

# Regressão Linear

## Prática 04: Regressão Lasso

Prof<sup>a</sup> Deborah Magalhães

Monitor: Davi Luis de Oliveira



UNIVERSIDADE  
FEDERAL DO PIAUÍ



## **Curso: Bacharelado em Sistema de Informação**

Disciplina: Sistemas Inteligentes

### ▷ **Predição com Regressão Lasso**

Você pode me encontrar em [deborah.vm@gmail.com](mailto:deborah.vm@gmail.com)

(Dúvidas e sugestões serão bem-vindas =D)

# Passo 1: Importar as bibliotecas

## Prática 03: Regressão Lasso

### 1. Importando as bibliotecas necessárias

```
In [1]: import graphlab
import random
import math
import numpy
from matplotlib import pyplot as plt
%matplotlib inline
```

This non-commercial license of GraphLab Create for academic use is assigned to deborah.v@gmail.com and will expire on January 26, 2019.

```
[INF0] graphlab.cython.cy_server: GraphLab Create v2.1 started. Logging:
/tmp/graphlab_server_1525336285.log
```

## Passo 2: Gerando os dados sinteticamente

### 2. Gerando artificialmente os dados

```
In [5]: random.seed(98103)
        n = 30
        x = graphlab.SArray([random.random() for i in range(n)]).sort()
```

```
In [6]: y = x.apply(lambda x: math.sin(4*x))
```

## Passo 3: Adicionando ruído aos dados

### 3. Adicionando aos dados um ruído Gaussiano

```
In [7]: random.seed(1)
e = graphlab.SArray([random.gauss(0,1.0/3.0) for i in range(n)])
y = y + e
```

## Passo 4: Criar um SFrame

### 4. Dispor os dados gerados em formato de SFrame

```
In [9]: data = graphlab.SFrame({'X1':x, 'Y':y})
```

# Passo 5: Definir as características do modelo de regressão

## 5. Definir uma função para criar as características do modelo de regressão polinomial de qualquer grau

```
In [11]: def polynomial_features(data, deg):  
    data_copy=data.copy()  
    for i in range(1,deg):  
        data_copy['X'+str(i+1)]=data_copy['X'+str(i)]*data_copy['X1']  
    return data_copy
```

## Passo 6: Definir uma função para ajustar o modelo de regressão Lasso aos dados

```
def polynomial_lasso_regression(data, deg, l1_penalty):  
    model =  
    graphlab.linear_regression.create(polynomial_features(data,deg),  
                                     target='Y', l2_penalty=0.,  
                                     l1_penalty=l1_penalty,  
                                     validation_set=None,  
                                     solver='fista', verbose=False,  
                                     max_iterations=3000,  
                                     convergence_threshold=1e-10)  
    return model
```



# Passo 7: Imprimir os coeficientes do modelo

## 7. Definir a função que imprime os coeficientes do modelo

```
In [14]: def print_coefficients(model):  
  
    deg = len(model.coefficients['value'])-1  
    w = list(model.coefficients['value'])  
  
    print 'Coeficientes do polinômio de grau ' + str(deg) + ':'  
    w.reverse()  
    print numpy.poly1d(w)
```

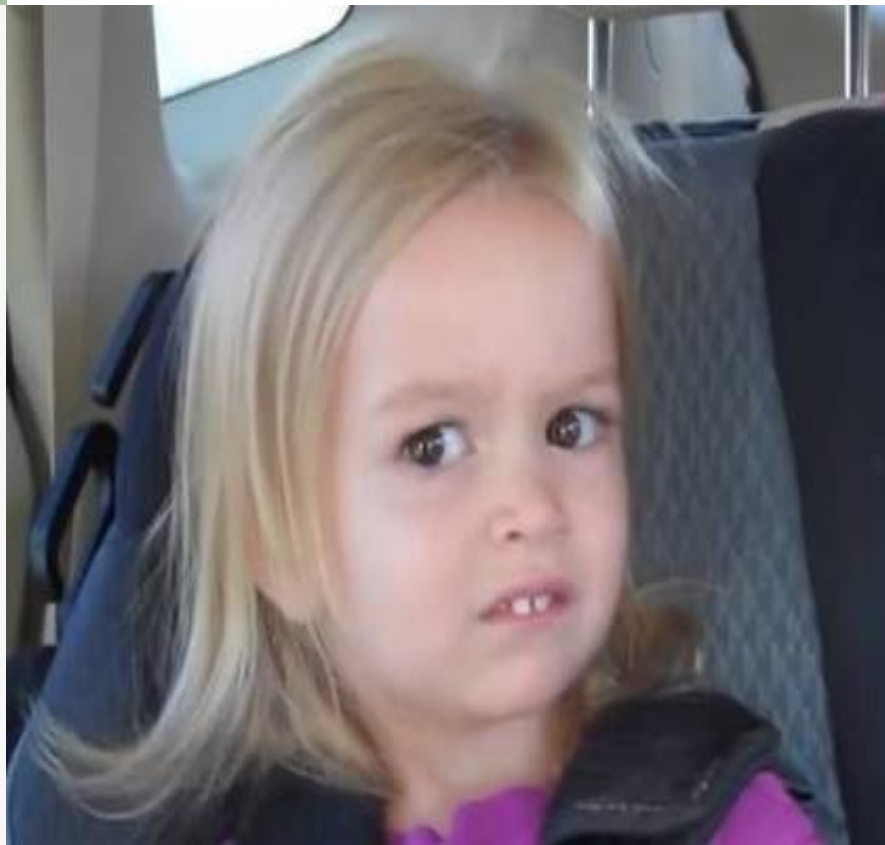
# Passo 8: Plotar dados e previsões

```
def plot_data(data):  
    plt.plot(data['X1'], data['Y'], 'k.')  
    plt.xlabel('x')  
    plt.ylabel('y')  
  
def plot_poly_predictions(data, model):  
    plot_data(data)  
    deg = len(model.coefficients['value'])-1  
  
    x_pred = graphlab.SFrame({'X1': [i/200.0 for i in range(200)]})  
    y_pred = model.predict(polynomial_features(x_pred, deg))  
  
    plt.plot(x_pred['X1'], y_pred, 'g-', label='degree ' + str(deg) + ' fit')  
    plt.legend(loc='upper left')  
    plt.axis([0, 1, -1.5, 2])
```

## Passo 9: Observar os modelos para uma sequência crescente de valores de lambda

```
for l1_penalty in [0.0001, 0.01, 0.1, 10]:  
    model = polynomial_lasso_regression(data, deg=16, l1_penalty=l1_penalty)  
    print 'l1_penalty = %e' % l1_penalty  
    print 'number of nonzeros = %d' % (model.coefficients['value']).nnz()  
    print_coefficients(model)  
    print '\n'  
    plt.figure()  
    plot_poly_predictions(data,model)  
    plt.title('LASSO, lambda = %.2e, # nonzeros = %d' % (l1_penalty,  
(model.coefficients['value']).nnz()))
```

Ao fim do Passo 9,  
observamos que a medida  
que  $\lambda$  aumenta, os dados vão  
ficando mais esparsos, ou  
seja com mais zeros



**Dúvidas? Sugestões?  
Inquietações?  
Aconselhamentos?**

- ▶ Desabafe em:  
**[deborah.vm@gmail.com](mailto:deborah.vm@gmail.com)**