

Predição do Mercado de Ações Usando Hidden Markov Model

Daniela Kuinchtner, Guilherme Afonso Madalozzo

Curso de Ciência da Computação – Universidade de Passo Fundo (UPF) Caixa Postal:
611 – 99.052-900 – Passo Fundo, RS – Brasil

{152064, guimadalozzo}@upf.br

***Abstract.** Stock markets are complex systems due its non-stationarity, the parameters are always in constant changes like economic conditions and changes in the politics of companies. There are several researches in predicting the stocks with Artificial Intelligence and Machine Learning techniques, such as Artificial Neural Networks, Support Vector Machine, Fuzzy Logic, Pattern Recognition, which, in the latter, the Hidden Markov Model, focus of this paper, fits. Because it is a stochastic model, where events are random, the results show that HMM can be well used in predicting the behavior of stocks based on their historical performance.*

***Resumo.** Os mercados de ações são sistemas complexos devido a sua não-estacionariedade, pois os parâmetros estão sempre em constantes mudanças, como condições econômicas e mudanças na política de empresas. Há várias pesquisas para a predição dos valores de ações com técnicas de inteligência artificial e aprendizagem de máquina, como redes neurais artificiais, máquina de vetores de suporte, lógica difusa, reconhecimento de padrões, onde, neste último, o Hidden Markov Model, foco deste trabalho, se encaixa. Por ser um modelo estocástico, onde os eventos são aleatórios, os resultados mostram que o HMM pode ser bem empregado na predição do comportamento das ações com base em seu desempenho histórico.*

1. Introdução

O mercado de ações é um sistema de compra e venda de frações do capital social de empresas. Assim, quando ações são compradas o investidor passa a se tornar sócio dessa empresa, podendo obter lucro com o crescimento da companhia. Essas ações são negociadas numa bolsa de valores, no caso do Brasil, na BM&FBovespa. Os investidores usam a tecnologia disponibilizada pelas corretoras de investimentos, chamada de *home broker*, para enviar ordens de compra e venda de ações à bolsa [Mercado de ações 2018]. Porém o processo de enviar a ordem para o *home broker* pode ser lento, fazendo com que a oportunidade de compra ou venda se perca [Infomoney 2018]. Por isso, vem surgindo uma grande tendência para o mercado de ações: a automatização do pregão (compra e venda de ações) por conta de robôs investidores [Robôs 2018].

O uso de robôs tem ganhado destaque no mercado financeiro pelo emprego dos mesmos no suporte de tomada de decisão no pregão. São chamados de robôs investidores os programas de inteligência artificial responsáveis pela análise do banco de dados e do comportamento histórico dos preços das ações, dessa forma, os mesmos são capazes de determinar o melhor momento de comprar ou vender um ativo em determinada tendência. Têm sido fortemente usados pelas principais corretoras de investimentos [Corretoras

2018], como: Clear [Clear 2018], XP Investimentos [XP 2018; Infomoney 2018; Smarttbot 2018], Mirae Asset [Mirae 2018], Rico [Trovoacademy 2018] e Easynvest [Easynvest 2018] devido a rapidez, flexibilidade, eficiência e lucratividade, ao automatizar os investimentos, com o uso de robôs investidores [Bússola 2018]. Alguns exemplos de robôs utilizados [Trovoacademy 2018]: *I Know First* [I Know First 2018], *TradeMachine* [TradeMachine 2018], Robô Investe [Robô Investe 2018], Warren [Warren 2018], *Magnetis* [Magnetis 2018], *Monetus* [Monetus 2018], *Verios* [Verios 2018], *Alkanza* [Alkanza 2018] e *SmarttBot* [SmarttBot 2018], sendo que todos os serviços prestados por esses robôs têm custos que variam de plataforma para plataforma.

Com isso, dentre as várias técnicas de inteligência artificial utilizadas para a predição do mercado de ações, pode-se citar: redes neurais artificiais [Moghaddam, A., Moghaddam, M. & Esfandyari, M. 2016], máquina de vetores de suporte [Chen, Y., & Hao, Y. 2017], algoritmos genéticos [Cheng, C., Chen, T. & Wei, L. 2010] e reconhecimento de padrões, onde o *Hidden Markov Model*, que será descrito no presente trabalho, é empregado.

O *Hidden Markov Model* (HMM) é uma ferramenta robusta para a análise e a predição de fenômenos em séries temporais não-estacionárias por meio de sequências de observações ao longo do tempo. Uma série temporal é dita não-estacionária quando a mesma não se desenvolve através de uma média constante no tempo, dessa forma, a mesma não apresenta estabilidade [Estacionariedade 2018].

O HMM tem sido usado em diversas aplicações, como: análise de sequência de DNA [Liebert, M. A. 2004], reconhecimento de sinais de fala [Rabiner 1989; Huang X, Ariki Y, Jack M 1990; Jelinek F, Kaufmann M, Mateo C S 1990; Xie H, Anrae P, Zhang M, Warren P 2004], aplicações em biologia computacional [Krogh, A., Brown, M., Mian, S., Sjölander, K. & Haussler, D. 1994], reconhecimento de padrões [M. Rafiul Hassan & Nath 2005], reconhecimento de palavras escritas à mão [Yacoubi A., Sabourin R., Gilloux M., Suen C. Y. 1999] [Vinciarelli, A. & Luetin, J. 2000] e de palavras em imagens [Chen, F. R., Wilcox, L. D. & Bloomberg, D. S. 1995].

Como apresentado nos trabalhos relacionados no Capítulo 8, o HMM mostra-se um algoritmo de grande valia para a previsão de valores com base no comportamento histórico analisado. Então, o presente trabalho tem por principal objetivo o desenvolvimento de um algoritmo para a predição do mercado de ações, levando em conta diferentes empresas e nichos de mercado, utilizando o algoritmo HMM. Para isso, o HMM será analisado de forma bruta e individualizada, sem a presença de um segundo método. O foco principal do presente trabalho não é apresentar o modelo matemático HMM, mas sim a aplicação deste modelo para a previsão de dados. Por fim, será mostrado a comparação dos valores previstos pelo HMM com os valores reais.

2. Hidden Markov Model

Como mencionado no Capítulo anterior, o HMM é um algoritmo com finalidades distintas, uma delas é o alto poder de cálculo para que possa ser reconhecido padrões em uma gama de dados. A ideia conceitual do HMM é que a probabilidade das observações depende dos estados “ocultos” para o observador. A transição de um estado para outro é um Processo de *Markov*, ou seja, o próximo estado depende apenas do estado atual, por isso o nome de *Hidden Markov Model* (Modelo Oculto de *Markov*) [Rabiner 1989].

Os elementos básicos que fazem parte de um HMM, são [Rabiner 1989]:

- a) N , número de estados (ocultos) no modelo: são as variáveis que o observador não tem conhecimento, como por exemplo, para um algoritmo que se deseja determinar a previsão do tempo pode-se usar dois estados: Alta e Baixa, para determinar a pressão atmosférica.
- b) M , número de observações por estado: são as variáveis visíveis que dependem dos estados, como por exemplo, “chuvoso” e “ensolarado”. Cada uma dessas observações terá probabilidades distintas para cada estado, como: $P(\text{Alta}(\text{chuvoso} = 0.4, \text{ensolarado} = 0.6))$ e $P(\text{Baixa}(\text{chuvoso} = 0.6, \text{ensolarado} = 0.4))$.
- c) A , distribuição da probabilidade de transição de estado: são as probabilidades das transições de um estado para o outro, como por exemplo: $P(\text{Alta} | \text{Baixa})$ e $P(\text{Baixa} | \text{Alta})$. Os resultados podem ser representados por uma matriz, como mostra um exemplo na Tabela 2 e 3.
- d) B , distribuição da probabilidade de emissão de estado: são as probabilidades de saída que interliga os estados ocultos e as observações, podem ser representadas em forma de matriz (como mostra a Tabela 1) e são calculadas pelo algoritmo Viterbi, mencionado nas Seções 2.2 e 2.3;
- e) π , distribuição da probabilidade inicial de estado: são as probabilidades que cada estado terá como partida, como por exemplo: $P(\text{Baixa}) = 0.4$, $P(\text{Alta}) = 0.6$. Essas probabilidades serão calculadas pelo algoritmo *Forward* e/ou *Backward*, mencionados nas Seções 2.2 e 2.3.
- f) λ , modelo.

A Figura 1 ilustra a estrutura do HMM.

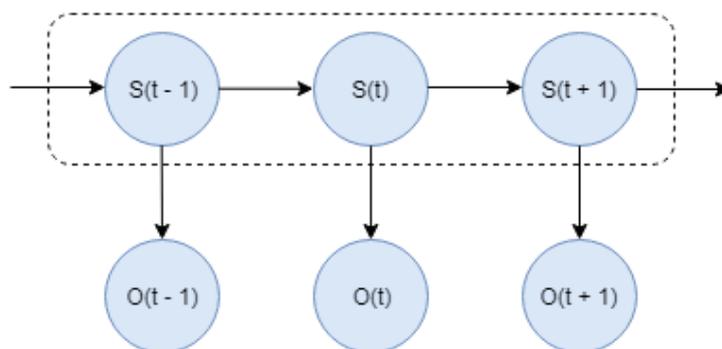


Figura 1. HMM

Fonte: adaptado de Harvard University, 2018

Como apresentado na Figura 1, os estados (S) são os estados ocultos para o observador, ou seja, são os parâmetros não observáveis que são cruciais para se obter o resultado de uma previsão, as observações (O) são visíveis e as mesmas representam a saída do sistema que está sendo modelado. Como pode-se notar, a variável oculta $S(t)$ está no instante de tempo t e a mesma depende do valor da variável oculta $S(t-1)$, ou seja, depende do valor anterior ao instante t . Os valores dos instantes de tempo $t-2$, $t-3$ e assim por diante, não tem influência, esse conceito é chamado de Processo de Markov, mencionado acima. Conseqüentemente, o valor da variável de observação $O(t)$ depende do valor oculto de $S(t)$, ambas no tempo t , dessa forma, o HMM pode gerar uma sequência de observações como saída do modelo.

Os estados no HMM sempre são discretos pela necessidade de determinar a quantidade de estados no modelo [Rabiner 1989], pois o processo de descobrir a parte oculta consiste em um conjunto finito de estados conectados entre si por transições com probabilidades, ilustrado pelo traço pontilhado na Figura 1 acima para indicar os estados e suas transições.

Já as observações podem ser *i*) discretas (do inglês, *Discrete Hidden Markov Model* - DHMM), como por exemplo: caracteres de um alfabeto finito, vetores quantizados ou um livro de códigos (do inglês, *codebook*), ou *ii*) contínuas (do inglês, *Continuous Hidden Markov Model* - CHMM), exemplo de vozes, medidas de temperatura, música, etc. Esse processo observado consiste em um conjunto de saídas ou observações, cada qual podendo ser emitido por cada estado de acordo com uma função de densidade de probabilidade, sendo elas funções discretas ou contínuas [Yacoubi A., Sabourin R., Gilloux M., Suen C. Y. 1999], dessa forma é possível determinar a probabilidade de um evento aleatório ocorrer.

A Figura 2 apresenta as probabilidades de emissão de estados e de transição de estados de um HMM.

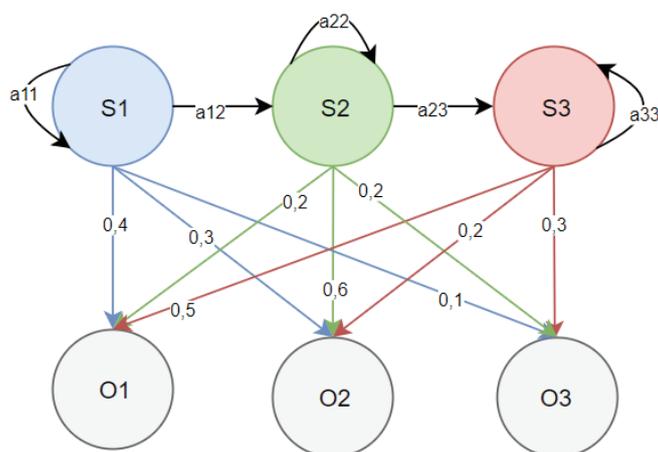


Figura 2. HMM com transição e emissão de estados

Fonte: Sinais, 2018

Na Figura 2 é ilustrado um HMM composto por 3 estados (S_1 , S_2 e S_3). As transições entre esses estados são caracterizadas por a_{11} , a_{12} , a_{22} , a_{23} e a_{33} , a Seção 2.1 irá apresentar os tipos de transições de estados do HMM. Também, pode-se notar que há 3 observações discretas (O_1 , O_2 e O_3) e as probabilidades de emissão de estados são ilustradas pelos valores emitidos pelos estados para as observações. Abaixo, na Tabela 1, são apresentadas as probabilidades de emissão em forma de uma matriz.

Tabela 1. Matriz das probabilidades de emissão da Figura 2

	O_1	O_2	O_3
S_1	0,4	0,3	0,1
S_2	0,2	0,6	0,2
S_3	0,5	0,2	0,3

As probabilidades de emissão de estados caracterizam a saída do modelo, podendo ser analisadas pelo Gráfico 1, onde cada estado do modelo (S_1 , S_2 e S_3) gera uma saída referente as observações.

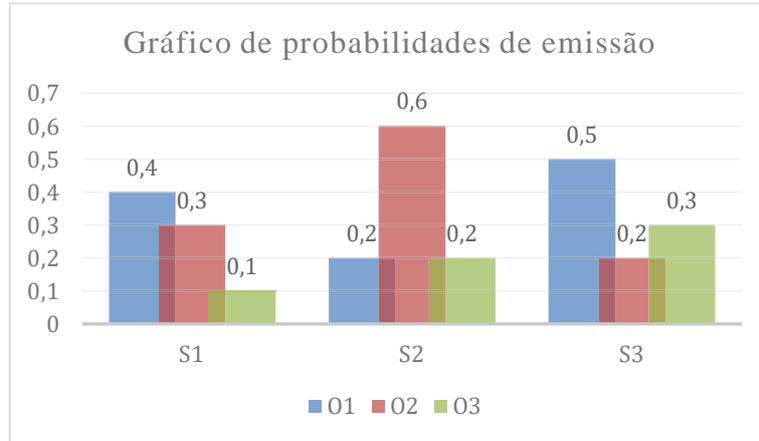


Gráfico 1. Probabilidades de emissão dos estados com observações contínuas

Em DHMM é utilizada uma função de densidade de probabilidade discreta, como apresentado no Gráfico 1, onde as observações, também chamadas de sinais, são naturalmente discretas e extraídas de um alfabeto ou de um *codebook* [Rabiner 189]. Ou seja, possui uma quantidade contável de parâmetros, pois pertencem a um conjunto finito de dados e é empregado para um caso mais simples de predição. Abaixo, na Figura 3 é apresentado um HMM discreto utilizando a Figura 1, que ilustra o processo de *Markov*, juntamente com o Gráfico 1.

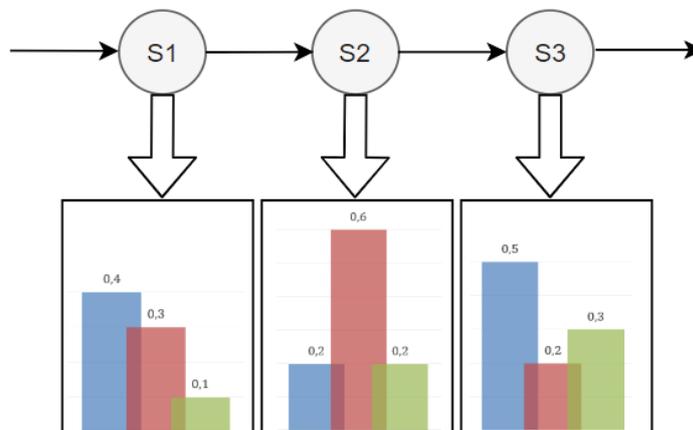


Figura 3. HMM com observações discretas

Fonte: adaptado de Huang, 2004

A Figura 3 apresenta um HMM de três estados (S_1 , S_2 e S_3) com observações discretas ilustradas pelos gráficos de barras. Essas observações podem assumir somente valores dentro do intervalo do conjunto de dados, ou seja, só é possível ter um número contável de valores como mostrado na representação dos gráficos. Como pode-se notar, as observações representam as probabilidades de emissão da Figura 2, ilustradas, também, pela Tabela 1 e pelo Gráfico 1.

No caso de um CHMM é utilizada uma função de densidade de probabilidade contínua, ilustrada pela Figura 4, geralmente é usada uma função Gaussiana, também

chamada de normal, podendo ser uma única distribuição ou misturas de distribuições Gaussianas [Chen, F. R., Wilcox, L. D. & Bloomberg, D. S. 1995]. Neste caso, há vários parâmetros de entrada que geram um vetor de saída do mesmo tamanho fazendo com que os resultados sejam mais precisos. Porém, em alguns casos, como no estudo de reconhecimento de palavras em imagens [Chen, F. R., Wilcox, L. D. & Bloomberg, D. S. 1995], o processo é mais lento que o HMM discreto, devido ao alto processamento de dados.

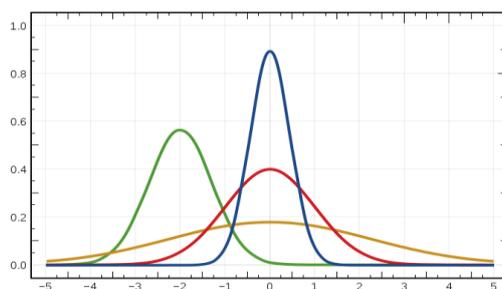


Figura 4. Função contínua

Fonte: adaptado de Gráfico contínuo, 2018

Na Figura 4 os dados podem assumir, teoricamente, quaisquer valores dentro de um intervalo (mínimo e máximo) [Variáveis 2018], sendo que o eixo y é a representação das observações no tempo contínuo e o eixo x representa o tempo contínuo. A Figura 5 apresenta o HMM contínuo utilizando as ilustrações das funções de probabilidade da Figura 1 juntamente com a Figura 4 de misturas de probabilidades Gaussianas.

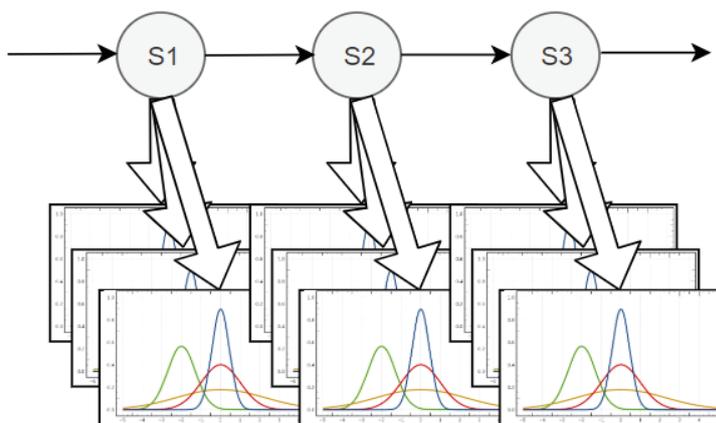


Figura 5. HMM com observações contínuas

Fonte: adaptado de Huang, 2004

A Figura 5 ilustra um HMM de três estados (S_1 , S_2 e S_3) com observações contínuas ilustradas pelos vetores de misturas de funções Gaussianas, onde é possível notar que as observações possuem um número infinito de valores entre dois valores quaisquer determinados pela área da curva.

Também, é importante ressaltar que há o HMM Semi-contínuo (do inglês, *Semi-Continuous Hidden Markov Model* - SCHMM), uma das variações do HMM, onde a função de densidade de probabilidade é representada por uma combinação das probabilidades discretas e das funções de densidade Gaussianas do modelo [Huang, X. D. & Jack, M. A. 2004].

2.1 Tipos de transição de estados

Há dois tipos mais conhecidos de transição de estados em HMMs que podem ser implementados: i) *Ergodic*, também chamado de Inteiramente Conectados e ii) *Left-right*, também chamado de *Bakis*.

No primeiro caso, chamado de *ergodic*, todo estado do modelo pode ser alcançado, em um número finito de passos, de todos os outros estados do modelo [Rabiner 1989]. A Figura 6 ilustra o tipo *Ergodic*.

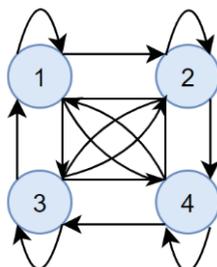


Figura 6. HMM Ergodic

Fonte: adaptado de Rabiner, 1989

Pode-se notar que na Figura acima contém 4 estados: 1, 2, 3 e 4, onde todos estão conectados, permitindo, assim, transições de estados com todos os estados presentes no modelo. A Tabela 2 mostra a matriz de transição gerada a partir desse modelo.

Tabela 2. Matriz de transição do modelo *ergodic*

	1	2	3	4
1	a_{11}	a_{12}	a_{13}	a_{14}
2	a_{21}	a_{22}	a_{23}	a_{24}
3	a_{31}	a_{32}	a_{33}	a_{34}
4	a_{41}	a_{42}	a_{43}	a_{44}

Já no segundo caso, o tipo *left-right* é um algoritmo sequencial que funciona como uma evolução em série do modelo através de seus estados, mesmo que qualquer um desses estados possa ser saltado no curso do processo [Rabiner 1989]. Muito usado em reconhecimento de fala [Cappé, O., Moulines, E. & Rydén, T. 2009], pela mesma apresentar propriedades que mudam conforme o tempo [Rabiner 1989].

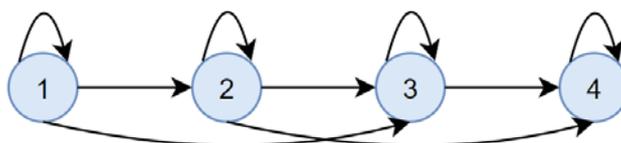


Figura 7. HMM Left-to-right

Fonte: adaptado de Rabiner, 1989

A Figura 7 representa um modelo do tipo *left-right* com 4 estados: 1, 2, 3 e 4, onde o mesmo começa num estado inicial (1) que pode ter transição de estados com os estados

intermediários (2 e 3) sem poder retroceder e, finalmente, termina num estado final (4). A matriz de transição gerada a partir do modelo *left-right* é apresentada pela Tabela 3.

Tabela 3. Matriz de transição do modelo left-right

	1	2	3	4
1	a_{11}	a_{12}	a_{13}	0
2	0	a_{22}	a_{23}	a_{24}
3	0	0	a_{33}	a_{34}
4	0	0	0	a_{44}

Como mencionado no Capítulo 2, os estados no HMM são sempre discretos, pelo motivo dos mesmos precisarem ter um número finito determinado, como por exemplo, na implementação foi usado o tipo *left-right* com 4 estados (preço de abertura, mais alto, mais baixo e de fechamento – OHLC, do inglês *open, high, low, close*). A escolha de qual tipo usar depende da aplicação a ser desenvolvida [Rabiner 1989].

2.2 Problemas

Embora o HMM seja crucial para várias aplicações, seu papel é apenas modelar a situação. Para poder aplicá-lo, há três problemas fundamentais que precisam ser resolvidos por meio de algoritmos: *i*) problema de avaliação, *ii*) problema de decodificação e *iii*) problema de treinamento. Os algoritmos são citados a seguir e descritos na Seção 2.3.

No primeiro problema, que se chama problema de avaliação, procura-se calcular a probabilidade ($P(O | \lambda)$), também chamada de verossimilhança, de uma sequência de observação ($O = O_1, O_2, \dots O_t$) dado um modelo λ . Desta forma, deve-se iniciar as probabilidades para determinar quais são as chances de começar em um estado específico. Pode-se utilizar, com o princípio da programação dinâmica, os algoritmos *Forward* e/ou *Backward* para tratar esse problema [Rabiner 1989].

O segundo problema, chamado de problema de decodificação, descobre a parte oculta do modelo para determinar a melhor sequência de estados do modelo, sendo que a previsão da sequência de mudanças de estado é obtida com base na sequência de observações [Rabiner 1989]. Em outras palavras, verifica-se a probabilidade de emissão de estado para determinar qual a sequência que melhor “explica” as observações. Utiliza-se o algoritmo *Viterbi*, algoritmo de programação dinâmica, para encontrar a sequência mais provável de estados ocultos que resulta em uma sequência de eventos observados [Rabiner 1989].

O terceiro problema, chamado de problema de treinamento, é onde ocorre o ajuste dos parâmetros do modelo para melhorar a otimização da probabilidade da sequência de observação. Essa sequência de observação usada para otimizar é chamada de sequência de treinamento, ou seja, para treinar o HMM. O algoritmo usado nessa etapa é o *Baum-Welch*, caso especial do algoritmo de maximização de expectativa [Rabiner 1989].

A escolha do algoritmo adequado depende muito da aplicação [Rabiner 1989]. Em [M. Rafiul Hassan & Nath 2005] utilizou-se os algoritmos *Forward-Backward*,

Viterbi e *Baum-Welch* para a solução dos problemas 1, 2 e 3, respectivamente. Na Seção a seguir é descrito cada algoritmo.

2.3 Algoritmos

Como mencionado na Seção anterior, há três problemas que precisam ser resolvidos para poder implementar o HMM, estes problemas (problema de avaliação, problema de decodificação e problema de treinamento) são resolvidos por meio de algoritmos: *forward* e/ou *backward* para o problema 1, *viterbi* para o problema 2 e *baum-welch* para o problema 3.

Os algoritmos *forward* e *backward* utilizam a técnica de inferência na sua implementação, para deduzir as propriedades de uma distribuição de probabilidade (contínua ou discreta, mencionadas no Capítulo 1), os mesmos fornecem como resultado o valor da verossimilhança dos estados, ou seja, o valor da probabilidade $P(O | \lambda)$ de cada estado oculto usando o princípio da programação dinâmica [Rabiner 1989]. Como por exemplo: há dois estados ocultos (alto e baixo) para representar as oscilações de uma ação, sendo que, por meio do algoritmo, as probabilidades de cada um dos estados são calculadas para obter quais são as chances de uma ação começar no estado “alto” ou no estado “baixo”.

O algoritmo *viterbi*, proposto em 1967 como sendo um algoritmo de programação dinâmica pelo uso da recursão em seus cálculos, resolve a parte oculta do HMM obtendo como resultado uma sequência de estados mais provável, chamado de caminho *Viterbi* [Forney, G. D. 1973]. Por exemplo, há duas observações (comprar e vender) no modelo, sendo que, através do algoritmo será calculado qual é a probabilidade dos estados ocultos alto e baixo ocorrerem em “comprar” e em “vender”.

Os parâmetros iniciais do modelo, obtidos pelos algoritmos acima, são submetidos aos ajustes determinados pelo algoritmo de treinamento, chamado de *baum-welch*. O algoritmo *baum-welch* utiliza a técnica do algoritmo *forward-backward* e é baseado na técnica de otimização Maximização de Expectativa (do inglês, *expectation-maximization*) que consiste em obter os parâmetros desconhecidos do modelo de modo iterativo (pela utilização do algoritmo *forward-backward*) com uma fase de estimação e outra de maximização [Rabiner 1989], ou seja, há algumas variáveis ausentes que precisam ser calculadas através das variáveis observáveis [Variáveis Latentes 2018] então, a fase de estimação determina a atribuição de um valor a um parâmetro, para o qual não se conhece o valor absoluto e a etapa de maximização calcula os parâmetros maximizando o resultado esperado.

3. Implementação

Este Capítulo apresenta a teoria do algoritmo utilizado, bem como, os parâmetros e os dados utilizados no HMM.

Para a validação do algoritmo implementado neste trabalho e as consequências financeiras de suas predições, usa-se os dados históricos e reais de ações das seguintes empresas: B2W Digital (BTOW3), Companhia Energética de Minas Gerais (CMIG4), Grendene (GRND3), Itaúsa (ITSA4), Itaú Unibanco (ITUB4), Petrobras (PTR4), Portobello S.A. (PTBR3) e Vale S.A. (VALE3). Para a execução da predição foram utilizados o preço de abertura, preço de fechamento, preço mais alto e preço mais baixo como entradas do HMM. Assim, utilizou-se quatro estados, onde o tipo de HMM

utilizado foi o *left-right*, pois a predição também contém propriedades que sempre estão em constantes mudanças como a fala, mencionada na Seção 2.1.

A ideia para prever o preço das ações do próximo dia de pregão é identificar um dia no passado cuja verossimilhança de seus eventos é a mais próxima da sequência de observações do preço do dia seguinte, que deve ser previsto. Em seguida, é calculado a mudança do preço diferencial do dia identificado para o dia seguinte. Essa alteração é então adicionada ao preço do dia atual para obter a previsão do próximo dia [M. Rafiul Hassan & Nath 2005].

Por exemplo, se o valor da verossimilhança para o dia atual for X, tenta-se localizar no conjunto de dados passados, usando o HMM, instâncias que produzem o mesmo X ou próximo a este valor. Ou seja, é localizado no passado o preço das ações que têm o mesmo comportamento que o preço atual. Então, assume-se que o dia seguinte no mercado de ações siga o mesmo comportamento que os dos dados no passado. Localizada a data, é calculado a diferença dos valores deste dia com o seu próximo. Esta diferença é adicionada nos valores das ações do dia atual, para que se saiba o valor da ação do próximo dia, que se deseja prever. Sendo que o preço de fechamento do dia seguinte é considerado o preço-alvo associado aos quatro recursos de entrada.

Na implementação foram consideradas as observações como contínuas (CHMM), porque se trata de uma implementação com uma grande quantidade de dados de entrada que podem assumir qualquer valor, e usando distribuição Gaussiana, por ser a distribuição de probabilidade mais utilizada para modelar fenômenos naturais [Distribuição e probabilidade 2018]. Os algoritmos utilizados para resolver os problemas 1, 2 e 3, mencionados na Seção 2.3, foram os algoritmos *Forward*, *Viterbi* e *Baum-Welch*, respectivamente.

Há observações de 2, 7 e 10 anos do histórico das ações das empresas mencionadas anteriormente. A Tabela 4 apresenta quantas observações foram consideradas para cada ação.

Tabela 4. Observações de cada ação

Ação/Anos	2 anos (500 observações)	7 anos (1923 observações)	10 anos (2913 observações)
B2W Digital			X
Com. Energ. MG			X
Grendene		X	
Itaúsa	X		
Itaú Unibanco		X	
Petrobras		X	
Portobello S.A.			X
Vale S.A.			X

Com base nos dados históricos das ações, foram realizadas 4 baterias de testes. Os conjuntos de testes basearam-se nos últimos 7, 15, 30 e 60 dias de pregão (observações) para avaliar e validar o algoritmo, enquanto o restante dos dias históricos foram observações utilizadas para o treinamento do modelo.

4. Código

O código *open source* foi obtido por meio do GitHub [Teja Darbha, R., Jain, A. & Mehta, P. 2018], o mesmo foi adaptado e nele foi utilizada a *hmmlearn*, uma biblioteca *open source* para a linguagem *Python* para treinar o modelo e calcular a verossimilhança das observações. A biblioteca implementa HMMs e nela há três tipos disponíveis para serem utilizados [Hmmlearn 2018]:

- i) HMM com emissão Gaussiana (`hmm.GaussianHMM`), tipo de emissão usada em HMMs com observações contínuas, mencionados no Capítulo 1.
- ii) HMM com emissão de misturas Gaussianas (`hmm.GMMHMM`), tipo de emissão também usada em HMMs com observações contínuas.
- iii) HMM com emissão multinomial discreta (`hmm.MultinomialHMM`), tipo de emissão usada em HMMs com observações discretas.

Cada tipo contém alguns parâmetros diferentes [Hmmlearn 2018], mas os elementos básicos que fazem parte de um HMM, mencionados no Capítulo 1 estão presentes em todos e, para um melhor entendimento sobre o HMM, abaixo, contém um exemplo de um pseudocódigo do mesmo com os elementos básicos utilizados pela biblioteca *hmmlearn*:

```
estados = ('Cair', 'Subir', 'Estagnado') # N
observações = ('comprar', 'vender') # M
prob_inicial = {'Cair': 0.5, 'Subir': 0.3, 'Estagnado': 0.2} #  $\pi$ 
probabilidade_transição = { # A
    'Cair': {'Cair': 0.2, 'Subir': 0.3, 'Estagnado': 0.5},
    'Subir': {'Cair': 0.3, 'Subir': 0.5, 'Estagnado': 0.2},
    'Estagnado': {'Cair': 0.5, 'Subir': 0.4, 'Estagnado': 0.1}
}
probabilidade_emissão = { # B
    'Cair': {'comprar': 0.3, 'vender': 0.7},
    'Subir': {'comprar': 0.6, 'vender': 0.4},
    'Estagnado': {'comprar': 0.5, 'vender': 0.5}
}
```

Nesse pseudocódigo há três estados ocultos (N): cair, subir e estagnado, para representar as oscilações das ações, onde os mesmos não podem ser observados diretamente pelo indivíduo, então, com a utilização do algoritmo *Forward* e/ou *Backward*, é analisado os dados para determinar a probabilidade inicial (π) que os mesmos terão.

Pelo exemplo acima, o padrão dos dados analisados resultou em 50% “cair”, 30% “subir” e 20% “estagnado” como probabilidade inicial, os mesmos têm probabilidade de transição de 20%, 50% e 10% para permanecerem em “cair”, “subir” e “estagnado”, respectivamente e, também, 30% de chance de um estado “cair” se transformar em “subir” e 50% de chance de transição para “estagnado”, 30% de um estado “subir” se transformar em “cair” e 20% para “estagnado”, bem como 50% e 40% de um estado “estagnado se transformar em “cair” e “subir”, respectivamente.

Pelo algoritmo *Viterbi* é encontrada a probabilidade de emissão (B) para determinar quais das observações (M) tem chance de ocorrer em um determinado estado. Se o estado for “cair” tem-se 30% de probabilidade de comprar e 70% de vender, já se o estado for “subir”, tem-se 60% de chance de comprar e 40% de vender e se o estado for “estagnado” tem-se 50% de comprar e 50% de chance de vender. A Figura 8 representa o pseudocódigo com as transições de estados em forma de diagrama.

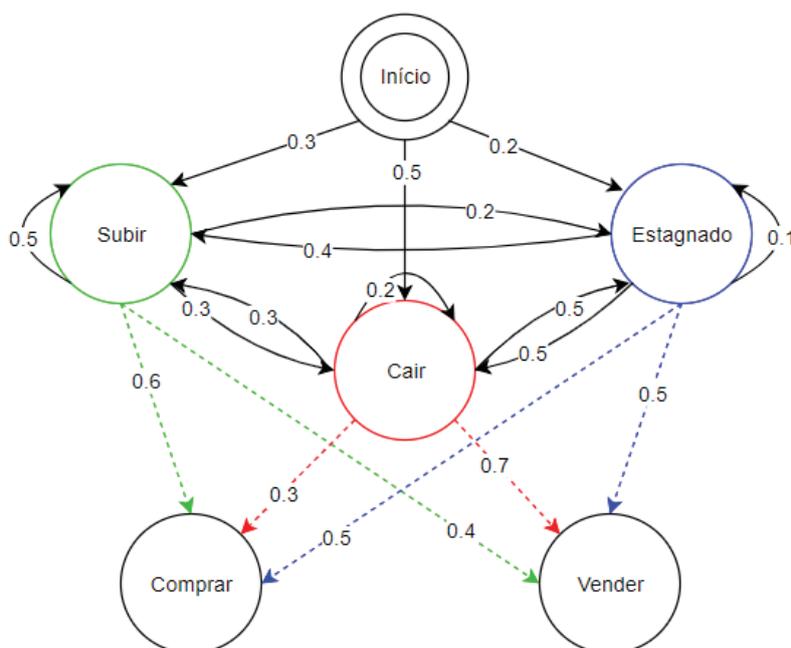


Figura 8. Ilustração do pseudocódigo do HMM

Fonte: adaptado de Github, 2018.

As modificações realizadas no código foram para adaptar a escolha aleatória dos estados e para ajustar o *fit* do conjunto de dados, método utilizado pela biblioteca *hmmlearn* para treinar o HMM. Foi escolhido o código de [Teja Darbha, R., Jain, A. & Mehta, P. 2018] porque o mesmo apresentou as soluções mais usadas e detalhadas em [Rabiner 1989] e implementadas em trabalhos anteriores, como em [Hassen & Nath 2005].

5. Resultados

Os gráficos de mapa de calor 3 ao 6, mostram as previsões dos preços das ações para: B2W Digital (BTOW3), Companhia Energética de Minas Gerais (CMIG4), Grendene (GRND3), Itaúsa (ITSA4), Itaú Unibanco (ITUB4), Petrobras (PETR4), Portobello S.A. (PTBL3) e Vale S.A. (VALE3) para 7, 15, 30 e 60 dias, respectivamente, usando o HMM e comparando os resultados das previsões e dos valores reais do preço de fechamento de cada ação.

Os dados em cores vermelhas nos gráficos representam a queda dos valores das ações e os em verdes são as ações que apresentam lucratividade, tanto para a predição quanto para o valor real da ação. O Gráfico 2 apresenta as empresas selecionadas para a comparação.

Ações do Comparativo	
BTOW3	CMIG4
GRND3	IRSA4
ITUB4	PETR4
PTBL3	VALE3

Gráfico 2. Ações do comparativo

Comparativo 7 Dias			
Predição		Real	
5,45	3,20	4,47	-1,11
-3,39	3,48	-3,82	2,84
0,00	4,52	-1,56	6,02
1,00	1,09	1,24	2,15

Gráfico 3. Comparativo de 7 dias

Comparativo 15 Dias			
Predição		Real	
11,31	1,09	10,27	-3,13
-5,83	0,36	-5,69	4,54
0,18	9,69	-1,38	11,27
9,73	10,32	10,00	11,48

Gráfico 4. Comparativo de 15 dias

Comparativo 30 Dias			
Predição		Real	
-5,20	-8,68	-6,08	-10,57
-9,71	-3,45	-9,57	0,57
18,62	5,54	14,36	9,37
1,75	14,42	2,01	15,62

Gráfico 5. Comparativo de 30 dias

Comparativo 60 Dias			
Predição		Real	
0,00	-3,33	-0,93	-5,33
-14,84	6,90	-15,21	7,44
5,18	20,13	3,17	21,86
10,63	24,39	10,90	25,69

Gráfico 6. Comparativo de 60 dias

Pode-se notar que o comparativo dos 7 dias, no Gráfico 3, apresentou sucesso na predição de 7 das 8 empresas analisadas. O algoritmo, levando em conta apenas as predições de lucro, teria sugerido o investimento para um lucro de 18,74%. Caso esta sugestão tivesse sido aplicada, o lucro real obtido teria sido de 15,61%.

O Gráfico 4 apresenta o comparativo de 15 dias de previsão, resultando sucesso na predição de 5 das 8 empresas analisadas. O algoritmo sugeriu o investimento para um ganho de 42,68%, considerando apenas investimentos previstos com qualquer lucratividade. O lucro obtido teria sido de 47,56% se esta sugestão tivesse sido aplicada na realidade.

Já o comparativo dos 30 dias, no Gráfico 5, apresentou sucesso na predição de 7 das 8 empresas analisadas. O algoritmo teria sugerido o investimento para um ganho de 40,33%, levando em conta apenas as predições de lucro. Caso esta sugestão tivesse sido aplicada na realidade, o lucro obtido teria sido de 41,36%.

Por fim, o Gráfico 6 do comparativo dos 60 dias apresentou sucesso na predição de 8 das 8 empresas analisadas. O algoritmo, levando em conta apenas as predições de lucro, teria sugerido o investimento para um ganho de 67,23%. Caso esta sugestão tivesse sido aplicada na realidade, o lucro obtido teria sido de 69,06%.

Pode-se notar que em todos os gráficos do modelo implementado obteve-se lucro em suas predições, sendo que a taxa de lucro mais alta foi obtida através do uso de 60 dias para testes.

6. Materiais e Métodos

Como mencionado anteriormente, o código inicial foi obtido por meio do Github, o mesmo foi escrito em Python e nele utilizou-se a biblioteca *hmmlearn*, importando o tipo *hmm.GaussianHMM*. O algoritmo usado para resolver o problema 1, de probabilidade inicial, foi o *Forward*, para resolver o problema 2, de emissão de probabilidade, foi o *Viterbi* e, para resolver o problema 3, de treinamento, foi o *Baum-Welch*.

Nas oito ações selecionadas foram utilizados o preço de abertura, preço de fechamento, mais alto e mais baixo como estados do HMM. Utilizando o preço de fechamento do dia seguinte como alvo da implementação. Cada ação teve uma base de dados de 2, 7 e 10 anos analisadas começando pela data mais antiga para a mais nova.

Um pacote adicional necessário para a execução do código é o NumPy, pacote fundamental para computação científica em Python [NumPy 2018]. Outro pacote dependente para este script em Python, voltado para a geração de gráficos para a análise dos resultados, foi o Gnuplot.py que faz interface com o gnuplot, popular programa de plotagem de código aberto [Gnuplot 2018].

7. Conclusão

Neste artigo foi apresentado o método *Hidden Markov Model* para predição do mercado de ações com o objetivo de prever o preço de fechamento da ação do próximo dia de pregão da bolsa de valores. Foram testadas oito bases de dados de aproximadamente 2, 7 e 10 anos, onde foram realizadas 4 baterias de testes.

Os testes mostraram que o algoritmo pode ser usado para prever valores futuros das ações da bolsa de valores. Independentemente da quantidade de dias que se deseja testar (7, 15, 30 ou 60) o algoritmo tem capacidade de indicar lucratividade. É importante ressaltar que o algoritmo desenvolvido neste trabalho teve êxito em todas as predições de empresas que sofreram quedas significativas em sua história, como é o caso da Petrobras [Henriques, E. & Vilela, P. 2018]. A previsão da empresa Vale S.A, onde foram

analisados 10 anos de dados, foi bem efetiva devido ao grande conjunto de dados para a predição, bem como a empresa Grendene que teve uma base de dados de 7 anos analisada.

Com isso, pode-se dizer que quanto mais dias usados para testar o modelo HMM, maior é a precisão no resultado, porém com 7 dias já se obtém lucro. Com poucos dias para teste, o HMM já consegue obter o padrão que os dados seguem, porém pode acontecer que naquele determinado período escolhido para obter os dados, aconteça algo inesperado na ação, fazendo com que a mesma não siga aos padrões anteriores e consequentemente, os resultados sejam inconsistentes. Por essa razão, aconselha-se usar uma quantidade maior de dias para teste, para se obter maior lucro e um resultado mais preciso.

Dentre os trabalhos relacionados citados no Capítulo 8, uma técnica que poderia ser utilizada para melhorar os resultados, quando usada uma menor quantidade de dias para teste, poderia ser a análise de *wavelets* proposta por [de Souza e Silva, Legey, & de Souza e Silva 2010]. *Wavelets* propõe remover os preços de alta frequência considerados como ruídos ou *outliers*, os resultados do trabalho mostraram-se ser satisfatórios para o apoio a decisão no pregão. O mesmo poderia ser implementado juntamente com o HMM Semi-contínuo de [Huang, X. D. & Jack, M. A 2004], pois seus resultados mostraram ser, para o reconhecimento de sinais de fala, melhores que o contínuo e discreto, podendo, assim, ter uma taxa alta de acertos para a predição do mercado de ações também.

8. Referência Bibliográfica

Nos trabalhos analisados, como em [Hassen & Nath 2005] o HMM foi usado para gerar previsões dos preços das ações referente a um dia de pregão, determinando os parâmetros ocultos, também chamado de estados ocultos, a partir dos parâmetros observáveis, também chamado de sequência de observações, ou seja, foi localizado os padrões de uma base de dados que correspondem ao comportamento atual do preço das ações usando quatro estados de entrada (preço de abertura, de fechamento, preço mais alto e mais baixo) e, em seguida, juntou-se esses dois conjuntos de dados com elementos dos preços vizinhos e previu-se o preço das ações do dia seguinte da variável de interesse (preço de fechamento). Usou-se o algoritmo *forward-backward* para calcular a probabilidade $P(O | \lambda)$, o algoritmo *Viterbi* para resolver o problema de emissão de probabilidade e o algoritmo *Baum-Welch* para treinar o HMM. O método proposto usando HMM para predição do mercado de ações é explicável, pois os resultados mostraram um potencial alcance da realidade no uso do HMM.

Em [Md Rafiul Hassan, Nath, & Kirley 2007] combinou-se o HMM usado em [Hassan & Nath 2005] com Rede Neural Artificial (ANN) e Algoritmos Genéticos (GA) para obter uma melhor performance que o modelo básico, onde somente um HMM é usado. A rede neural artificial, que é capaz de realizar o aprendizado de máquina e, também, o reconhecimento de padrões, é usada para transformar as entradas do HMM usando o algoritmo *feedforward*. Já o algoritmo genético foi usado para buscar uma melhor otimização dos parâmetros iniciais do HMM transformados pela rede neural artificial, para aqueles que são desconhecidos ou difíceis de identificar. Apesar desse método de fusão mostrar melhor performance, o mesmo é bastante complexo, e para simplificá-lo, os Autores escolheram o número de estados usado em [Hassan & Nath 2005] como atributos numéricos nos vetores de entradas da rede. Esta decisão pode não ser adequada para alguns casos, pois necessita-se de outro GA para encontrar a melhor arquitetura de HMM para um determinado conjunto de dados.

Já em [Md Rafiul Hassan 2009], combinou-se o HMM e a Lógica *Fuzzy* (FL) para melhorar a precisão da previsão para conjuntos de dados de mercado de ações não-estacionários. A FL é usada para gerar o valor da previsão de um dia à frente, enquanto o HMM é usado para identificar padrões de dados semelhantes e agrupá-los com base nessas similaridades. A abordagem combina: *i*) o método de identificação de padrões de dados do HMM, para particionar o espaço de dados; *ii*) a geração da lógica *Fuzzy*, para a previsão de dados de séries temporais financeiras multivariadas. Os resultados apresentados pelos Autores mostram que a precisão das previsões para os dados de ações usando a combinação do modelo HMM e *Fuzzy* atingiu um maior nível de precisão comparado ao modelo de fusão de base em [Md Rafiul Hassan et al. 2007].

Em [Rafiul Hassan, Nath, Kirley, & Kamruzzaman 2012] foi utilizado uma combinação com HMM-*Fuzzy* e Algoritmo Evolutivo Multiobjetivo (EA), técnica utilizada para problemas de otimização, onde o HMM é usado para classificar os dados conforme o padrão, para que as regras *Fuzzy* possam ser geradas usando os dados classificados. Após, é usado o algoritmo *top-down*, algoritmo de divisão e conquista, para controlar a geração de regras difusas para atender o Erro Médio Quadrático (MSE). Na última etapa, o EA multiobjetivo seleciona o MSE adequado para o conjunto de dados. Os resultados experimentais demonstram que o modelo é capaz de gerar um número reduzido de regras *Fuzzy* com desempenho similar, e em alguns casos melhor, comparado com outros modelos.

Os Autores, em [Md Rafiul Hassan, Ramamohanarao, Kamruzzaman, Rahman, & Maruf Hossain 2013], propuseram um novo tipo de sistema de inferência difusa adaptativa (AFIS). Da mesma maneira que no trabalho anterior analisado, o HMM é usado para classificar os dados conforme o padrão, e as regras *Fuzzy* são usadas para agrupar esses dados classificados com valor mais alto de semelhança. O diferencial é que o sistema se adequa dinamicamente as regras difusas conforme a chegada de novos dados. Devido ao mecanismo de adaptação inteligente, o modelo apresenta um desempenho melhor do que outro modelo proposto com modelo *Fuzzy* estático, em [Md Rafiul Hassan 2009].

A pesquisa apresentada em [Cao, Li, Coleman, Belatreche, & McGinnity 2015] utilizou HMM Adaptativo com Estados de Anomalias (AHMMAS), modelo proposto para detectar anomalias nos preços de compra e venda, juntamente com as transformadas *Wavelets*. Estas, apresentam uma abordagem para decomposição hierárquica de funções que modifica a representação de dados de maneira que estes possam ser facilmente analisados, como método de extração de características. Os experimentos de avaliação realizados em sete ações da NASDAQ e da Bolsa de Valores de Londres e 10 preços de ações simulados por equações mostram que o modelo AHMMAS proposto pode efetivamente detectar padrões de manipulação de preços.

Em [de Souza e Silva, Legey, & de Souza e Silva 2010] foi desenvolvido uma metodologia de previsão dos preços do petróleo bruto que consiste basicamente em três etapas. Primeiro, emprega-se a análise de *wavelets* para remover movimentos de preços de alta frequência, que podem ser considerados como ruídos. Em seguida, o HMM é usado para prever a distribuição de probabilidade do retorno de preço acumulado nos próximos n dias. Finalmente, a partir dessa distribuição, infere-se tendências futuras dos preços. Os resultados indicaram que a metodologia proposta pode ser uma ferramenta útil de apoio à decisão para agentes que participam do mercado de petróleo bruto.

Em [Huang, X. D. & Jack, M. A 2004] foi usado um HMM Semi-contínuo para reconhecimento de sinais de fala. A função de densidade de probabilidade de saída semicontínua é representada por uma combinação das probabilidades de saída discretas do modelo e as funções de densidade Gaussianas contínuas de uma mistura de densidade Gaussiana de um *codebook*. A quantidade de dados de treinamento necessários, bem como a complexidade computacional do HMM Semi-contínuo, pode ser significativamente reduzida em comparação com o HMM contínuo. São incluídos resultados experimentais que mostram que a precisão de reconhecimento do modelo HMM Semi-contínuo é mais alta que o discreto e o contínuo.

No estudo proposto em [Fornaciari & Grillenzoni 2017], procura-se comparar sistemas de negociação *on-line* baseados em modelos estatísticos e fornecer soluções técnicas para melhorar seu desempenho. O trabalho compara os métodos de *Markov Switching* (MS), que visa capturar a dinâmica dos valores das ações por uma variável de estado oculta, conceito do Processo de *Markov* e de Parâmetros Variáveis no Tempo (TVP), que trabalha com técnicas de média móvel (MA) que são amplamente utilizadas por investidores, pois é calculada através do preço médio de uma ação ao longo de um determinado período. Contudo, os métodos TVP projetados dão melhores resultado do que MS, devido à sua simplicidade e robustez, porque envolvem apenas dois coeficientes.

Em [Zhang, Aggarwal, & Qi 2017], foi proposta uma rede neural recorrente, chamada Memória de Frequência de Estado (SFM), para capturar os padrões de negociação de frequência múltipla. Uma previsão de curto prazo depende dos padrões de alta frequência para modelar a alta volatilidade das séries temporais de preços e uma previsão a longo prazo é mais relevante para padrões com baixas frequências. Assim, o SFM decompõe os estados da memória em diferentes estados de frequência, de modo que eles possam aprender explicitamente as dependências dos padrões de baixa e alta frequência. Com os componentes de frequência aprendidos, o modelo proposto pode acompanhar a tendência do periódico e gerar uma previsão confiável sobre os preços das ações.

Já em [Hu, Tang, Zhang, & Wang 2018], utilizou-se redes neurais artificiais otimizadas com *Google Trends*, ferramenta de pesquisa *on-line* que permite ao usuário ver com que frequência palavras-chave específicas, assuntos e frases foram consultados durante um período de tempo específico, dessa forma, os dados podem ser usados para antecipar certas tendências futuras. O algoritmo usado foi o Cosseno Senoidal Melhorado (ISCA), para otimizar os pesos e a base da Rede Neural *Back Propagation* (BPNN), método usado para treinar redes neurais. Os resultados indicam que o modelo híbrido ISCA-BPNN supera o BPNN e que as taxas de acerto para o ISCA-BPNN com o *Google Trends* alcançaram, em média, 87% para a base de dados analisada.

Referências

- Alkanza. Disponível: <https://www.rico.com.vc/robo-advisor/alkanza>. Acesso: novembro/2018.
- Bússola. Razões para Investir em Robôs Investidores. Disponível: <https://www.bussoladoinvestidor.com.br/razoes-para-investir-em-robos-investidores>. Acesso: novembro/2018.
- Cao, Y., Li, Y., Coleman, S., Belatreche, A., & McGinnity, T. M. (2015). Adaptive hidden Markov model with anomaly states for price manipulation detection. *IEEE*

- Transactions on Neural Networks and Learning Systems*, 26(2), 318–330. <https://doi.org/10.1109/TNNLS.2014.2315042>.
- Cappé, O., Moulines, E. & Rydén, T. Springer Series In Statistics. Inference in Hidden Markov Models, pages 33-34, 2009.
- Chen, F. R., Wilcox, L. D. & Bloomberg, D. S. (1995). A comparison of discrete and continuous hidden Markov models for phrase spotting in text images. *IEEE*. <http://doi.org/10.1109/ICDAR.1995.599022>.
- Cheng, C., Chen, T. & Wei, L. (2010). A hybrid model based on rough sets theory and genetic algorithms for stock price forecasting. *Information Sciences*, 180, 1610–1629. <http://dx.doi.org/10.1016/j.ins.2010.01.014>.
- Clear. Disponível: <https://www.clear.com.br/site>. Acesso: novembro/2018.
- Corretoras. Disponível: <https://trovoacademy.com/dinheiro/maiores-corretoras-do-brasil>. Acesso: novembro/2018.
- De Souza e Silva, E. G., Legey, L. F. L., & de Souza e Silva, E. A. (2010). Forecasting oil price trends using wavelets and hidden Markov models. *Energy Economics*, 32(6), 1507–1519. <https://doi.org/10.1016/j.eneco.2010.08.006>.
- Distribuição e probabilidade. Disponível: http://www.producao.ufrgs.br/arquivos/disciplinas/489_estaind005_distprob.pdf. Acesso: outubro/2018.
- Easynvest. Disponível: <https://www.easynvest.com.br>. Acesso: novembro/2018.
- Estacionariedade. Disponível: <http://www.portalaction.com.br/series-temporais/11-estacionariedade>. Acesso: novembro/2018.
- Forney, G. D. (1973). The viterbi algorithm. *Proceedings of the IEEE*, 61, 268–278. <https://doi.org/10.1109/PROC.1973.9030>.
- Fornaciari, M., & Grillenzoni, C. (2017). Evaluation of on-line trading systems: Markov-switching vs time-varying parameter models. *Decision Support Systems*, 93, 51–61. <https://doi.org/10.1016/j.dss.2016.09.005>.
- Github. Disponível: <https://github.com/jmschrei/yahmm/wiki>. Acesso: julho/2018.
- Gnuplot. Disponível: <http://gnuplot-py.sourceforge.net>. Acesso: outubro/2018.
- Gráfico contínuo. Disponível: <https://ericlippert.com/2012/02/21/generating-random-non-uniform-data>. Acesso: novembro/2018.
- Harvard University. Disponível: <http://iacs-courses.seas.harvard.edu/courses/am207/blog/lecture-18.html>. Acesso: outubro/2018.
- Hassan, M. R. (2009). A combination of hidden Markov model and fuzzy model for stock market forecasting. *Neurocomputing*, 72(16–18), 3439–3446. <https://doi.org/10.1016/j.neucom.2008.09.029>.
- Hassan, M. R., & Nath, B. (2005). Stock market forecasting using hidden Markov model: a new approach. *5th International Conference on Intelligent Systems Design and Applications (ISDA '05)*, 192–196. <https://doi.org/10.1109/ISDA.2005.85>.

- Hassan, M. R., Nath, B., & Kirley, M. (2007). A fusion model of HMM, ANN and GA for stock market forecasting. *Expert Systems with Applications*, 33(1), 171–180. <https://doi.org/10.1016/j.eswa.2006.04.007>.
- Hassan, M. R., Ramamohanarao, K., Kamruzzaman, J., Rahman, M., & Maruf Hossain, M. (2013). A HMM-based adaptive fuzzy inference system for stock market forecasting. *Neurocomputing*, 104, 10–25. <https://doi.org/10.1016/j.neucom.2012.09.017>.
- Henriques, E. & Vilela, P. (2018). A Petrobras após os escândalos de corrupção da operação Lava-Jato: uma análise do valor da empresa até 2016. <http://dx.doi.org/10.18316/desenv.v7i3.4664>.
- Hmmlearn. Disponível: <https://hmmlearn.readthedocs.io/en/latest/tutorial.html>. Acesso: junho/2018.
- Hu, H., Tang, L., Zhang, S., & Wang, H. (2018). Predicting the direction of stock markets using optimized neural networks with Google Trends. *Neurocomputing*, 285, 188–195. <https://doi.org/10.1016/j.neucom.2018.01.038>.
- Huang X, Ariki Y, Jack M (1990), Hidden Markov Models for speech recognition. Edinburgh University Press.
- Huang, X. D. & Jack, M. A (2004). Semi-continuous hidden Markov models for speech signals. [https://doi.org/10.1016/0885-2308\(89\)90020-X](https://doi.org/10.1016/0885-2308(89)90020-X).
- I Know First. Disponível: <https://iknowfirst.com>. Acesso: novembro/2018.
- Infomoney. Disponível: <https://www.infomoney.com.br/conteudo-patrocinado/noticia/7134609/clear-traz-robo-para-automatizar-operacoes-eliminados-maiores-problemas>. Acesso: novembro/2018.
- Jelinek F, Kaufmann M, Mateo C S (1990), Selforganized language modelling for speech recognition, in Readings in Speech Recognition (Eds. Alex Waibel and Kai-Fu Lee), Morgan Kaufmann, San Mateo, California, pp. 450-506.
- Krogh, A., Brown, M., Mian, S., Sjölander, K. & Haussler, D. (1994). Hidden Markov Models in Computational Biology: Applications to Protein Modeling. *Journal of Molecular Biology*, Vol. 235, pp. 1501-1531. <https://doi.org/10.1006/jmbi.1994.1104>
- Liebert, M. A. (2004). Use of runs statistics for pattern recognition in genomic DNA sequences. *Journal of Computational Biology*, Vol. 11, pp. 107-124.
- Magnetis. Disponível: <https://magnetis.com.br>. Acesso: novembro/2018.
- Mercado de ações. Disponível: <https://www.infomoney.com.br/infograficos/mercado-de-acoes>. Acesso: novembro/2018.
- Mirae. Disponível: <https://corretora.miraeasset.com.br>. Acesso: novembro/2018.
- Moghaddam, A., Moghaddam, M. & Esfandyari, M. (2016). Stock market index prediction using artificial neural network. *Journal of Economics, Finance and Administrative Science*, 21, 89-93. <http://dx.doi.org/10.1016/j.jefas.2016.07.002>.
- Monetus. Disponível: <https://monetus.com.br>. Acesso: novembro/2018.
- NumPy. Disponível: <http://www.numpy.org>. Acesso: julho/2018.

- Rabiner, L. R. (1989). A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of the IEEE*. <https://doi.org/10.1109/5.18626>.
- Rafiul Hassan, M., Nath, B., Kirley, M., & Kamruzzaman, J. (2012). A hybrid of multiobjective Evolutionary Algorithm and HMM-Fuzzy model for time series prediction. *Neurocomputing*, 81, pp. 1–11. <https://doi.org/10.1016/j.neucom.2011.09.012>.
- Robô Investe. Disponível: <https://roboinveste.com.br>. Acesso: novembro/2018.
- Robôs. Disponível: <https://smarttbot.com/blog/como-criar-um-robo-investidor>. Acesso: novembro/2018.
- Sinais. Disponível: <http://www.cear.ufpb.br/juan/wp-content/uploads/2016/08/Aula-2-Sinais-Continuas-e-Discretas.pdf>. Acesso: novembro/2018.
- Smarttbot. Disponível: <https://ajuda.smarttbot.com/hc/pt-br/articles/360007955574--Em-quais-corretoras-meus-robô-podem-operar-Qual-corretora-você-me-recomendam>. Acesso: novembro/2018.
- Teja Darbha, R., Jain, A. & Mehta, P. (2018). Stock Forecasting using Hidden Markov Models. Disponível: <https://github.com/ayushjain1594/Stock-Forecasting>, Fevereiro/2018.
- TradeMachine. Disponível: <https://www.trademachine.co>. Acesso: novembro/2018.
- Trovoacademy. Disponível: <https://trovoacademy.com/dinheiro/robo-de-investimento-vale-a-pena>. Acesso: novembro/2018.
- Variáveis. Disponível: <https://mundoeducacao.bol.uol.com.br/matematica/variaveis-na-estatistica.htm>. Acesso: outubro/2018.
- Variáveis Latentes. Disponível: <https://pt.scribd.com/doc/15886719/Dicionario-de-tipos-de-variaveis>. Acesso: novembro/2018.
- Verios. Disponível: <https://verios.com.br>. Acesso: novembro/2018.
- Vinciarelli, A. & Luetin, J. (2000). Off-line cursive script recognition based on continuous density HMM, Proceedings of the 7th International Workshop on Frontiers in Handwriting Recognition, Amsterdam, pp. 493-498.
- Warren. Disponível: <https://warrenbrasil.com>. Acesso: novembro/2018.
- Xie H, Anrae P, Zhang M, Warren P (2004), Learning Models for English Speech Recognition, Proceedings of the 27th Conference on Australasian Computer Science, pp. 323-329.
- XP. Disponível: <https://www.xpi.com.br/>. Acesso: novembro/2018.
- Yacoubi A., Sabourin R., Gilloux M., Suen C. Y. (1999). An HMM-based approach for off-line unconstrained handwritten word modeling and recognition. Knowledge Techniques in Character Recognition, CRC Press LLC. <http://doi.org/10.1109/34.784288>.
- Zhang, L., Aggarwal, C., & Qi, G.-J. (2017). SFM for stock prediction. *Kdd*. <https://doi.org/10.1145/3097983.3098117>.