Maximização do valor de portfólio de ações da Bolsa de Valores de São Paulo usando um sistema de múltiplos agentes autônomos racionais baseado em perfis de investidores

Renato Nobre
Departamento de Ciência da Computação
Universidade de Brasília
Brasília, Brasil
rekanobre@gmail.com

Khalil Carsten
Departamento de Ciência da Computação
Universidade de Brasília
Brasília, Brasil
khalilcarsten@gmail.com

I. Introdução

Todos os dias ações são transacionadas em grande escala em mercados de ações ao redor do mundo. Tais mercados podem ser considerados ambientes dinâmicos, com inúmeras incertezas existentes. Estes devido a diversos fatores que interagem com o mercado, como eventos políticos, condições econômicas, e expectativas de negociadores [10]. Neste cenário de mercado, a capacidade de um investidor prever a direção de fechamento de algum valor de uma determinada ação, permite a investidores a possibilidade de criar estratégias para negociar ações e, possivelmente, aumentar seu lucro.

Para consolidar esta tarefa complexa e sensível a falhas, seria interessante apresentar abordagens computacionais da área de aprendizagem de máquina. Há atualmente na literatura acadêmica uma variedade de trabalhos relacionados a modelos de aprendizagem em mercado de ações. Mas foi percebido que sua grande maioria são voltados a mercados consolidados, como o Nasdaq, e poucos trabalhos com mercados emergentes que é o caso da Bolsa de Valores de São Paulo (BOVESPA). Outra falta também notada na literatura, diz respeito ao funcionamento destes sistemas desenvolvidos, a maioria dos sistemas funcionam como classificadores, apenas tomando decisões com base no que foi apreendido. No entanto, esses sistemas não realizam ações independentes, surgindo assim a necessidade de criar sistemas atuadores, que após a classificação realize o ato de comprar, manter ou vender uma ação, sem que haja em nenhum momento a necessidade de interação humana.

Visto os problemas apresentados, busca-se realizar um sistema baseado em agentes autônomos racionais, capazes de realizar compras e vendas de ações no BOVESPA, maximizando seu lucro e se adaptando em diversas situações.

II. Proposta

Este trabalho propõe uma abordagem para maximizar o valor do portfólio do investidor no mercado de ações do IBOVESPA, baseada em um sistema multiagente racional capaz de atuar de forma autônoma, utilizando como informação diversos indicadores econômicos e informações de tendência externas, e agindo de forma adaptativa ao seu perfil de investidor pré definido.

Nota-se que a abordagem não busca maximizar o valor de uma única ação e lucrar somente sobre ela. O agente contará com um conjunto de ações diversas, tais quais definem o portfólio do investidor. Logo, o agente buscará selecionar o melhor portfólio e dentro do mesmo, tentar maximizar o lucro absoluto.

Outro detalhe importante da proposta, é que o agente vai possuir em sua definição interna, um perfil de investidor. Quando fala-se desse perfil, queremos que o agente aja de forma mais próxima do perfil de seu usuário, em relação ao risco no qual o usuário está disposto a correr para aumentar seus lucros. Na qual poderá ser uma das seguintes: Conservador, Moderado, Agressivo.

Em suma, a inovação proposta neste artigo é a possibilidade de um agente atuador agir de forma autônoma sobre o mercado de ações, de forma a maximizar o valor de seu portfólio, e trabalhar em conjunto com outros agentes de forma cooperativa para maximizar ganhos do grupo.

III. Metodologia

Com base na Seção II percebe-se que ainda precisa definir detalhes de como essa metodologia vai ser implementada e de onde os dados vão ser retirados. Desenvolveremos melhor então cada ponto da proposta.

Primeiro precisamos definir o que é um agente. Consideraremos um agente como uma entidade que percebe o ambiente por meio de sensores e atua neste ambiente, de forma autônoma, por meio de atuadores [13]. No contexto do problema, temos então que nosso agente seria um investidor, cadastrado em uma corretora de investimentos, que atua no ambiente, o IBOVESPA, com atos de compras e vendas de ações, e recebe a informação do ambiente por via dos dados

gerados pela plataforma de investimento e pelo retorno de suas ações.

Visto que temos a definição de um agente capaz de atuar de forma autônoma, precisamos desenvolver o conceito de racional. Para o escopo deste trabalho o classificamos como agentes que buscam ações que maximizam uma determinada medida de desempenho, que no casso do problema apresentando é o lucro do portfólio.

Quando múltiplos agentes coexistem e interagem em um ambiente dinâmico de maneira cooperativa ou competitiva, denominamos o sistema como multiagente. Observe que o principio do mercado de ações é um ambiente inerentemente competitivo, onde os agentes visam gerar lucro em cima de outros. No entanto, queremos testar a abordagem de diversos agentes homogêneos agindo de forma competitiva para influenciar tendencias de mercado e maximizar o lucro do grupo.

Precisamos definir agora quais dados e informações que alimentarão o sistema. A Seção II, menciona informações de duas fontes diferentes, indicadores econômicos e tendencias externas. Os indicadores econômicos são grandezas de carácter econômico, expressas em valor numérico, cuja principal utilidade consiste na aferição dos níveis das ações. Tais indicadores serão usados como principal conjunto de atributos da função de aprendizagem do sistema, e serão calculados com dados fornecidos pelo IBOVESPA de 1986 a 2018 [1].

Tendo em vista que todas as terminologias para o entendimento da proposta foram definidas e as fontes de dados foram apresentadas. Cabe então definir as ferramentas utilizadas para construir o sistema. A base do sistema será o simulador do IBOVESPA, que vai ser construído com os dados de cotações históricas, sendo considerado uma unidade de tempo t um dia no mercado. Desta forma o agente, imerso nesse simulador, aprenderá uma politica capaz de descobrir se no instante $t+1$ os valores de uma determinada ação vão aumentar ou diminuir. O agente após identificar a direção do índice de cada uma de suas ações contidas no seu portfólio, deverá então decidir como alocará seus recursos de forma a tentar maximizar o valor do mesmo, realizando as devidas transações.

O modelo de aprendizado do agente funcionará com uma abordagem por reforço, pois ao analisar a proposta descrita, observa-se que é o modelo mais intuitivo para utilizar, o algoritmo de aprendizagem ainda está para ser definido. No entanto, podemos definir três passos de aprendizagem de antemão. O agente precisará aprender a selecionar as melhores inferências econômicas para levar em consideração, com informação dessas inferências o agente deve aprender a prever quando uma ação vai valorizar ou desvalorizar no instante $t+1$, e por fim, sabendo a direção estimada dos índices das ações deve aprender a realocar seus recursos de forma a maximizar o valor de seu portfólio.

A. Deep Q Learning

Quando trata-se de aprendizado por reforço pensamos em em pares estados-ação onde o agente fara sua exploração sabendo em qual estado está e qual será sua recompensa passando para um próximo estado através de uma ação. Porém estamos tratando do mercado financeiro, um ambiente muito estocástico, portando a definição de estados torna-se impossível tendo em vista que cada janela de valores, presentes nas ações, mudam completamente com o tempo gerando estados completamente aleatórios. O desafio se encontra em gerar uma politica de ação para o agente. Sem esses estados bem definidos torna-se inviável a execução de um aprendizado por reforço. Para isso utilizaremos uma método chamado Deep Q Learning (Figura2) que acrescenta uma rede neural como geradora da politica de ações desse agente visando uma solução para o problema anteriormente citado.

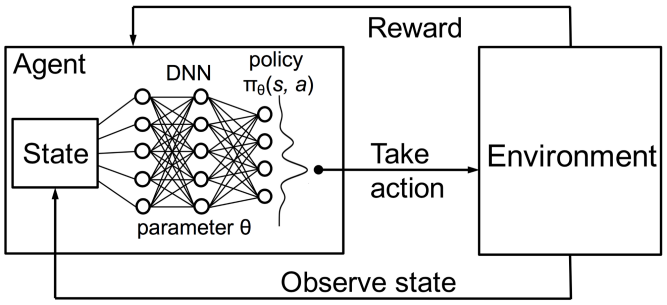

Figura 1. Diagrama de funcionamento Deep Q Learning

Como visto na figura 2 o agente se relaciona com o ambiente através de uma politica gerada pela própria rede neural. Assim o nosso estado sendo um vetor de indicadores ou uma janela dos valores gerará uma ação ao final da passagem pela rede neural. Essa ação será executada pelo agente no ambiente o que retornará uma recompensa pela ação executada. A recompensa é utilizada para dizer o quão bom ou ruim foi esta ação e o valor final é repassado para a rede neural como em um treino supervisionado.

$$y_j = r + \gamma * max(\phi(j+1)), \quad \text{onde } \phi\text{é a ação a ser tomada retornado pela DNN e r é valor de recompensa} \tag{1}$$

Para o nosso problema utilizamos uma equação de recompensa baseada na quantidade de dinheiro na carteira em instantes t. O dinheiro na c_t sendo t o instante logo após efetuado a ação e c_{t-1} o instante logo antes de efetuar a ação.

$$r = \log \frac{c_t}{c_{t-1}} \tag{2}$$

B. Rede Neural

Para efetuar a escolha de política utilizamos uma rede neural composta por 4 camadas onde as três primeira são compostas por camadas com ativação RELU, com 64, 16, 8, 3 nós e a última om ativação linear composta por 3 nós. A rede é basicamente curta com aproximadamente 23.000 parâmetros.

[1]Disponível em: http://www.bmfbovespa.com.br/

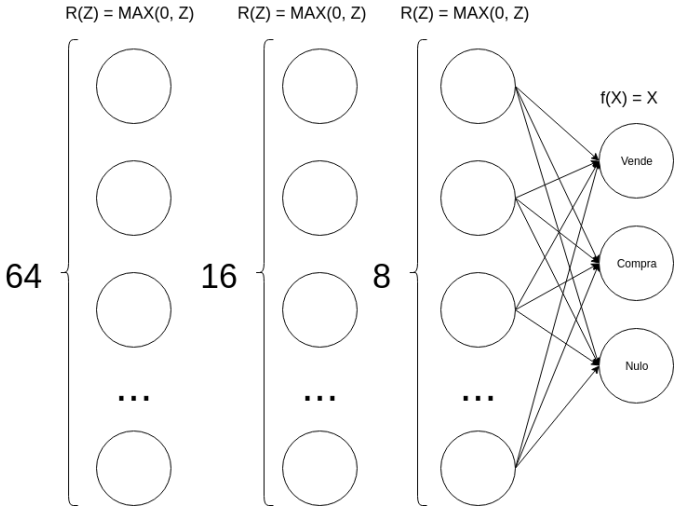

Figura 2. Diagrama da Deep Neural Network utilizada no DQN

1) Inicialização de pesos: Todos os pesos de todas as camadas foram inicializados de maneira uniformemente aleatória tentando influenciar o menos possível as ações inciais do agente.

C. Algorimto DQN

Como princípio para o aprendizado em reforço necessitamos de uma ambiente. Esse ambiente é o mesmo que o de uma bolsa de valores, ou seja, temos ações que mudam de preço com o passar do tempo e podemos comprá-las ou vendê-las transacionando o dinheiro que temos em carteira. Para simular o mercado utilizamos inicialmente um simulador disponível para *Python* nomeado *BackTrader*. Infelizmente esse simulador se tornou muito complexo e pouco adaptável ao projeto, então optamos por escrevermos o nosso próprio com funcionalidades direcionadas para o nosso objetivo. Tendo isso em vista agora temos pronto o nosso ambiente de atuação do agente podemos apresentar o algoritmo original utilizado pelo projeto Figura 8. O código do algoritmo foi desenvolvido e adaptados de diversos códigos anteriores.

Algorithm 1 Deep Q-learning with Experience Replay

Initialize replay memory $\mathcal{D}$ to capacity N
Initialize action-value function Q with random weights
for episode $= 1, M$ **do**
  Initialise sequence $s_1 = \{x_1\}$ and preprocessed sequenced $\phi_1 = \phi(s_1)$
  for $t = 1, T$ **do**
    With probability ϵ select a random action a_t
    otherwise select $a_t = \max_a Q^*(\phi(s_t), a; \theta)$
    Execute action a_t in emulator and observe reward r_t and image x_{t+1}
    Set $s_{t+1} = s_t, a_t, x_{t+1}$ and preprocess $\phi_{t+1} = \phi(s_{t+1})$
    Store transition $(\phi_t, a_t, r_t, \phi_{t+1})$ in $\mathcal{D}$
    Sample random minibatch of transitions $(\phi_j, a_j, r_j, \phi_{j+1})$ from $\mathcal{D}$
    Set $y_j = \begin{cases} r_j & \text{for terminal } \phi_{j+1} \\ r_j + \gamma \max_{a'} Q(\phi_{j+1}, a'; \theta) & \text{for non-terminal } \phi_{j+1} \end{cases}$
    Perform a gradient descent step on $(y_j - Q(\phi_j, a_j; \theta))^2$ according to equation 3
  end for
end for

Figura 3. Pseudo-código algoritmo DQN

D. Dados e características

Como citado anteriormente esse projeto trata-se da bolsa de valores de São Paulo, então adquirimos os dados a partir da série histórica de todas as ações disponibilizada pelo site bmfbovespa .

Desses dados conseguimos preços de todas as ações desde 1987 a 2017 presentes na Bovespa. Por um princípio de alcançarmos um comportamento mais estável do mercado utilizamos dados somente a partir do ano de 1995, um ano após a instauração do pregão eletrônico na bolsa de valores.

A partir desses dados optamos por duas possibilidades de características de entrada para o sistema DQN. Primeiramente optamos por um vetor de indicadores gerados a cada interação pelo dados da bolsa e depois por um vetor com os valores de uma janela alocada dentro das ações. A primeira forma gerou um vetor de 7 valores composto pelos indicadores:

- SMA (Simple Moving Avarage)
- RSI (Relative Strength Index)
- MACD (Moving Average Convergence Divergence)
- CCI (Commodity Channel Index)
- AO (Awesome Oscillator)
- WMA (Weighted Moving Average)
- SMI (Stochastic Momentum)

O segundo recupera os valores de ações em uma janela definida inicialmente no algoritmo e será mais detalhada suas escolhas na próxima seção.

IV. Solução e análise

Nosso experimento passou por diversas etapas, onde precisamos adaptar nossas escolhas inciais a cerca de conseguir algum resultado favorável. Nosso experimento primário veio em cima da ação da empresa VALE nomeada VALE3. Nessa executamos o treino com 1000 épocas e utilizando como característica o vetor de 7 indicadores nos anos de 1995 a 2014. O teste foi feito em cima dos anos de 2016 infelizmente notamos que a rede estava em uma estado de *overfitting* pois a única ação que executava era a de venda. Durante o teste guardávamos os modelos de 10 em 10 épocas o que possibilitou que fizemos uma análise nos modelos intermediários antes do modelo final. Realizamos testes nos modelos intermediários e notamos que o *overfitting* estava presente desde o inicio. Devido a isso fizemos testes alterando as taxas de aprendizado da rede e do gamma. presente na função de aprendizado.

Nessa etapa supomos que o problema poderia ser na função de recompensa na qual era somente o lucro ou prejuízo obtido na compra e na venda de uma ação. Então criamos a carteira do agente no ambiente e fizemos a função de recompensa em 2. Também não houve diferença ao trocar a função de recompensa então teríamos que olhar mais a fundo na rede neural para entendermos o problema.

Com o intuito de simplificar o problema para aumentar o tamanho dos estados, ou seja, o tamanho do vetor de entradas alteramos o vetor de indicadores para a janela de valores das ações. Isso facilitaria a manipulação da entrada a fim de descobrirmos melhor o problema. Testamos com 10, 20, 30

Parâmetro				
Tetativas	1	2	3	4
Gamma	0.8	0.5	0.3	0.1
LR	0.001	0.0001	0.00001	0.000001

Tabela I
PARÂMETROS TENTADOS NO PROJETO PARA CONSEGUIR UM MELHOR RESULTADO

e 50 como entrada e percebemos que não havia diferença, o *overfitting* permanecia. m seguida olhamos diretamente para o vetor de saída da rede a cada interação que ela fazia com o ambiente. Notamos que o *overfitting* já se estabelecia no inicio e independente da diferença dos estados que entravam os pesos na saída permaneciam praticamente os mesmos.

Nessa etapa decidimos multiplicar os valores das ações por 100 e reduzir a taxa de aprendizado da rede para 0.0000001 antes 0.001. Nesse momento houve um certo avanço onde o *overfitting* demorou mais para acontecer. Fizemos testes com o treino de somente 1 época e conseguimos resultados diferentes de somente uma ação para todos os teste. Abaixo alguma tabelas de parâmetros tentados no projeto.

V. RESULTADOS

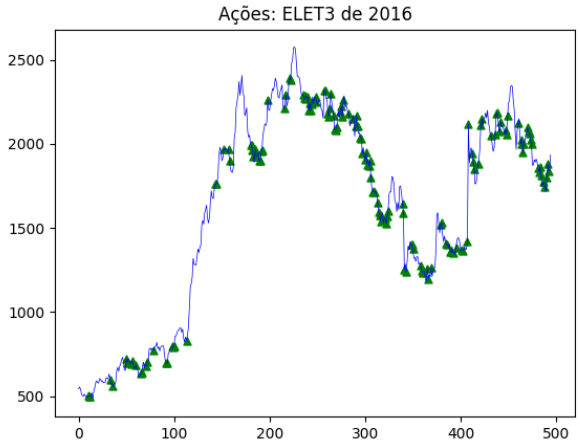

Figura 4. Teste com Ação ELET3 de 2016 compras e vendas

Figura 5. Teste 2 com Ação ELET3 de 2016 compras e vendas

Figura 6. Teste com Ação VALE3 de 2016 compras e vendas

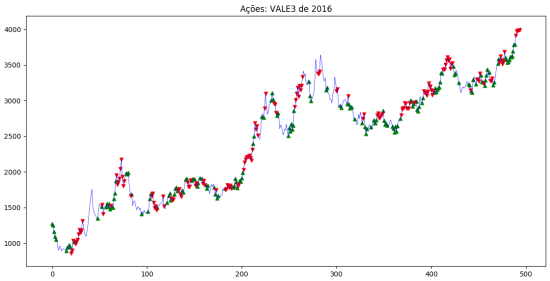

Figura 7. Teste 2 com Ação VALE3 de 2016 compras e vendas

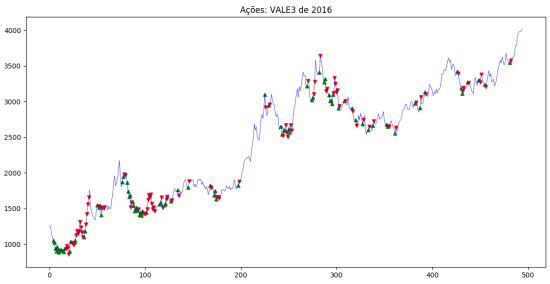

Figura 8. Pseudo-código algoritmo DQN

REFERÊNCIAS

[1] Y. Kara, M. Acar Boyacioglu and Ö. Baykan, "Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the Istanbul Stock Exchange", Expert Systems with Applications, vol. 38, no. 5, pp. 5311-5319, 2011.

[2] T. Preis, H. Moat and H. Stanley, "Quantifying Trading Behavior in Financial Markets Using Google Trend", Scientific Reports, vol. 3, no. 1, 2013.

[3] J. Patel, S. Shah, P. Thakkar and K. Kotecha, "Predicting stock and stock price index movement using Trend Deterministic Data Preparation and machine learning techniques", Expert Systems with Applications, vol. 42, no. 1, pp. 259-268, 2015.

[4] D. Enke and S. Thawornwong, "The use of data mining and neural networks for forecasting stock market returns", Expert Systems with Applications, vol. 29, no. 4, pp. 927-940, 2005.

[5] L. Malagrino, N. Roman and A. Monteiro, "Forecasting stock market index daily direction: A Bayesian Network approach", Expert Systems with Applications, vol. 105, pp. 11-22, 2018.

[6] L. Cocco, G. Concas and M. Marchesi, "Using an artificial financial market for studying a cryptocurrency market", Journal of Economic Interaction and Coordination, vol. 12, no. 2, pp. 345-365, 2015.

[7] F. Paiva, R. Cardoso, G. Hanaoka and W. Duarte, "Decision-making for financial trading: A fusion approach of machine learning and portfolio selection", Expert Systems with Applications, vol. 115, pp. 635-655, 2018.

[8] R. Sutton, Reinforcement Learning. Boston, MA: Springer US, 1992.

[9] L. Yu, W. Shouyang and K. Keung Lai, "Mining stock market tendency using GA-based support vector machines", International Workshop on Internet and Network Economics, Springer, pp. 336-345, 2005.

[10] R. Choudhry and K. Garg, "A hybrid machine learning system for stock market forecasting", World Academy of Science, Engineering and Technology, vol. 2, no. 3, pp. 315-318, 2008

[11] X. Fu, J. Du, Y. Guo, M. Liu, T. Dong and X. Duan, “A Machine Learning Framework for Stock Selection” arXiv preprint, arXiv:1806.01743, 2018
[12] C. Yi Huang, “Financial Trading as a Game: A Deep Reinforcement Learning Approach”, arXiv preprint, arXiv:1807.02787, 2018
[13] S. Russell and P. Norvig, Artificial intelligence. Malaysia, Pearson Education Limited, 2016.