hiper-parâmetro de aprendizado  $\eta$  do  $MFSS$ . Neste capítulo, entretanto, fazemos isso de forma online e sem utilização de indicador ou modelo explícito.

#### 4.1

##### Benchmark

O benchmark escolhido nesta seção também é o **Portfólio Uniformemente e Constantemente Rebalanceado (UCRP)** descrito na Seção 2.4. Da mesma forma como feito anteriormente, avaliamos a performance do algoritmo através do retorno total acumulado da estratégia.

Por conta da complexidade de tempo elevada de um dos algoritmos propostos neste capítulo, os testes foram executados em subconjuntos aleatórios de quatro ações. Naturalmente, o *UCRP* usado como benchmark em cada um dos testes considera apenas as ações escolhidas para o universo. Para não perder precisão e utilizar o máximo da base possível, foram repetidos diversos experimentos com combinações diferentes de quatro ativos. A avaliação final se dá por meio da análise da distribuição de resultados do *UCRP* vis a vis a performance do algoritmo proposto.

## 4.2

### Aprendizado com parâmetros distintos para cada ação

Nesta versão do algoritmo consideramos a possibilidade de cada ativo se comportar de maneira distinta no mesmo período de tempo. Dessa forma, na camada inferior, precisamos considerar *MFSS* com todas as possíveis combinações de sinal do parâmetro de aprendizado para cada um dos quatro ativos escolhidos para o universo.

#### 4.2.1

##### Um meta-algoritmo para partição de regimes *TF* e *MR* distintos por ativo

Para um universo de  $n$  ações, o algoritmo recebe um parâmetro de aprendizado base  $\eta_{base}$  e outro parâmetro de aprendizado  $\eta_{master}$ . Considera-se que a cada ativo do universo pode ser atribuído um parâmetro de aprendizado de  $\pm\eta_{base}$ , dependendo da sua natureza *TF* ou *MR*. Sendo assim, calcula todas as  $2^n$  possíveis combinações de parâmetros entre os ativos, e cria um algoritmo *MFSS* para cada uma delas. Estes são os *experts* da camada inferior utilizados pelo meta-algoritmo.

O *MA* master da camada superior funciona da mesma forma que seus *MFSS* base, porém utiliza um parâmetro de aprendizado distinto  $\eta_{master}$ . A cada período, o algoritmo executa uma atualização multiplicativa e aumenta ou diminui o peso de cada sub-estratégia base de acordo com seu retorno. O valor de  $\eta_{master}$  deve ser positivo, de forma a aumentar a alocação nas sub-estratégias com mais aderência ao atual comportamento do mercado.

O método proposto, definido no Algoritmo 6, foi batizado de *Expert of Experts* em função da sua dupla utilização do aprendizado com experts. Veja Figura 4.1 para uma ilustração do funcionamento da atualização multiplicativa em duas camadas proposta.

#### 4.2.2

**Algorithm 6** Experts of Experts (EoE)

- 
- 1: **Inicialização:** Em posse da taxa de aprendizado  $\eta_{base}$ , são inicializados  $m = 2^n$  MFSS base
  - 2: **Inicialização:** Uma taxa de aprendizado  $\eta_{master}$  é fixada. Os pesos iniciais de cada MFSS base são inicializados uniformemente  $\forall w, w_i^1 = 1$
  - 3: **for**  $t = 1, 2, \dots, T$  **do**
  - 4:   **for**  $i$  de 1 até  $m$  **do**
  - 5:      $b_i^t =$  Vetor de alocações do  $i$ -ésimo MFSS base
  - 6:   **end for**
  - 7:   Obtém o vetor de alocações final utilizado no instante  $t$  através de normalização dos pesos e combinação dos vetores de alocação dos MFSS base

$$c_i^t = \frac{w_i^t}{\sum_j w_j^t}$$

$$a_i^t = \sum_{i=1}^m c_i^t \cdot b_i^t$$

- 8:   Os retornos do tempo  $t$  são revelados. O retorno do portfólio no instante  $t$  é calculado utilizando os retornos individuais  $x_i$  de cada ação e o vetor de alocações final  $a^t$  escolhido pelo algoritmo

$$r_{master}^t = \langle a^t, x^t \rangle = \sum_{i=1}^n a_i x_i$$

- 9:   **for**  $i$  de 1 até  $m$  **do**
- 10:      $r_i^t =$  Retorno individual do  $i$ -ésimo MFSS base
- 11:     Faz atualização multiplicativa do MFSS
- 12:   **end for**
- 13:   O peso de cada MFSS base  $i$  é modificado de acordo com seus retornos individuais  $r_i^t$

$$w_i^{t+1} = w_i^t e^{\eta_{master} \cdot r_i^t}$$

- 14: **end for**
- 

**Experimento**

Novamente utilizamos granularidade diária para tomada de decisão, e os dados foram divididos em parte *In-sample* (de 01/2006 até 12/2010), utilizado para otimização de parâmetros, e *Out-of-sample* (de 01/2011 até 03/2018) para validação experimental.

Na fase de otimização de parâmetros, definimos uma região discreta de possíveis valores para os parâmetros  $\eta_{base}$  e  $\eta_{master}$  e fazemos uma execução para cada combinação possível dos parâmetros. Importante ressaltar que foi feita uma otimização independente para cada conjunto de ações sorteado, portanto,



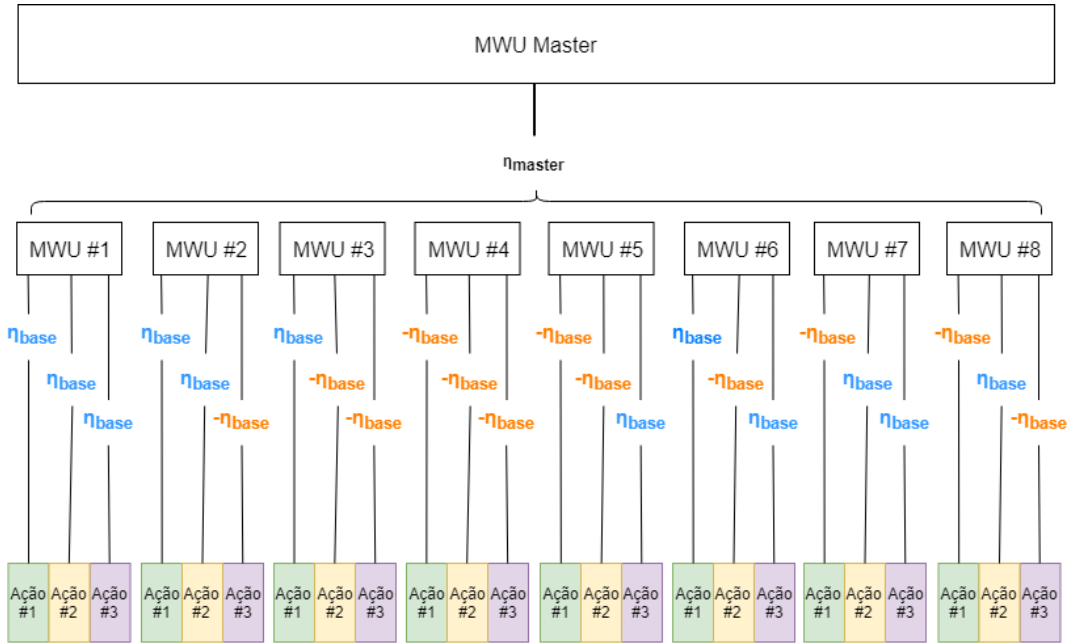


Figura 4.1: Exemplo de fluxo de atualização multiplicativa do meta-algoritmo com  $n = 3$ .

cada experimento tem parâmetros ótimos distintos. Baseado no Teorema 2.1 a região de escolha do parâmetro  $\eta_{base}$  e  $\eta_{master}$  é em torno do valor  $\frac{\sqrt{\ln n}}{\sqrt{T}}$ . Utilizamos essa heurística como base, porém observamos que o valor calculado não pareceu alto o suficiente para observar diferenças entre nosso algoritmo e o *benchmark*. Por ser muito pequeno, a atualização multiplicativa acaba por não alterar muito o balanceamento entre os ativos e o algoritmo se reduz ao *UCRP* inicial. Por essa razão, a região escolhida para a otimização foi feita com valores arbitrários, múltiplos do valor calculado original, bastante maiores que a indicação da heurística. Utilizamos a região  $(0.34, 3.35, 6.71, 13.41)$  para escolha do  $\eta_{base}$ , e a região  $(0.47, 4.74, 9.48, 18.97)$  para escolha do  $\eta_{master}$ .

Depois dessa rodada de otimização, passamos a validação do algoritmo. Foram conduzidas 100 rodadas de execução do algoritmo e seu *benchmark*, com combinações de quatro ativos diferentes. Avaliamos os resultados calculando a estatística *excedente retorno*:

$$\text{excedente retorno} = \text{retorno algoritmo} - \text{retorno benchmark}.$$

A Figura 4.2 reporta o histograma da distribuição do excedente retorno. Os resultados mostram uma assimetria positiva na distribuição dessa estatística, o que indica uma possível eficácia do algoritmo proposto. Pode ser observado que em alguns casos o algoritmo fica com performance ligeiramente pior do que o *benchmark*. Porém, ao supera-lo, tem um ganho significativo.

A média amostral do excedente retorno foi de 36.76%. Em 56% dos casos o algoritmo superou o *benchmark*.

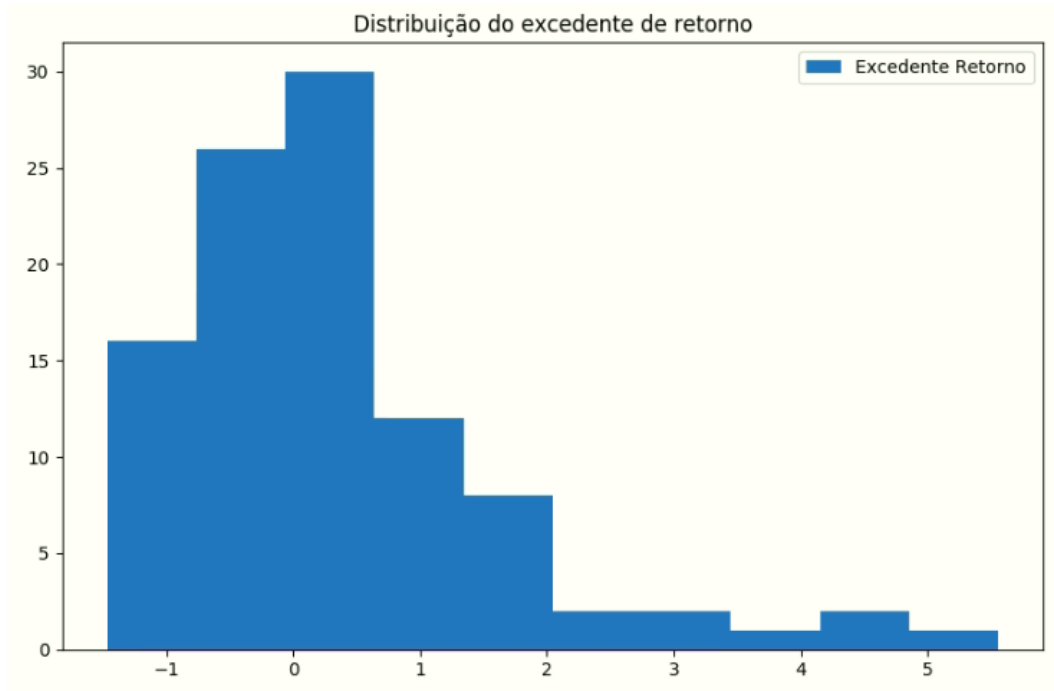


Figura 4.2: Histograma da distribuição do excedente retorno. O eixo  $x$  representa o retorno excedente, em valores absolutos.

Na Figura 4.3 ilustramos o retorno acumulado do algoritmo proposto para a rodada com os ativos CMIG4, ELET3, ITUB4 e USIM5. Neste caso, o algoritmo teve uma performance muito superior ao *benchmark* escolhido.

### 4.3

#### Aprendizado com parâmetro único para todo universo

Consideramos a possibilidade de existência um regime único  $TF$  ou  $MR$  para todos os ativos objeto, que muda ao longo do tempo. Para capturar essa evolução de comportamento, faz sentido utilizar um *MWU master* fazendo a escolha entre apenas dois algoritmos base.

#### 4.3.1

##### Meta-algoritmo para aprendizado de parâmetro comum a todos os ativos

O algoritmo proposto aqui trabalha de forma análoga ao explicado na Seção 4.2.1. Sua diferença é que atua apenas com dois *experts* base - um deles utilizando o parâmetro de aprendizado único positivo para todos os ativos, e o outro negativo (veja Figura 4.4).

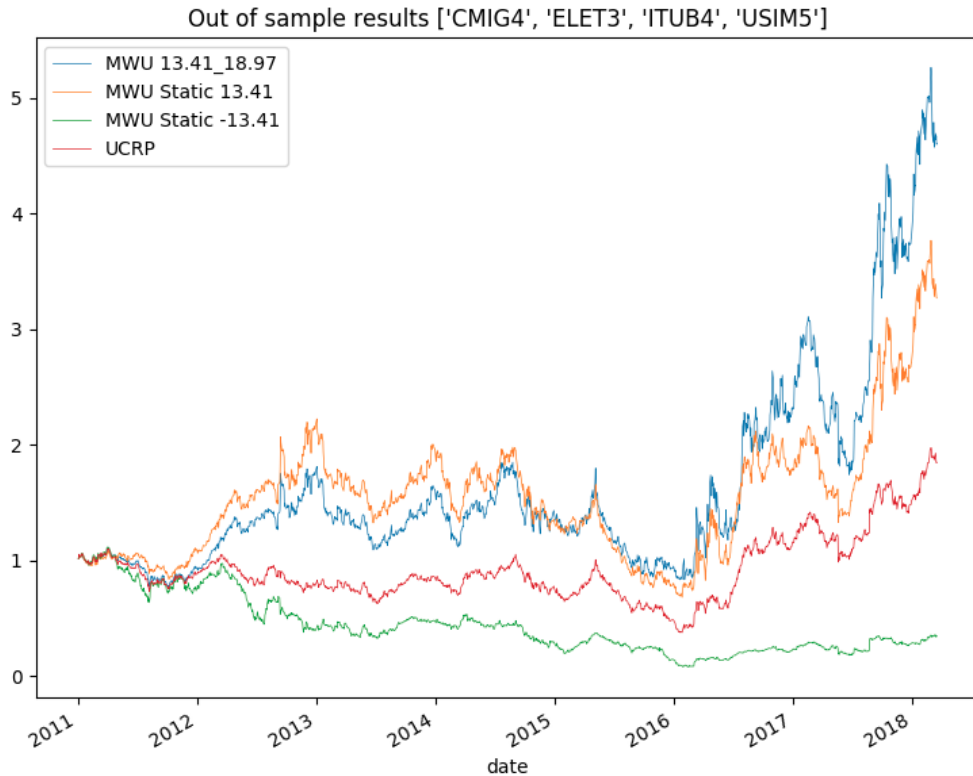


Figura 4.3: Exemplo de retorno acumulado do algoritmo proposto para os ativos CMIG4, ELET3, ITUB4 e USIM5. Para este conjunto de ativos a otimização feita resultou na escolha dos parâmetros de  $\eta_{base} = 13.41$  e  $\eta_{master} = 18.97$ . O algoritmo em questão está representado pela curva azul. Pode ser observado que a técnica supera os resultados dos algoritmos com  $\eta$  estático em ambos casos - com hiperparâmetro de aprendizado positivo e negativos - e também o *benchmark* UCRP.

### 4.3.2 Experimento

Como temos agora somente 2 algoritmos no nível inferior (ao invés de  $2^n$  da seção anterior), esse algoritmo poderia ser executado sobre todas as ações do índice Ibovespa. Entretanto, conduzimos o experimento em múltiplos conjuntos aleatórios de quatro ativos, para efeito de comparação com o experimento anterior.

Novamente, fazemos uma otimização no período  $IS$  que vai de 2006 até 2010, e escolhemos os parâmetros que maximizam o retorno total.

A Figura 4.5 reporta o histograma da distribuição do excedente retorno. Nota-se o mesmo comportamento do experimento anterior, porem com menos eficiência. Os resultados da amostra tem média do excedente retorno igual a 12.07%. O algoritmo supera o *benchmark* em 46% das execuções.

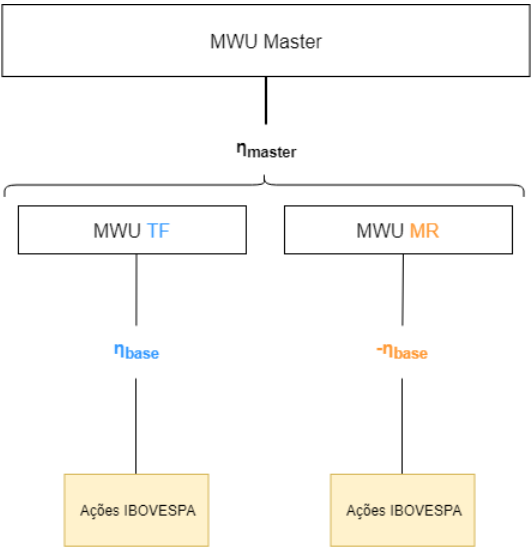


Figura 4.4: Exemplo de fluxo de atualização multiplicativa do meta-algoritmo de regime único.

PUC-Rio - Certificação Digital Nº 1621785/CA

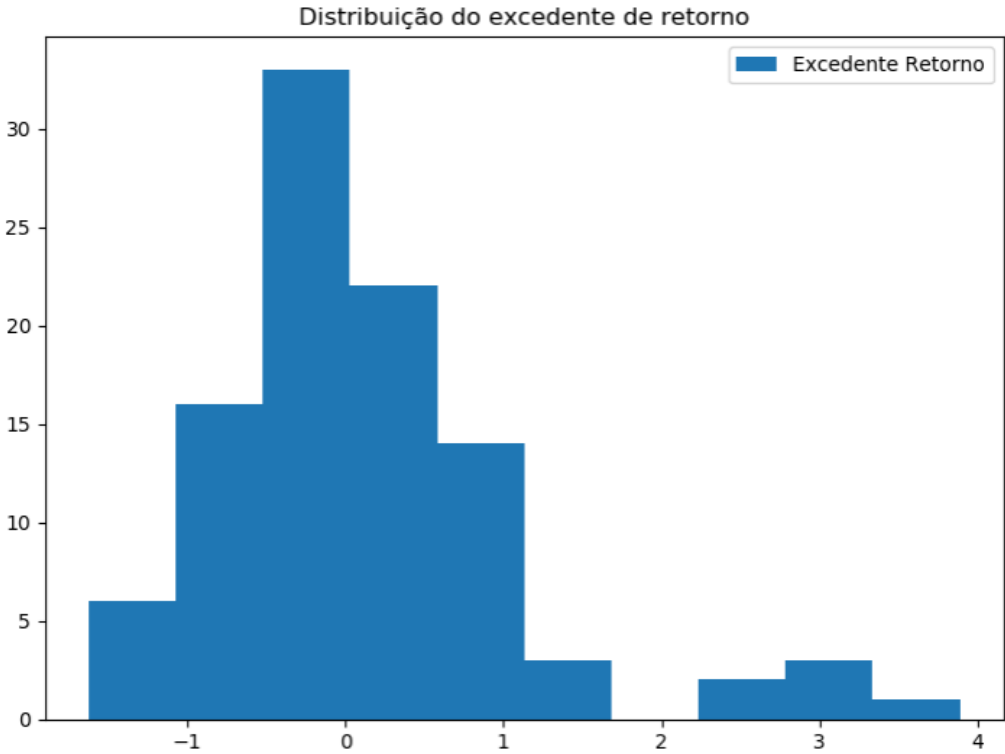


Figura 4.5: Histograma da distribuição do excedente retorno para o algoritmo com parâmetro de aprendizado comum a todos os ativos.

Em comparação com o algoritmo anteriormente proposto, esse resultado é ligeiramente pior. Pode-se dizer, portanto, que faz sentido considerar as características individuais de cada ação e suas diferentes combinações de

regimes  $TF$  e  $MR$ . Não parece existir um regime único entre todos os ativos. O algoritmo anterior é mais flexível para aprender esse tipo de comportamento.

## 5

## Conclusão

Nesta dissertação foram discutidas metodologias para a seleção de portfólios de ações no mercado brasileiro. Para isso, utilizamos um método conhecido na literatura de *Machine Learning* chamado *Multiplicative Weights Update (MWU)*, o qual possui uma garantia teórica de ter performance próxima da melhor ação do universo.

A principal contribuição do trabalho foram duas versões de meta-algoritmos para execução do *MWU* no mercado de ações brasileiras sem depender majoritariamente da escolha do parâmetro de aprendizado. Os resultados sugerem uma performance superior ao portfólio uniformemente rebalanceado, que por sua vez é um *benchmark* bastante desafiador. Futuros trabalhos podem fazer experimentos com os meta-algoritmos propostos em outros mercados, tal qual as bolsas de ações dos Estados Unidos.

Inicialmente, focamos o trabalho no estudo do parâmetro de aprendizado do *MWU* a partir da diferenciação entre valores positivos e negativos. Associamos os valores positivos a comportamentos seguidores de tendência (*TF*), e valores negativos a comportamentos de reversão à média (*MR*).

Utilizamos medidas conhecidas na literatura econométrica para detecção de memória em séries financeiras e, posteriormente, para escolha do parâmetro de aprendizado do *MWU*. Essa primeira tentativa não gerou resultados satisfatórios, indicando que essas medidas não parecem surtir efeito em conjunto com a atualização multiplicativa. Isso não significa, entretanto, que as medidas não tem pertinência. Testes futuros podem ser conduzidos utilizando os indicadores estudados em conjunto com outras estratégias de tendência ou contra-tendência.

Um possível direção é explorar mecanismos de *stop-loss* para algoritmos *MWU* com parâmetro de aprendizado negativo. No presente trabalho, optamos por um mecanismo trivial, porém existem alternativas que possivelmente podem ser superiores. Os resultados no período usado como *out of sample* foram muito afetados por ações que foram a falência rapidamente. Seria interessante ter algum mecanismo de proteção mais rebuscado.

Um aspecto importante não explorado em nosso algoritmo principal *Experts of Experts* é o horizonte da tendência ou contra-tendência esperada. No

*MWU*, isso tem relação com a frequência de atualização multiplicativa — por exemplo, pode-se optar por atualizar o portfólio mensalmente, trimestralmente ou até mesmo em intervalos de quinze minutos. Nesse trabalho, optamos por fixar a atualização diária.

Finalmente, para tornar os algoritmos propostos viáveis *do ponto de vista prático*, é necessário explorar a questão do custo transacional, que não foi considerado nesse trabalho.

De forma geral, buscamos nesse trabalho aplicar técnicas recentes da área de *machine learning* à seleção de portfólios. Existe uma tendência global de automatização dos processos de investimento, portanto o trabalho se mostra pertinente ao cenário atual. O problema de escolha de portfólios, por sua vez, é de extrema importância para a gestão de investimentos. Com a evolução das técnicas de decisão sob incerteza, acreditamos que novas soluções serão propostas para esse problema.

Em se tratando dos algoritmos propostos, percebemos maior eficácia nos modelos que não utilizam definições explícitas dos regimes de TF e MR, indicando possivelmente que a tentativa de classificação entre esses dois regimes não faça sentido. Todavia, a utilização simultânea dos conceitos contrários de FTW e FTL se provou benéfica e deve ser explorada em futuros trabalhos.

## Referências bibliográficas

- [1951] HURST, H. E.. Long-term storage capacity of reservoirs. Trans. Amer. Soc. Civil Eng., 116:770–799, 1951.
- [1952] MARKOWITZ, H.. Portfolio selection. The journal of finance, 7(1):77–91, 1952.
- [1988] LITTLESTONE, N.. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. Machine learning, 2(4):285–318, 1988.
- [1991] BEST, M. J.; GRAUER, R. R.. On the sensitivity of mean-variance-efficient portfolios to changes in asset means: some analytical and computational results. The review of financial studies, 4(2):315–342, 1991.
- [1991] COVER, T. M.. Universal portfolios. Mathematical Finance, 1(1):1–29, 1991.
- [1994] PETERS, E.; PETERS, D.. Fractal Market Analysis: Applying Chaos Theory to Investment and Economics. Wiley Finance. Wiley, 1994.
- [1996] COVER, T. M.; ORDENTLICH, E.. Universal portfolios with side information. IEEE Transactions on Information Theory, 42(2):348–363, 1996.
- [1997] FREUND, Y.; SCHAPIRE, R. E.; SINGER, Y. ; WARMUTH, M. K.. Using and combining predictors that specialize. In: PROCEEDINGS OF THE TWENTY-NINTH ANNUAL ACM SYMPOSIUM ON THEORY OF COMPUTING, STOC '97, p. 334–343, New York, NY, USA, 1997. ACM.
- [1998] ORDENTLICH, E.; COVER, T. M.. The cost of achieving the best portfolio in hindsight. Mathematics of Operations Research, 23(4):960–982, 1998.
- [1998] HELMBOLD, D. P.; SCHAPIRE, R. E.; SINGER, Y. ; WARMUTH, M. K.. On-line portfolio selection using multiplicative updates. Mathematical Finance, 8(4):325–347, 1998.



- [1998] HERBSTER, M.; WARMUTH, M. K.. **Tracking the best expert.** Machine learning, 32(2):151–178, 1998.
- [1999] BLUM, A.; KALAI, A.. **Universal portfolios with and without transaction costs.** Machine Learning, 35(3):193–205, 1999.
- [1999] FREUND, Y.; SCHAPIRE, R. ; ABE, N.. **A short introduction to boosting.** Journal-Japanese Society For Artificial Intelligence, 14(771-780):1612, 1999.
- [2003] KALAI, A.; VEMPALA, S.. **Efficient algorithms for online decision problems.** In: COLT, 2003.
- [2004] BORODIN, A.; EL-YANIV, R. ; GOGAN, V.. **Can we learn to beat the best stock.** In: ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS, p. 345–352, 2004.
- [2005] TSAY, R. S.. **Analysis of financial time series**, volumen 543. John Wiley & Sons, 2005.
- [2006] EVEN-DAR, E.; KEARNS, M. ; WORTMAN, J.. **Risk-sensitive on-line learning.** In: INTERNATIONAL CONFERENCE ON ALGORITHMIC LEARNING THEORY, p. 199–213. Springer, 2006.
- [2007] DEMIGUEL, V.; GARLAPPI, L. ; UPPAL, R.. **Optimal versus naive diversification: How inefficient is the  $1/n$  portfolio strategy?** The review of Financial studies, 22(5):1915–1953, 2007.
- [2012] ARORA, S.; HAZAN, E. ; KALE, S.. **The multiplicative weights update method: a meta-algorithm and applications.** Theory of Computing, 8(1):121–164, 2012.
- [2014] LI, B.; HOI, S. C.. **Online portfolio selection: A survey.** ACM Computing Surveys (CSUR), 46(3):35, 2014.
- [2014] GAILLARD, P.; STOLTZ, G. ; VAN ERVEN, T.. **A second-order bound with excess losses.** In: Balcan, M. F.; Feldman, V. ; Szepesvári, C., editors, PROCEEDINGS OF THE 27TH CONFERENCE ON LEARNING THEORY, volumen 35 de **Proceedings of Machine Learning Research**, p. 176–196, Barcelona, Spain, 13–15 Jun 2014. PMLR.
- [2015] BEN-TAL, A.; HAZAN, E.; KOREN, T. ; MANNOR, S.. **Oracle-based robust optimization via online learning.** Operations Research, 63(3):628–638, 2015.

- [2016] HAZAN, E.; OTHERS. **Introduction to online convex optimization**. Foundations and Trends® in Optimization, 2(3-4):157–325, 2016.
- [2017] RATHER, A. M.; SASTRY, V. ; AGARWAL, A.. **Stock market prediction and portfolio selection models: a survey**. OPSEARCH, 54(3):558–579, 2017.
- [2017] BÄRMANN, A.; POKUTTA, S. ; SCHNEIDER, O.. **Emulating the expert: Inverse optimization through online learning**. In: INTERNATIONAL CONFERENCE ON MACHINE LEARNING, p. 400–410, 2017.
- [Adamskiy2016] ADAMSKIY, D.; KOOLEN, W. M.; CHERNOV, A. ; VOVK, V.. **A closer look at adaptive regret**. Journal of Machine Learning Research, 17(23):1–21, 2016.
- [Helmbold et al. (1998)] HELMBOLD, D. P.; SCHAPIRE, R. E.; SINGER, Y. ; WARMUTH, M. K.. **On-line portfolio selection using multiplicative updates**. Mathematical Finance, 8(4):325–347, 1998.