Implementação do Algoritmo LMS para predição de ações da BOVESPA

MATHEUS FARIAS
Universidade Federal de Pernambuco
Departamento de Eletrônica e Sistemas
msf4@cin.ufpe.br

PABLO GODOY
Universidade Federal de Pernambuco
Departamento de Eletrônica e Sistemas
pablogodoy@gmail.com

RAILTON ROCHA
Universidade Federal de Pernambuco
Departamento de Eletrônica e Sistemas
railton.rocha@ufpe.br

***Resumo*— O presente projeto busca aplicar os conhecimentos obtidos na primeira unidade da disciplina de Filtragem Adaptativa para projetar um preditor de ações da bolsa de valores com o uso do algoritmo LMS.**

I. INTRODUÇÃO

Um filtro adaptativo é um sistema com um filtro linear que possui uma função transferência controlada por parâmetros variáveis e uma média que se ajusta aos parâmetros de acordo com seu algoritmo de otimização. Por conta da complexidade de tais algoritmos, quase todos os filtros adaptativos são filtros digitais, no presente projeto, usa-se o algoritmo de otimização LMS (Least Mean-Square).

Algoritmo 1: LMS

Entrada:
x : o vetor de entrada
d: o vetor desejado
μ: a taxa de aprendizagem
N: a ordem do filtro
Saída:
y: a resposta do filtro
e: o erro do filtro
início
  M = *tamanho(x)* ;
  $x_n(0) = w_n(0) = [0\ 0\ \ldots\ 0]^T$;
  repita
    $x_{n+1} = [x(n);\ x_n(1:N)]$;
    $y(n) = w_n^H * x_n$;
    $e(n) = d(n) - y(n)$;
    $w_{n+1} = w_n + 2\mu e(n) x_n$;
  até M;
fim

O LMS é um algoritmo de busca cuja simplificação da computação do vetor gradiente mostra-se possível a partir de uma mudança adequada da função objetivo. Tal algoritmo, assim como outros parecidos, é usado largamente em várias aplicações de filtros adaptativos por conta da sua simplicidade computacional. Sua característica de convergência é analisada com base em estabelecer o intervalo no qual o fator de convergência garantirá a estabilidade. Pode-se mostrar que a velocidade de convergência do LMS depende dos autovalores da matriz de correlação do sinal de entrada.

O projeto proposto considera-se um preditor de ações da bolsa de valores. O diagrama do preditor é mostrado na Figura 1.

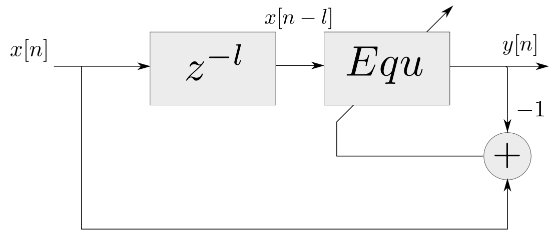

Fig. 1: Equalizador cego adaptativo para uma transmissão digital

É pedido que se projete tal preditor para prever o comportamento de uma determinada ação em até 10 dias, considerando um filtro LMS com ordem 20, 40 e 60 e avaliar o desempenho para cada situação.

O vetor $\mathbf{w_o}$ de coeficientes de um filtro capaz de minimizar o erro quadrático médio na estimativa do sinal de referência $d(k)$ é conhecido como solução de Wiener e pode ser obtido através de,

$$\mathbf{w_o} = \mathbf{R^{-1}p}$$

em que $\mathbf{R} = E[\mathbf{x}(k)\mathbf{x}^T(k)]$ e $\mathbf{p} = E[d(k)\mathbf{x}(k)]$, assumindo que $d(k)$ e $\mathbf{p}(k)$ são ambos WSS. Porém, estimavas precisas desses parâmetros ($\mathbf{R}$ e $\mathbf{p}$) não estão disponíveis na prática, tornando seu cálculo computacionalmente custoso. Portanto, desde que satisfeitas algumas condições, partindo do algoritmo da "descida mais íngreme" (*steepest-descent algorithm*) é possível se aproximar da solução de Wiener através da atualização dos coeficientes, dada por:

$$\mathbf{w}(k+1) = \mathbf{w} - \mu \mathbf{g_w}(k) \tag{1}$$

em que $\mathbf{g_w}(k)$ é o vetor gradiente da função objetivo, e o seu negativo indica a maior variação descrescente possível.

O algoritmo LMS se baseia na Equação 1 utilizando, ao invés do gradiente, um estimador $\hat{\mathbf{g}}_\mathbf{w}(k)$ dado por:

$$\hat{\mathbf{g}}_\mathbf{w}(k) = -2e(k)\mathbf{x}(k)$$

Resultando em,

$$\mathbf{w}(k+1) = \mathbf{w} + 2\mu e(k)\mathbf{x}(k) \tag{2}$$

em que o parâmetro μ deve ser escolhido de modo a garantir a convergência. O Algoritmo 1 ilustra o procedimento geral do algoritmo LMS.

II. Projeto

A. Base de Dados

Para simulação, considera-se a série temporal (em unidades de dia) dos valores reais da ação ABEV3 (ON) ao longo do ano de 2019 (até setembro). Tal informação é facilmente obtida através do site oficial da BOVESPA[2]. Os dados obtidos pelo site da BOVESPA das ações no intervalo escolhido são condensados em um arquivo texto, dele foi filtrado as informações relevantes para o projeto, como o preço da ação e as datas (igualmente espaçadas), definindo o vetor de entrada do sinal $x(k)$, sendo n variando de 0 a 177 (incluso), visto que as datas são sempre igualmente espaçadas. Na Figura 2 pode-se ver o vetor representado de maneira discreta, como é naturalmente, e na Figura 3 mostra-se o gráfico com fit para as flutuações aparecerem com mais característica.

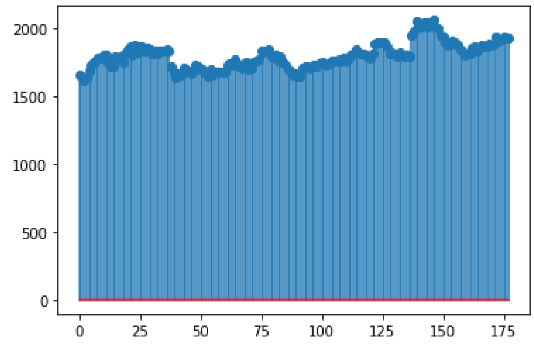

Fig. 2: Sinal x[n] representando o preço por data

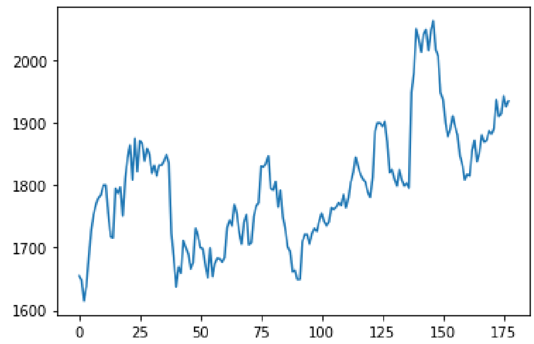

Fig. 3: Sinal x[n] continuamente

B. Parâmetros

Para o projeto do algoritmo LMS, tem-se que primeiramente estabelecer qual taxa de aprendizagem μ será usada, para isso, precisa-se calcular a matriz de correlação do sinal de entrada.

A matriz de correlação $\mathbf{R}$ é definida como $\mathbf{R} = E[\mathbf{x}(k)\mathbf{x}^H(k)]$, onde $\mathbf{x}(k)$ é o vetor que representa o sinal de entrada:

$$\mathbf{x}(k) = [x_0(k)\ x_1(k)\ \ldots\ x_N(k)]^T$$

E $\mathbf{x}^H(k)$ é o seu hermitiano.

Disso, tem-se que a matriz de correlação $\mathbf{R}$ pode ser escrita como:

$$\mathbf{R} = \begin{bmatrix} E[|x_0(k)|^2] & E[x_0(k)x_1^*(k)] & \cdots & E[x_0(k)x_N^*(k)] \\ E[x_1(k)x_0^*(k)] & E[|x_1(k)|^2] & \cdots & E[x_1(k)x_N^*(k)] \\ \vdots & \vdots & \ddots & \vdots \\ E[x_N(k)x_0^*(k)] & E[x_N(k)x_1^*(k)] & \cdots & E[|x_N(k)|^2] \end{bmatrix}$$

Sendo assim, calcula-se a matriz de correlação do sinal de entrada para $N = 3, 4, 5$, respectivamente $\mathbf{R_3}$, $\mathbf{R_4}$ e $\mathbf{R_5}$:

$$\mathbf{R_3} = \begin{bmatrix} 3249134.2 & 3230451.1 & 3211910.1 & 3193417.4 \\ 3230451.1 & 3249134.2 & 3230451.1 & 3211910.1 \\ 3211910.1 & 3230451.1 & 3249134.2 & 3230451.1 \\ 3193417.4 & 3211910.1 & 3230451.1 & 3249134.2 \end{bmatrix}$$

$$\mathbf{R_4} = \begin{bmatrix} 3249134.2 & 3230451.1 & 3211910.1 & 3193417.4 & 3175050.7 \\ 3230451.1 & 3249134.2 & 3230451.1 & 3211910.1 & 3193417.4 \\ 3211910.1 & 3230451.1 & 3249134.2 & 3230451.1 & 3211910.1 \\ 3193417.4 & 3211910.1 & 3230451.1 & 3249134.2 & 3230451.1 \\ 3175050.7 & 3193417.4 & 3211910.1 & 3230451.1 & 3249134.2 \end{bmatrix}$$

$$\mathbf{R_5} = \begin{bmatrix} 3249134.2 & 3230451.1 & 3211910.1 & 3193417.4 & 3175050.7 & 3156332.0 \\ 3230451.1 & 3249134.2 & 3230451.1 & 3211910.1 & 3193417.4 & 3175050.7 \\ 3211910.1 & 3230451.1 & 3249134.2 & 3230451.1 & 3211910.1 & 3193417.4 \\ 3193417.4 & 3211910.1 & 3230451.1 & 3249134.2 & 3230451.1 & 3211910.1 \\ 3175050.7 & 3193417.4 & 3211910.1 & 3230451.1 & 3249134.2 & 3230451.1 \\ 3156332.0 & 3175050.7 & 3193417.4 & 3211910.1 & 3230451.1 & 3249134.2 \end{bmatrix}$$

Sabe-se que o intervalo de escolha para a taxa de aprendizagem μ dado uma matriz de correlação $\mathbf{R}$ com os autovalores λ_1, λ_2, ..., λ_N, é [1]:

$$0 < \mu < \frac{1}{\lambda_{max}} \tag{3}$$

Onde λ_{max} é o maior autovalor encontrado para $\mathbf{R}$. Sendo assim, é preciso primeiramente encontrar os autovalores para cada matriz $\mathbf{R_3}$, $\mathbf{R_4}$ e $\mathbf{R_5}$.

$$\mathbf{R_3} : \begin{bmatrix} 12903456.3 & 63403.8 & 18680.7 & 10996.1 \end{bmatrix}$$

$$\mathbf{R_4} : \begin{bmatrix} 16096961.8 & 97019.8 & 27013.9 & 14287.9 & 10387.8 \end{bmatrix}$$

$$\mathbf{R_5} : \begin{bmatrix} 19278105.5 & 138271.2 & 37092.3 & 18833.1 & 12405.5 & 10097.9 \end{bmatrix}$$

Então, usando a Equação 3, tem-se que os intervalos admitidos para a taxa de aprendizagem para cada matriz de correlação, respectivamente, são:

$$0 < \mu_3 < 7.5 \times 10^{-8}$$

$$0 < \mu_4 < 6.2 \times 10^{-8}$$

$$0 < \mu_5 < 5.2 \times 10^{-8}$$

Usa-se, para facilitar cálculos computacionais, um valor de μ potência de 2, neste projeto foi usado $\mu = 2^{-30}$. De posse do μ, é possível aferir três parâmetros relevantes na análise do projeto do filtro LMS, o tempo de convergência dos coeficientes, o tempo de convergência da saída do filtro e o erro mínimo do filtro.

Para encontrar o tempo de convergência dos coeficientes, usa-se [1]:

$$\tau_w = \frac{1}{2\mu\lambda_{min}} \tag{4}$$

Para encontrar o tempo de convergência da saída do filtro, usa-se [1]:

$$\tau_e = \frac{1}{4\mu\lambda_{min}} \tag{5}$$

Sendo assim, usando a taxa de aprendizagem escolhida e os autovalores das matrizes $\mathbf{R_3}$, $\mathbf{R_4}$ e $\mathbf{R_5}$, tem-se:

$$\tau_{w3} = 48823.5$$

$$\tau_{w4} = 51682.6$$

$$\tau_{w5} = 53166.6$$

$$\tau_{e3} = 24411.7$$

$$\tau_{e4} = 25841.3$$

$$\tau_{e5} = 26583.3$$

A unidade que representa os tempos τ é iteração do algoritmo.

Como um estimador do gradiente é utilizado no filtro LMS, o erro mínimo não é exatamente ótimo devido a um erro em excesso, ξ_{exc}, provocado pela aproximação. Nessas condições, o erro mínimo teórico pode ser encontrado através da Equação 6,

$$\xi_{min} = E[d^2(k)] - 2\mathbf{w}_o^T\mathbf{p} + \xi_{exc} \tag{6}$$

Com $\mathbf{w}_o = \mathbf{R}^{-1}\mathbf{p}$ e $\mathbf{p} = E[d(k)\mathbf{x}(k)]$

Sendo $d(k)$, para o esquema proposto na Figura 1, igual a $x(k)$.

O erro em excesso, ξ_{exc}, pode ser encontrado a partir da Equação 7, uma vez que μ muito pequeno,

$$\xi_{exc} \approx \mu\sigma_n{}^2 tr[\mathbf{R}] = \mu(N+1)\sigma_n{}^2\sigma_x{}^2 \tag{7}$$

em que $\sigma_x{}^2$ é a variância do sinal de entrada e $\sigma_n{}^2$ é a variância do ruído adicional associado ao sistema. Os erros serão calculados na seção de validação do presente projeto.

Algoritmo 2: LMSPred

Entrada:

x : o vetor de entrada

l: a quantidade de dados atrasados

μ: a taxa de aprendizagem

N: a ordem do filtro

Saída:

y: a resposta do filtro

início

$M = tamanho(x_d)$;

$x_n(0) = w_n(0) = [0\ 0\ ...\ 0]$;

$x_d = [0\ 0\ ..\ 0\ x]$;

repita

$x_{n+1} = [x_d(n);\ x_n(1:N)]$;

$y(n) = w_n^H * x_n$;

if $n > M - l$ **then**

$e = 0$;

else

$e(n) = d(n) - y(n)$;

end

$w_{n+1} = w_n + 2\mu e(n) x_n$;

até M;

fim

C. Previsão

Com o arranjo demonstrado na Figura 1 é possível realizar a predição de l dados (dias) seguintes utilizando o Algoritmo 2.

Primeiramente, toma-se a entrada x dos 178 valores obtidos da ABEV3 (ON), mostrados graficamente na Figura 2, a partir dos dados, realiza-se a comparação dos valores de x com os valores da saída do filtro, que é o processamento dos valores de x atrasados de l valores equalizados. Os parâmetros do equalizador são obtidos com base no algoritmo LMS.

A grande ideia do Algoritmo 2 é que os parâmetros w são treinados ate o indice $M - l$, onde M é o tamanho do vetor atrasado, e depois de treinado, o erro a partir desse indice é forçado como 0, e então o LMS passa a não aprender mais, e sim aplicar os coeficientes de w para os l próximos valores, e então ocorre a predição propriamente dita.

A taxa de aprendizagem é definida como a amplitude do passo dado pelo gradiente, sabe-se que o algoritmo LMS é baseado em algoritmos de minimização através de descida de gradiente, e tais algoritmos estão associados ao gradiente pois este aponta para a tendência de aumento da função, então ao tomar o negativo do gradiente, tem-se um vetor que aponta para um mínimo local (ou global, na melhor hipótese).

A grande necessidade de um coeficiente de amplitude para o passo dado pelo gradiente, é que não se sabe se o tamanho do gradiente é tão grande que faria divergir a minimização, e isso é visto pela Equação 3, onde é possível estabelecer 4 características para a escolha do valor μ, mostrado na Figura 4 para uma visualização em duas dimensões.

No primeiro caso, o valor de μ é grande, pelo menos

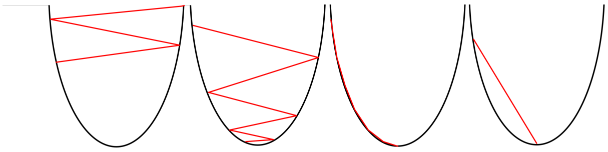

Fig. 4: Valores de μ na busca pela minimização da função custo em duas dimensões

maior que $\frac{1}{\lambda_{max}}$, e portanto o gradiente faz a estratégia de minimização divergir (não se aproxima dos mínimos).

No segundo caso, o valor de μ é grande, mas é suficientemente menor que $\frac{1}{\lambda_{max}}$, como é grande, o gradiente impulsiona um passo largo e ultrapassou o ponto de mínimo, observa-se na imagem portanto um formato de *zig-zag* até chegar no ponto de mínimo, esse *zig-zag* acarreta em um aumento do tempo de convergência também, o que contradiz com as Equações 4 e 5 pois sua demonstração assume μ pequeno.

No terceiro caso, o valor de μ é pequeno em relação a $\frac{1}{\lambda_{max}}$, por ser pequeno, o gradiente impulsiona passos curtos, garantindo que os passos não ultrapassam o ponto de mínimo. Porém, quanto menor o passo dado (a taxa de aprendizagem), mais lento será a convergência do algoritmo.

No quarto e último caso, escolheu-se um valor de μ nem tão grande quanto o valor do segundo caso, mas também nem tão pequeno quanto o terceiro caso, tal escolha ótima garantiu uma convergência em apenas uma iteração, chama-se tal valor como a taxa de aprendizagem ótima.

III. Validação

Para projetar o previsor e efetivamente analisar a eficiência do arranjo, tomou-se os primeiros 100 valores da ABEV3 (ON) e utilizou-se $l = 5$, ou seja, uma predição de 5 dias. Traça-se os gráficos para diferentes ordens do filtro, mostrado nas Figuras 5, 6 e 7

Para mensurar o erro da predição, toma-se os l últimos valores preditos e compara-se com os valores esperados das ações. O erro é normalizado com base na média dos l valores esperados, sendo assim, obtem-se os erros percentuais pontualmente (de cada dia) e o erro global é a media dos erros pontuais, é importante salientar que a análise dos erros também depende da ordem, e portanto toma-se os 3 casos estudados separadamente, conforme visto na Tabela I:

	1	2	3	4	5	Global
20	4.26	3.43	2.95	3.43	5.01	3.82
40	3.47	2.58	2.04	2.46	3.98	2.91
60	2.58	1.69	1.15	1.56	3.08	2.01

Tabela I: Erros na predição

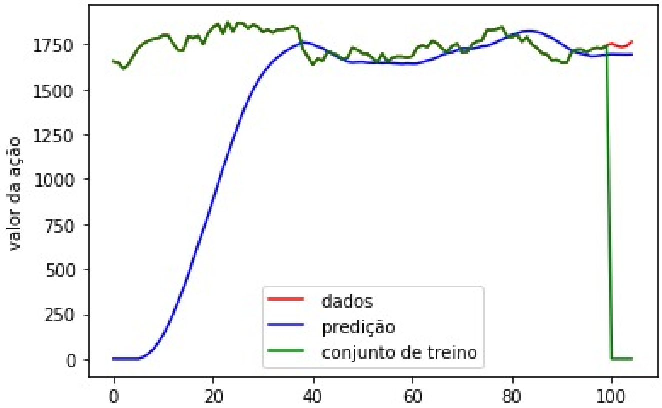

Fig. 5: Predição do arranjo utilizando ordem 20

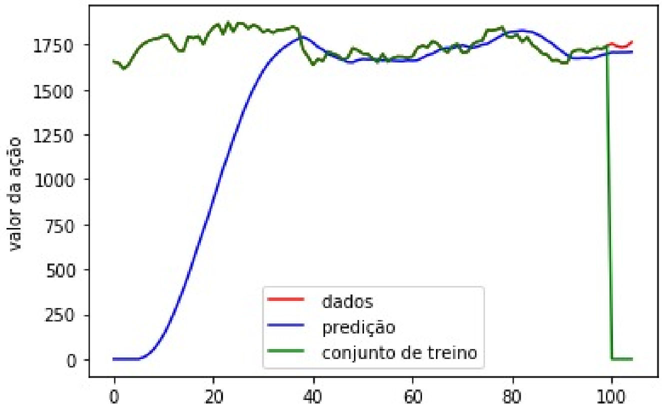

Fig. 6: Predição do arranjo utilizando ordem 40

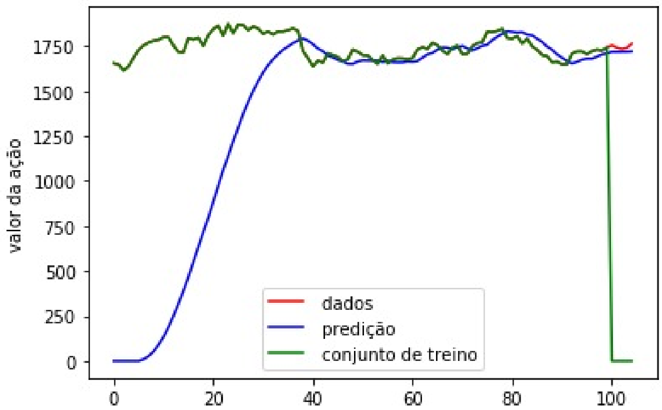

Fig. 7: Predição do arranjo utilizando ordem 60

No cenário prático de aplicação para esse esquema de preditor não é possível quantificar exatamente quanto lucro irá se obter, porém a utilização torna-se significante para estimar os melhores momentos para a compra e venda das ações a partir das tendências passadas. Na Figura 8 observa-se uma previsão de 10 dias ($l = 10$), contando 120 dias de aprendizado e um filtro de ordem 20. O preditor (curva verde) aponta que o dia de menor valor da ação corresponde ao dia 121, portanto, o melhor dia para comprar a ação, uma vez que se observa uma tendência de crescimento do seu valor. Enquanto o dia 127 corresponde ao melhor dia para vender, uma vez que há um pico no valor da ação, seguido de uma tendência de decrescimento. O gráfico real (curva vermelha), valida a tendência de crescimento apontada pelo preditor, sendo assim, ao realizar a compra e venda nos momentos indicados, se obteria um lucro de 5,78%.

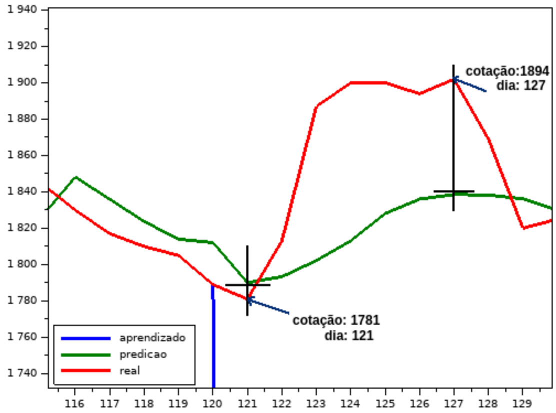

Fig. 8: Estimativa de lucro para um cenário de aplicação

IV. Conclusão

É possível observar que com o arranjo e utilizando no equalizador um filtro adaptativo, e portanto, utilizando de técnicas de predição puramente lineares, consegue-se predições bastante razoáveis para os valores das ações, tendo um erro máximo visto na Tabela I de 5.01%.

Em uma aplicação real sabe-se que fatores externos influenciam bastante na variação de precificação das ações de uma empresa, no exemplo utilizado da ABEV3 (ON): mudança de gerência da empresa e possível tomada de decisões erradas, acontecimentos naturais como biológicos (pragas e contaminações) ou climáticos (falta de chuva, terremotos e furacões). É importante reforçar que tais impactos podem ser classificados como positivos, caso ocorram em concorrentes, por exemplo, além de possíveis fatores positivos para a empresa repentinos, mas também podem ser impactos negativos, e tais impactos, por serem repentinos, criam variações muito bruscas na precificação das ações, e isso cria dificuldades para um preditor atuar.

Diante disso, não é interessante para a aplicação de preditor de ações uma predição de um valor futuro distante, e portanto a temporização usada no presente projeto pode ser considerada razoável para um preditor real.

Bibliografia

[1] PAULO S. R. DINIZ *Adaptive Filtering: Algorithms and Practical Implementation* . Springer.
[2] http://www.bmfbovespa.com.br/pt_br/servicos/market-data/historico/mercado-a-vista/series-historicas/