Aula 4

Séries Temporais -
Parte III - Modelos Preditivos

Introdução

- Os métodos de AM para predição, em oposição aos modelos estatísticos, buscam descrever as propriedades dos dados sem o conhecimento prévio da distribuição dos mesmos

- Por não dependerem explicitamente de parâmetros para modelar o comportamento do fenômeno, esses métodos são mais simples de serem ajustados e demonstram considerável desempenho mesmo quando aplicados à séries complexas e altamente não lineares

Introdução

- De acordo com [Islam and Sivakumar, 2002], pelo modo com as observações da série temporal são aproveitadas no modelo preditivo, os métodos de AM podem ser divididos em duas abordagens:
  - Aproximação Global
  - Aproximação Local

Aproximação Global

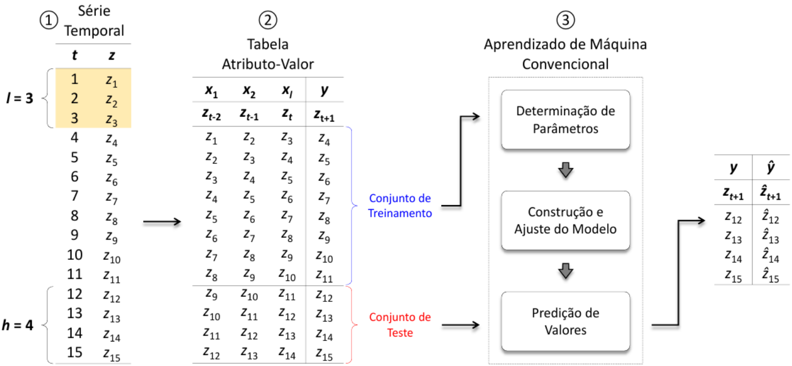
- A aproximação global é uma abordagem na qual os métodos de AM constroem modelos a partir de um procedimento de treinamento que recebe como entrada todas as observações de uma série

- O uso dessa abordagem envolve a transposição da sequência de dados para uma tabela atributo valor, de maneira que seja possível fornecê-la como entrada para os algoritmos convencionais de AM destinados à tarefa de regressão

Aproximação Global

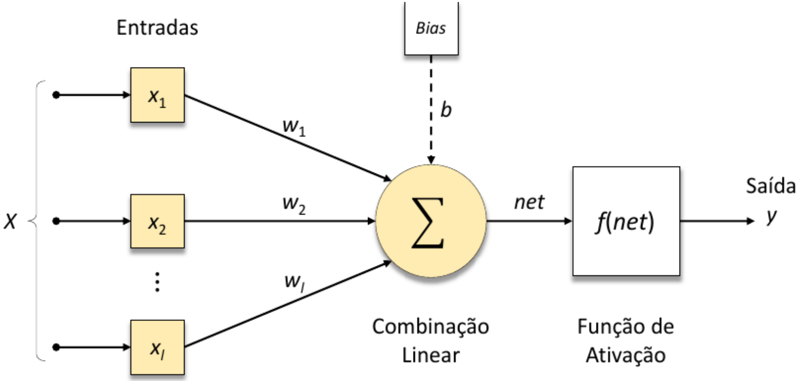
- Para tal, utiliza-se uma janela deslizante de comprimento l
- Essa janela é iterativamente deslocada sobre a série temporal com o propósito de coletar todas as subsequências originadas pelas l observações consecutivas
- Cada subsequência extraída remete a um para (X_i, y_i)
  - $X_i = (x_{i_1}, x_{i_2}, \ldots, x_{i_l})$ e corresponde ao padrão temporal de comprimento l
  - y_i indica o valor subsequente à X_i $\rightarrow$ observado no instance $l + 1$

Aproximação Global



Redes Neurais Artificiais

- O Perceptron constitui a forma mais simples de um ANN usada para a classificação de padrões linearmente separáveis
- O Perceptron corresponde à um único neurónio
- Cada entrada da rede é ponderada por um peso sináptico daquela entrada
- Existe uma entrada padrão denominada de bias


Entradas
Bias
x_1
x_2
x_l
X
w_1
w_2
w_l
b
$\sum$
net
$f(net)$
Saída
y
Combinação
Linear
Função de
Ativação

Redes Neurais Artificiais

- O cálculo da combinação linear de um perceptron é dada por

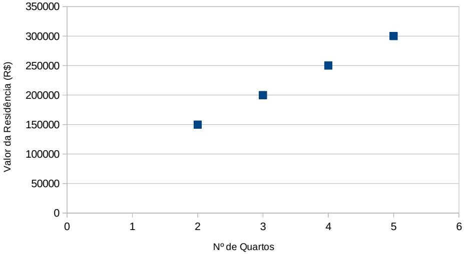
$$net = \sum_{i=1}^{I} w_i x_i + b$$

- A aprendizado do perceptron consistem em aprender os pesos w_i que diminuam a diferença entre o valor real e o valor predito pela rede $\rightarrow$ algoritmo *Least Mean Squares*
- No caso de uma tarefa de regressão, o perceptron irá aprender uma reta que melhor se aproximem dos pontos informados
- **OBSERVAÇÃO:** no caso do perceptron utilizado aqui para tarefa de regressão, a função de ativação será desconsiderada

Algoritmo LMS

- Para entender o algoritmo LMS, vamos começar com um caso simples: um único atributo de entrada e um único atributo de saída
- Neste caso, o aprendizado do perceptron consistirá apenas em aprender o w_1 e o *bias* (b)
- Vamos também considerar a entrada simples apresentada abaixo

Nº de Quartos	**Valor da Residência (R$)**
2	150.000
3	200.000
4	250.000
5	300.000



Algoritmo LMS

- O funcionamento da abordagem *LMS* é bem simples e intuitiva
- Para exemplificar, considere $w_0 = 1$ e $b = 1$ e vamos considerar o exemplo na primeira entrada da tabela anterior x_0
- Com esses parâmetros e a respectiva entrada, a saída do perceptron é igual a 3

$$w_0 * x_0 + b = 2 * 1 + 1 = 5$$

- Perceba que a saída desejada para o primeiro exemplo é 5 $\rightarrow$ o valor de w_1, ou b ou ambos teria que ser aumentados para produzir a reposta correta
- **OBSERVAÇÃO:** o mesmo vale para as demais exemplos

Algoritmo LMS

- O funcionamento da abordagem *LMS* é bem simples e intuitiva
- Para exemplificar, considere $w_0 = 3$ e $b = 1$ e vamos considerar o exemplo na primeira entrada da tabela anterior x_0
- Com esses parâmetros e a respectiva entrada, a saída do perceptron é igual a 3

$$w_0 * x_0 + b = 2 * 3 + 1 = 7$$

- Perceba que a saída desejada para o primeiro exemplo é 7 $\rightarrow$ o valor de w_0, ou b ou ambos teria que ser diminuídos para produzir a reposta correta
- **OBSERVAÇÃO:** o mesmo vale para as demais exemplos

Algoritmo LMS

- Seguindo as observações anteriores, se há um erro "para cima", deve-se diminuir os pesos e se há um erro "para baixo", deve-se aumentar os pesos.

- O quanto deve-se aumentar/diminuir?
    - A alteração é proporcional ao:
        - Erro $\rightarrow$ quanto maior o erro, maior deve ser a alteração
        - Os valores de entrada $\rightarrow$ quanto maior o valor da entrada, maior deve ser a alteração
        - Uma taxa de correção de erro (η) $\rightarrow$ controla o quanto as duas informações serão utilizadas para atualizar o erro (questões de convergência)

Algoritmo LMS

- Dado isso, a atualização dos pesos do perceptron para o caso anterior (e simples) é dado por:

$$w_0^{t+1} = w_0^t + \eta(y_{x_n} - f(x_n)) * x_n$$

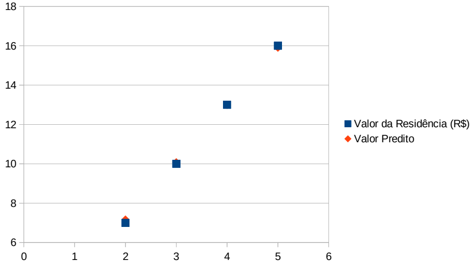
$$b^{t+1} = b^t + \eta(y_{x_n} - f(x_n)) * 1$$

na qual x_n é o n-ésimo exemplo de treinamento, $f(x_n)$ é a saída predita pelo *perceptron* para o exemplo x_n, e y_{x_n} é a saída real para o exemplo x_n

Algoritmo LMS

Nº de Quartos	Valor da Residência (R$)	Valor Predito
2	7	7,17
3	10	10,08
4	13	12,99
5	16	15,90

$w_1 = 2.91$
$b = 1.35$



Algoritmo LMS

- Vale ressaltar que pra valores numéricos maiores que 0 e para valores de η não muito pequenos, pode haver problemas na convergência no perceptron
- Por exemplo, considere um valor de $\eta = 1$
- Após a primeira época, temos $w_0 = 83307.0eb = 49057.0$
- ISSO NEM DE LONGE É A SOLUÇÃO PARA O PROBLEMA!!!

Algoritmo LMS

- Valores com escalas muito grandes também podem afetar a convergência do *perceptron*
- Por exemplo, considere $\eta = 0.01$ e número máximo de épocas $= 100$

Nº de Quartos	Valor da Residência (R$)
2000	7000000
3000	10000000
4000	13000000
5000	16000000

$W_0 = NaN$
$b = NaN$

Algoritmo LMS

- Uma forma de evitar esses problemas é padronizar os dados de forma que estes fiquem em uma escala menor e evitar os problemas mencionados anteriormente
- Uma forma comum de padronização é dividir cada valor de atributo pelo valor máximo daquele atributo

Valores Originais

Nº de Quartos	Valor da Residência (R$)
2	7
3	10
4	13
5	16

Padronização pelo valor máximo

Valores Padronizados

Nº de Quartos	Valor da Residência (R$)
0,40	0,44
0,60	0,63
0,80	0,81
1,00	1,00

$W_0 = 0.93$
$b = 0.06$

Algoritmo LMS

- **Regra geral** de atualização dos pesos no algoritmo *LMS*:

$$w_i^{t+1} = w_i^t + \eta(y_{x_i} - f(x_i)) * x_i$$

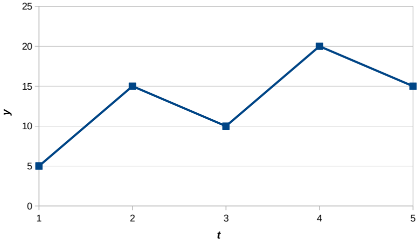
- **Critérios de parada**:
  - Número máximo de épocas (uma época é igual a uma passada completa por todos os exemplos de treinamento)
  - Erro quadrático médio mínimo

OBSERVAÇÃO: os pesos inicias podem ser definidos aleatoriamente ou todos iguais a 0

Exercício

- Considerando a abordagem de aprendizado global com tamanho de janela $= 3$, e o algoritmo perceptron com $\eta = 0.2$, pesos sinápticos iniciais $= 0$, $b = 2$ e numero máximo de épocas $= 2$, faça a predição para $t = 6$ considerando a série temporal apresentada abaixo

t	*y*
1	5
2	15
3	10
4	20
5	15

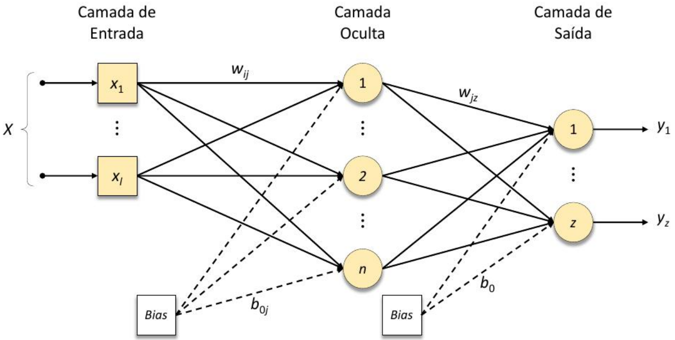


OBSERVAÇÃO: obviamente, o valor previsto estará longe do ideal uma vez o algoritmo LMS é interrompido precocemente

Limitações do Perceptron

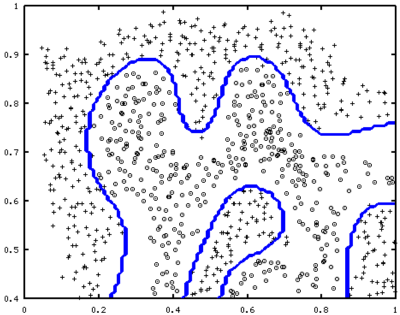
- O fato de utilizar uma única reta ou considerar a linearidade entre os dados limita a capacidade do Perceptron
- Para isso, foram desenvolvidas redes neurais com múltipas camadas → *Multi-Layer Perceptrons* (*MLP*)
- Normalmente utiliza-se o algoritmo *Backpropagation* para atualizar os pesos da rede

Limitações do Perceptron



Limitações do Perceptron

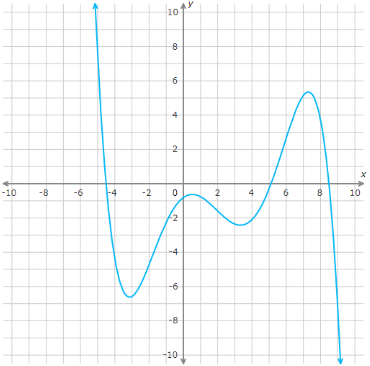
- Ao combinar funções de aproximação lineares, pode-se obter funções de aproximação não lineares



https://qph.ec.quoracdn.net/main-qimg-614ac8f108c43cc9c158b83d5d9e4843

Limitações do Perceptron

- Ao combinar funções de aproximação lineares, pode-se obter funções de aproximação não lineares



https://www.onlinemath4all.com/images/nonlinearfunctiongraph1.png

Aproximação Global

- A aproximação global não está isenta de limitações
- Consideram que os pares (X_{i_j}, y_{i_j}) são considerados independentes e identicamente distribuídos pelos algoritmos de AM tradicionais
- Essa suposição acarreta em perda de informação temporal, o que implica na degradação do desempenho preditivo

Aproximação Local

- Na abordagem local, os métodos particionam a série temporal original em subsequências cujos valores mais próximos ou mais importantes em relação ao valor atual são combinados para produzir o valor futuro
- Essas combinações são empreendidas por funções de aproximação, como a média local simples, relativa, ...
- Dentre os métodos aplicados conforme essa abordagem encontram-se as variações do algoritmo *k-Nearest Neighbors*

Material Complementar

- Descrição de Modelos Estatísticos e de Aprendizado de Máquina para Predição de Séries Temporais

  http://conteudo.icmc.usp.br/CMS/Arquivos/arquivos_enviados/BIBLIOTECA_158_RT_412.pdf

- Support vector machine

  https://en.wikipedia.org/wiki/Support_vector_machine

- Perceptron

  https://en.wikipedia.org/wiki/Perceptron

Imagem do Dia

Sistemas de Apoio à Decisão
http://lives.ufms.br/moodle/

Rafael Geraldeli Rossi
rafael.g.rossi@ufms.br

Slides baseados na dissertação de mestrado de Antonio Rafael Sabino Parmenzan (Predição de Séries Temporais por Similaridade)

Referências Bibliográficas I

Islam, M. and Sivakumar, B. (2002).
Characterization and prediction of runoff dynamics: a nonlinear dynamical view.
Advances in water resources, 25(2):179–190.