Regressão Linear

Gustavo Alves Pacheco*

11821ECP011

1 Introdução

No processo de regressão linear, busca-se a modelagem da relação linear entre variáveis, mais especificamente a relação entre a variável dependente e as independentes, também chamadas de explanatórias [3].

Em regressões lineares simples, utiliza-se apenas uma variável explanatória (eq. 1), enquanto nas múltiplas, diversas variáveis são somadas para formar o resultado (eq. 2).

$$y = ax + b \tag{1}$$

$$y = a_1 * x_1 + a_2 * x_2 + a_3 * x_3 + ... + b \tag{2}$$

Nestas equações, y é a variável dependente, enquanto x_i são as variáveis independentes. a_i representam os coeficientes angulares e b, o intercepto.

Logo, para determinar o modelo que descreve tais relações, é necessário descobrir os valores dos coeficientes a_i e do intercepto b. Para tal, algumas técnicas podem ser aplicadas. Duas delas serão citadas neste trabalho.

A primeira envolve a utilização do coeficiente de Pearson, r, definido pela equação 3, abaixo. Este coeficiente determina o grau de correlação entre duas variáveis. Aliado a este, o valor de r^2 também é interessante, sendo conhecido como coeficiente de determinação e medindo o percentual da variação de y que é explicado pela variação de x.

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{3}$$

No geral, esta correlação entre duas variáveis pode ser positiva (quando o crescimento da variável independente é acompanhado pelo da dependente), negativa (quando o crescimento da independente ocasiona um decrescimento na dependente), não linear (a relação entre ambas não é descrita por uma equação de reta) ou sem correlação.

*gap1512@gmail.com

Pelo coeficiente de Pearson, um valor de $r < 0$ representa uma correlação negativa, enquanto $r > 0$, positiva. $r = 0$ indica que não há correlação linear entre as variáveis. Neste caso simples, é possível determinar o modelo, através das equações 4 [1] e 5 [2], abaixo:

$$a = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2} \tag{4}$$

$$b = \bar{y} - a\bar{x} \tag{5}$$

A segunda técnica é uma modelagem através de um Adaline. Até o momento, alguns pesos eram descobertos, os quais descreviam uma saída por meio da relação entre as entradas multiplicadas por pesos. E é exatamente esta a definição de regressão linear. Portanto, a aplicação do Adaline para tal propósito é simples e direta.

2 Objetivos

- Aprimorar o conhecimento sobre Redes Neurais Artificiais e obter experiência prática na implementação das mesmas.
- Implementar um Adaline que realize a regressão linear para os dados da tabela 1.
- Encontrar o coeficiente de correlação de Pearson e o coeficiente de determinação para a mesma base de dados.
- Calcular a e b (eq. 4 e 5).
- Comparar resultados obtidos com ambas as técnicas.

Table 1: Base de Dados

x	y
0.00	2.26
0.50	3.80
1.00	4.43
1.50	5.91
2.00	6.18
2.50	7.26
3.00	8.15
3.50	9.14
4.00	10.87
4.50	11.58
5.00	12.55

3 Materiais e Métodos

Para implementação da rede neural foi utilizada a linguagem de programação Common Lisp, compilando-a com o SBCL (Steel Bank Common Lisp). Como interface de desenvolvimento, foi utilizado o Emacs em Org Mode, configurado com a plataforma SLIME (The Superior Lisp Interaction Mode for Emacs) para melhor comunicação com o SBCL. Foi utilizada uma abordagem bottom-up para o desenvolvimento. O código produzido segue majoritariamente o paradigma funcional, sendo este trabalho como um todo uma obra de programação literária. Parte das funções já foram implementadas em Regra de Hebb e Perceptron e Adaline.

4 Adaline

Inicialmente, será implementada uma função que gera a equação da reta a partir dos coeficientes. Desta forma, a função definida abaixo faz o desejado, entretanto se utiliza da função `eval` para a transformação da *s-expression* em código interpretado, técnica que em vários casos deve ser evitada [1].

```lisp
(defun linear-regression (weights)
  (let* ((wi (butlast weights))
         (b (first (last weights)))
         (args (loop for nil in wi collecting (gensym))))
    (eval `#'(lambda ,args
               (+ ,@(mapcar #'(lambda (a w)
                                `(* ,a ,w))
                            args wi)
                  ,b)))))
```

Continuando a implementação utilizando Adaline, tem-se que os pesos (w e b), encontrados durante o treinamento, após 1000 ciclos, e com `learning-rate` de 0.005 são:

```lisp
(iterative-training
 tb01-x tb01-y
 (random-weights 3 -1 1) 0 0.05 0 1000
 #'adaline-update #'adaline-stop-condition #'net #'adaline-activation)
```

2.0179942	2.4513268

[1] `https://stackoverflow.com/questions/2571401/why-exactly-is-eval-evil/2571549`

Assim, a equação da reta (eq. 6) seria:

$$y = 2.0179942x + 2.4513268 \tag{6}$$

Para plotagem dos pontos, a função `linear-boundary` deve ser adaptada, da seguinte forma:

```
(defun linear-between (fn min max)
  (list (list min (funcall fn min))
        (list max (funcall fn max))))
```

Portanto, o gráfico 1:

```
(scatter-plot "plots/scatter-plot-adaline.png"
              tb01
              (linear-between (linear-regression w-adaline) 0 5))
```

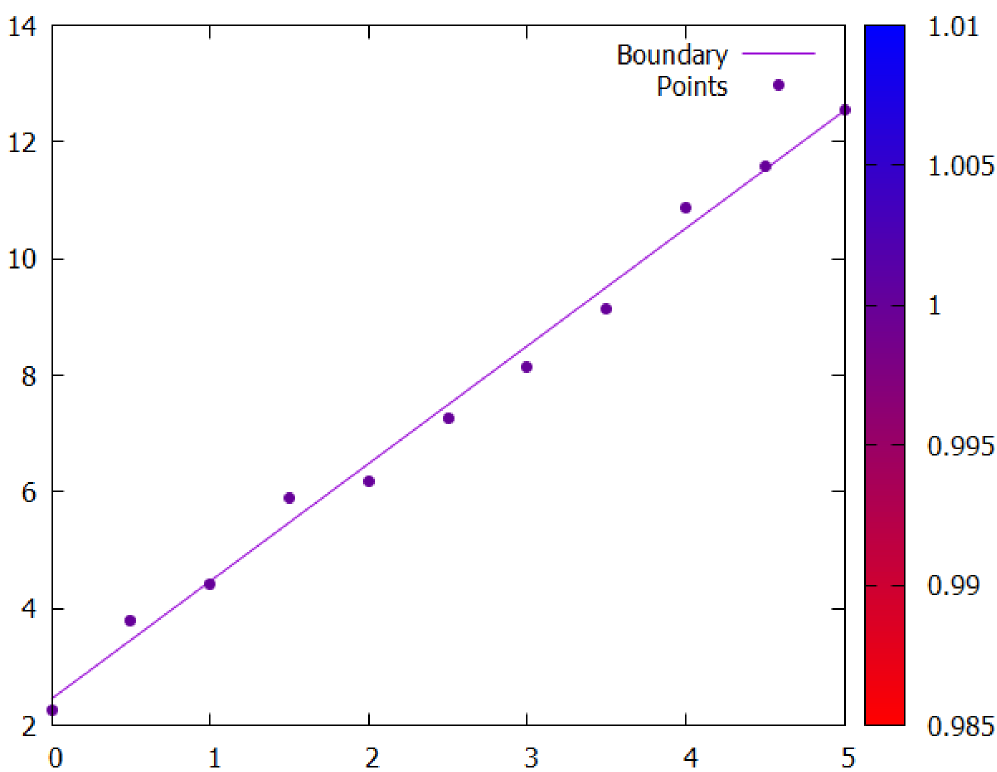

Figure 1: Regressão Linear Por Adaline

5 Pearson

Primeiramente, a função para cálculo do r e do r quadrático:

```
(defun average (points)
  (loop for (x y) in points
     summing x into c-x
     summing y into c-y
     counting t into i
     finally (return (values (/ c-x i) (/ c-y i)))))

(defun r-pearson (points)
  (multiple-value-bind (av-x av-y)
      (average points)
    (loop for (x y) in points
       summing (* (- x av-x) (- y av-y)) into cov-xy
       summing (expt (- x av-x) 2) into var-x
       summing (expt (- y av-y) 2) into var-y
       finally (return (/ cov-xy (sqrt (* var-x var-y)))))))

(defun r-sqrd (points)
  (expt (r-pearson points) 2))
```

Utilizando-as para a tabela 1, tem-se que o coeficiente r é:

```
(r-pearson tb01)
```

```
0.99611324
```

Enquanto r^2 é:

```
(r-sqrd tb01)
```

```
0.99224156
```

O cálculo de a e b, seguindo as equações 4 e 5 é definido da seguinte forma:

```
(defun simple-linear-regression (points)
  (multiple-value-bind (av-x av-y)
  (average points)
```

```
(loop for (x y) in points
   summing (* (- x av-x) (- y av-y)) into s
   summing (expt (- x av-x) 2) into s-sqrd
   finally (let ((a (/ s s-sqrd)))
              (return (list a (- av-y (* a av-x))))))))
```

Para a base de dados, temos que a e b são:

```
(simple-linear-regression tb01)
```

2.0058184 2.4518175

Portanto, a equação da reta (eq. 7):

$$y = 2.0058184x + 2.4518175 \tag{7}$$

De forma semelhante, a impressão se dá da seguinte maneira (fig. 2):

```
(scatter-plot "plots/scatter-plot-pearson.png"
              tb01
              (linear-between (linear-regression w-simple) 0 5))
```

6 Conclusão

Ambos os algoritmos encontraram valores muito semelhantes de a e b. Percebe-se que o valor do intercepto foi mais próximo nas duas estratégias se comparado ao valor da inclinação. Apesar deste fato, observa-se nos gráficos que ambas as retas descrevem com bastante fidelidade o conjunto de pontos.

Em relação ao valor do coeficiente, $r = 0.99611324$, verifica-se que a relação entre x e y é positiva ($r > 0$) e bastante forte, com $r^2 = 0.99224156$

References

[1] Boston University School of Public Health. Correlation and regression with r. https://sphweb.bumc.bu.edu/otlt/MPH-Modules/BS/R/R5_Correlation-Regression/R5_Correlation-Regression3.html.

[2] PennState. What is the "best fitting line"? https://online.stat.psu.edu/stat501/lesson/1/1.2. Eberly College of Science.

[3] K. Yamanaka. Aprendizagem de máquina (machine learning - ml). Universidade Federal de Uberlândia.

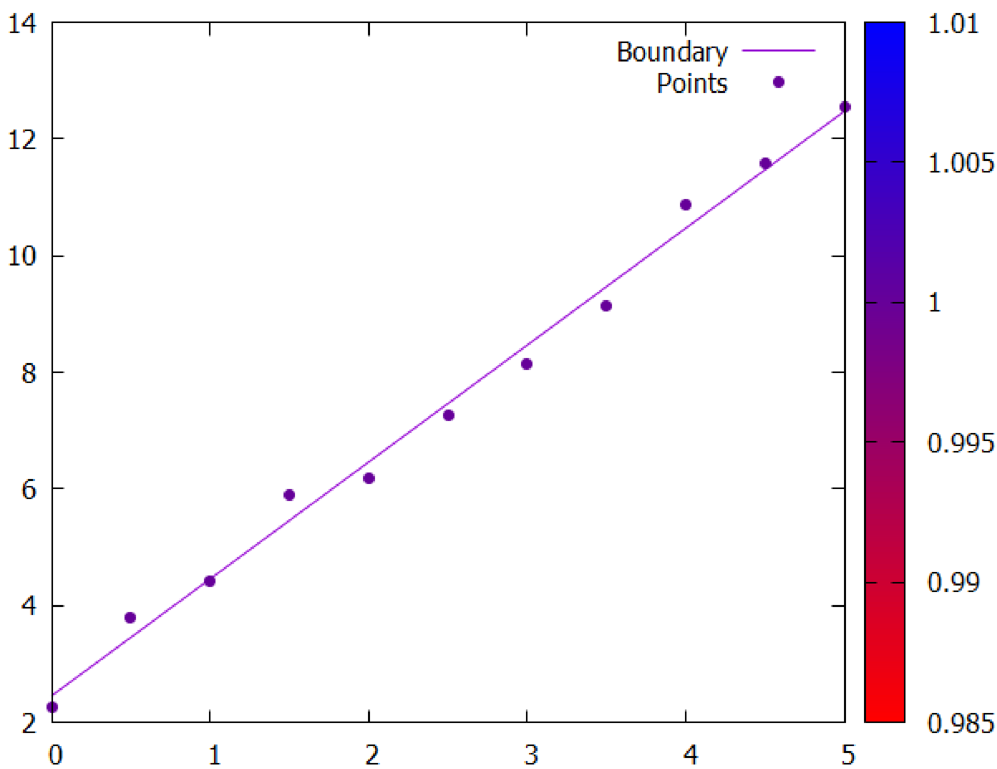

Figure 2: Regressão Linear Simples