

# Fouille de Données

# Data Mining

## Régression Linéaire - Exemple

# Plan du cours

1. Régression / Estimation : Définition et principe
2. Régression linéaire
3. Régression linéaire avec OLS
4. Entrainement et évaluation – Loss functions
5. **Régression linéaire avec le Gradient Descent**

# La régression linéaire

## Régression linéaire – Utilisation d'un algorithme d'optimisation

Ex : **Algorithme de Gradient Descent** (Descente du Gradient)

**Etapes :**

- Modèle linéaire avec  $f(x)$  :  $\textcolor{red}{y = bj * xj + bo}$
  - Fonction coût : **MSE** – Mean Squared Error = 
$$\frac{1}{m} \sum_{i=1}^m (\hat{y}^{(i)} - y^{(i)})^2$$
  - 1** - Initialiser les paramètres  $\textcolor{red}{bj}$  et  $\textcolor{red}{bo}$  à 0
  - 2** - Choisir et fixer le nombre d'itération (**epochs**) et **learning\_rate  $\alpha$** .
  - 3** – Calculer les prédictions  $\textcolor{violet}{y\_pred}$  pour chaque exemple **i** dans le dataset, selon :  $\textcolor{violet}{y\_pred} = \textcolor{red}{bj} * \textcolor{red}{xj} + \textcolor{red}{bo}$
  - 4** – Calculer le gradient : i.e. les dérivées partielles  $\textcolor{red}{D\_bj}$  et  $\textcolor{red}{D\_bo}$ .
  - 5** - **Mettre à jour** les valeurs de  $\textcolor{red}{bj}$  et de  $\textcolor{red}{bo}$  en fonction du gradient.
- Répéter** (3 4 5) *epochs* fois pour minimiser gradient et **optimiser  $bj, bo$**

# La régression linéaire

## Régression linéaire – Utilisation d'un algorithme d'optimisation

Ex : **Algorithme de Gradient Descent** (Descente du Gradient)

**Etapes** :

$$y = b_0 + b_1 * x_1 + b_2 * x_2 + \dots + b_n * x_n$$

4 – Calculer le gradient : i.e. les dérivées partielles **D\_bj** et **D\_bo** :

- **D\_bj** : Dérivée partielle de la fonction coût MSE selon les paramètres bj
- **D\_bo** : Dérivée partielle de la fonction coût MSE selon le paramètre bo

$$\mathbf{D_bj} = \frac{\partial}{\partial b_j} MSE = \frac{2}{m} \sum_{i=1}^m (\hat{y}^{(i)} - y^{(i)}) x_j^{(i)}$$

# La régression linéaire

## Régression linéaire – Utilisation d'un algorithme d'optimisation

Ex : **Algorithme de Gradient Descent** (Descente du Gradient)

**Etapes :**

$$y = b_0 + b_1 * x_1 + b_2 * x_2 + \dots + b_n * x_n$$

4 – Calculer le gradient : i.e. les dérivées partielles  $D_{bj}$  et  $D_{bo}$  :

- $D_{bj}$  : Dérivée partielle de la fonction coût MSE selon les paramètres  $b_j$
- $D_{bo}$  : Dérivée partielle de la fonction coût MSE selon le paramètre  $b_0$

$$D_{bo} = \frac{\partial}{\partial b_0} MSE = \frac{2}{m} \sum_{i=1}^m (\hat{y}^{(i)} - y^{(i)})$$

# La régression linéaire

## Régression linéaire – Utilisation d'un algorithme d'optimisation

Ex : **Algorithme de Gradient Descent** (Descente du Gradient)

**Etapes :**

**5 – Mettre à jour** les valeurs de ***bj*** et de ***bo*** en fonction du gradient et du **learning rate α**, comme suit :

New parameter = old parameter – learning rate × gradient

For  $b_0$ :

$$b_0 := b_0 - \alpha \cdot \frac{2}{m} \sum (\hat{y} - y)$$

For each  $b_j$ :

$$b_j := b_j - \alpha \cdot \frac{2}{m} \sum (\hat{y} - y) x_j$$

# La régression linéaire

## Régression linéaire – Utilisation d'un algorithme d'optimisation

Ex : **Algorithme de Gradient Descent** (Descente du Gradient)

**Etapes :**

**5 – Mettre à jour** les valeurs de ***bj*** et de ***bo*** en fonction du gradient et du **learning rate  $\alpha$** , comme suit :

$$b_j := b_j - \alpha \cdot \frac{\partial}{\partial b_j} MSE$$

**Répétez** les étapes (3, 4, et 5) ***epochs*** fois; jusqu'à ce que la fonction de coût a une très petite valeur ou idéalement = 0 (ce qui signifie 0 erreur ou 100% de précision).

Les dernières valeurs de ***bj*** et de ***bo*** trouvées représentent leurs valeurs optimales.

# La régression linéaire

## Régression linéaire – Utilisation d'un algorithme d'optimisation

### Ex : Algorithme de Gradient Descent (Descente du Gradient)

```
Initialize b0, b1, ... , bn = 0
Choose learning rate α
repeat until convergence:
    for i in 1..m:
        y_pred[i] = b0 + Σ(bj * xj[i])
    Compute gradients:
        grad_b0 = (2/m) * Σ(y_pred[i] - y[i])
        grad_bj = (2/m) * Σ((y_pred[i] - y[i]) * xj[i]) for each j
    Update parameters:
        b0 = b0 - α * grad_b0
        bj = bj - α * grad_bj for each j
```

- 1 Start with random weights
- 2 Predict using the linear model
- 3 Compute error
- 4 Compute gradient
- 5 Update weights
- 6 Repeat until convergence

## Linear Regression with Gradient Descent

Exemple :

<b>x1</b>	<b>x2</b>	<b>x3</b>	<b>y</b>
1	2	3	3.10
2	1	0	5.80
3	0	2	7.55
4	1	1	6.40
5	2	0	10.00

We set:

- $m = 5$
- learning rate  $\alpha = 0.05$
- initial  $\beta^{(0)} = [0, 0, 0, 0]$

## Linear Regression with Gradient Descent

Exemple :

Iteration 1 (initial)

Step A – Predictions with  $\beta^{(0)}$ :

Since  $\beta^{(0)} = 0$ :

1 - Initialiser les paramètres  $b_j$  et  
 $b_0$  à 0

3 – Calculer les prédictions  $y_{pred}$  pour chaque exemple  $i$  dans le dataset, selon :  $y_{pred} = b_0 + b_j * x_j$

$$\hat{y}^{(0)} = X\beta^{(0)} = [0, 0, 0, 0, 0]^T$$

<b>x1</b>	<b>x2</b>	<b>x3</b>	<b>y_true</b>
1	2	3	3.10
2	1	0	5.80
3	0	2	7.55
4	1	1	6.40
5	2	0	10.00

<b>x1</b>	<b>x2</b>	<b>x3</b>	<b>y_true</b>	<b>y_pred</b>	<b>y_pred</b>
1	2	3	3.10	o	$b_0 + (b_1 * x_1) + (b_2 * x_2) + (b_3 * x_3) = 0 + 0 * 1 + 0 * 2 + 0 * 3 = 0$
2	1	0	5.80	o	$b_0 + (b_1 * x_1) + (b_2 * x_2) + (b_3 * x_3) = 0 + 0 * 2 + 0 * 1 + 0 * 0 = 0$
3	0	2	7.55	o	$b_0 + (b_1 * x_1) + (b_2 * x_2) + (b_3 * x_3) = 0 + 0 * 3 + 0 * 0 + 0 * 2 = 0$
4	1	1	6.40	o	$b_0 + (b_1 * x_1) + (b_2 * x_2) + (b_3 * x_3) = 0 + 0 * 4 + 0 * 1 + 0 * 1 = 0$
5	2	0	10.00	o	$b_0 + (b_1 * x_1) + (b_2 * x_2) + (b_3 * x_3) = 0 + 0 * 5 + 0 * 2 + 0 * 0 = 0$

## Linear Regression with Gradient Descent

Exemple :

Step B – Residuals  $r_i = \hat{y}_i - y_i$ :

$$r^{(0)} = [0 - 3.10, 0 - 5.80, 0 - 7.55, 0 - 6.40, 0 - 10.00]^T =$$

$$= [-3.10, -5.80, -7.55, -6.40, -10.00]^T.$$

$$\left( \hat{y}^{(i)} - y^{(i)} \right)$$

y_true	y_pred
3.10	o
5.80	o
7.55	o
6.40	o
10.00	o

## Linear Regression with Gradient Descent

### Exemple :

**4 – Calculer les gradients, grad\_o :  $\text{grad}_o = \frac{2}{m} \sum_{i=1}^m (\hat{y}^{(i)} - y^{(i)})$**

Step C — Compute gradients (recall factor  $2/m = 2/5 = 0.4$ ).

We compute the sums  $S_j = \sum_{i=1}^5 r_i X_{i,j}$  then  $\text{grad}_j = 0.4 \cdot S_j$ .

- $j = 0$  (intercept,  $X_{i,0} = 1$ ):

$$S_0 = \sum r_i = -3.10 - 5.80 - 7.55 - 6.40 - 10.00 = -32.85$$

$$\text{grad}_0 = 0.4 \times (-32.85) = -13.14$$

## Linear Regression with Gradient Descent

Exemple :

4 – Calculer les gradients :  $\frac{2}{m} \sum_{i=1}^m (\hat{y}^{(i)} - y^{(i)}) x_j^{(i)}$

## Linear Regression with Gradient Descent

Exemple :

4 – Calculer les gradients :  $D_{bj} = \frac{2}{m} \sum_{i=1}^m (\hat{y}^{(i)} - y^{(i)}) x_j^{(i)}$

j= 1 : D\_b1

$$[-3.10, -5.80, -7.55, -6.40, -10.00]'$$

x1
1
2
3
4
5

## Linear Regression with Gradient Descent

Exemple :

4 – Calculer les gradients :  $\frac{2}{m} \sum_{i=1}^m (\hat{y}^{(i)} - y^{(i)}) x_j^{(i)}$

j= 1 : D\_b1

- $j = 1$  (feature 1, column = [1,2,3,4,5]):

$$S_1 = (-3.10) \cdot 1 + (-5.80) \cdot 2 + (-7.55) \cdot 3 + (-6.40) \cdot 4 + (-10.00) \cdot 5$$

$$= -3.10 - 11.60 - 22.65 - 25.60 - 50.00 = -112.95$$

$$\text{grad}_1 = 0.4 \times (-112.95) = -45.18$$

## Linear Regression with Gradient Descent

### Exemple :

4 – Calculer les gradients :  $\mathbf{D}_{\mathbf{b}j} = \frac{2}{m} \sum_{i=1}^m (\hat{y}^{(i)} - y^{(i)}) x_j^{(i)}$

- $j = 2$  (feature 2, column = [2,0,1,3,2]):

$$\begin{aligned} S_2 &= (-3.10) \cdot 2 + (-5.80) \cdot 0 + (-7.55) \cdot 1 + (-6.40) \cdot 3 + (-10.00) \cdot 2 \\ &= -6.20 + 0 - 7.55 - 19.20 - 20.00 = -52.95 \end{aligned}$$

$$\text{grad}_2 = 0.4 \times (-52.95) = -21.18$$

- $j = 3$  (feature 3, column = [3,1,2,0,1]):

$$\begin{aligned} S_3 &= (-3.10) \cdot 3 + (-5.80) \cdot 1 + (-7.55) \cdot 2 + (-6.40) \cdot 0 + (-10.00) \cdot 1 \\ &= -9.30 - 5.80 - 15.10 + 0 - 10.00 = -40.20 \\ \text{grad}_3 &= 0.4 \times (-40.20) = -16.08 \end{aligned}$$

## Linear Regression with Gradient Descent

Exemple :

4 – Calculer les gradients :  $D_{bj}$  et  $D_{bo}$

$D_{bo}, D_{b1}, D_{b2}, D_{b3}$  :

So the gradient vector is

$$\nabla J(\beta^{(0)}) = [-13.14, -45.18, -21.18, -16.08]^T.$$

## Linear Regression with Gradient Descent

### Exemple :

5 – Mettre à jour les valeurs de  $\mathbf{bj}$  et de  $\mathbf{bo}$  :

New parameter = old parameter – learning rate × gradient

Step D — Parameter update:

$$\beta_j^{(1)} = \beta_j^{(0)} - \alpha \cdot \text{grad}_j.$$

Compute each:

- $\beta_0^{(1)} = 0 - 0.05 \cdot (-13.14) = 0 + 0.657 = 0.657$
- $\beta_1^{(1)} = 0 - 0.05 \cdot (-45.18) = 0 + 2.259 = 2.259$
- $\beta_2^{(1)} = 0 - 0.05 \cdot (-21.18) = 0 + 1.059 = 1.059$
- $\beta_3^{(1)} = 0 - 0.05 \cdot (-16.08) = 0 + 0.804 = 0.804$

So after Iteration 1:

$$\beta^{(1)} = [0.657, 2.259, 1.059, 0.804]^T.$$

## Linear Regression with Gradient Descent

### Exemple :

**Répétez** les étapes (3, 4, et 5) **epochs** fois; jusqu'à ce que la fonction de coût a une très petite valeur ou idéalement = 0 (ce qui signifie 0 erreur ou 100% de précision).

Les dernières valeurs de **bj** et de **bo** trouvées représentent leurs valeurs optimales.

## Iteration 2

Step A — Predictions  $\hat{y}^{(1)} = X\beta^{(1)}$ . Compute each row:

1. Row1:  $0.657 + 1 \cdot 2.259 + 2 \cdot 1.059 + 3 \cdot 0.804$   
 $= 0.657 + 2.259 + 2.118 + 2.412 = 7.446$
2. Row2:  $0.657 + 2 \cdot 2.259 + 0 \cdot 1.059 + 1 \cdot 0.804$   
 $= 0.657 + 4.518 + 0 + 0.804 = 5.979$
3. Row3:  $0.657 + 3 \cdot 2.259 + 1 \cdot 1.059 + 2 \cdot 0.804$   
 $= 0.657 + 6.777 + 1.059 + 1.608 = 10.101$
4. Row4:  $0.657 + 4 \cdot 2.259 + 3 \cdot 1.059 + 0 \cdot 0.804$   
 $= 0.657 + 9.036 + 3.177 + 0 = 12.870$
5. Row5:  $0.657 + 5 \cdot 2.259 + 2 \cdot 1.059 + 1 \cdot 0.804$   
 $= 0.657 + 11.295 + 2.118 + 0.804 = 14.874$

So  $\hat{y}^{(1)} \approx [7.446, 5.979, 10.101, 12.870, 14.874]$

### Step C — Compute gradients (same formula, 0.4 factor)

Compute sums  $S_j = \sum_i r_i X_{i,j}$ :

- $S_0 = \sum r_i = 4.346 + 0.179 + 2.551 + 6.470 + 4.874 = 18.420$   
 $\Rightarrow \text{grad}_0 = 0.4 \times 18.420 = 7.368$
- $S_1 = \sum r_i \cdot X_{i,1}$  ( $X \text{ col1} = [1,2,3,4,5]$ ):  
 $= 4.346 \cdot 1 + 0.179 \cdot 2 + 2.551 \cdot 3 + 6.470 \cdot 4 + 4.874 \cdot 5$   
 $= 4.346 + 0.358 + 7.653 + 25.880 + 24.370 = 62.607$   
 $\Rightarrow \text{grad}_1 = 0.4 \times 62.607 = 25.0428$
- $S_2 = \sum r_i \cdot X_{i,2}$  ( $X \text{ col2} = [2,0,1,3,2]$ ):  
 $= 4.346 \cdot 2 + 0.179 \cdot 0 + 2.551 \cdot 1 + 6.470 \cdot 3 + 4.874 \cdot 2$   
 $= 8.692 + 0 + 2.551 + 19.410 + 9.748 = 40.401$

$$\Rightarrow \text{grad}_2 = 0.4 \times 40.401 = 16.1604$$

- $S_3 = \sum r_i \cdot X_{i,3}$  ( $X \text{ col}3 = [3, 1, 2, 0, 1]$ ):

$$= 4.346 \cdot 3 + 0.179 \cdot 1 + 2.551 \cdot 2 + 6.470 \cdot 0 + 4.874 \cdot 1$$

$$= 13.038 + 0.179 + 5.102 + 0 + 4.874 = 23.193$$

$$\Rightarrow \text{grad}_3 = 0.4 \times 23.193 = 9.2772$$

Gradient vector:

$$\nabla J(\beta^{(1)}) \approx [7.368, 25.0428, 16.1604, 9.2772]^T.$$

Step D — Update parameters:

$$\beta_j^{(2)} = \beta_j^{(1)} - 0.05 \cdot \text{grad}_j.$$

Compute each:

- $\beta_0^{(2)} = 0.657 - 0.05 \times 7.368 = 0.657 - 0.3684 = 0.2886$
- $\beta_1^{(2)} = 2.259 - 0.05 \times 25.0428 = 2.259 - 1.25214 = 1.00686$
- $\beta_2^{(2)} = 1.059 - 0.05 \times 16.1604 = 1.059 - 0.80802 = 0.25098$
- $\beta_3^{(2)} = 0.804 - 0.05 \times 9.2772 = 0.804 - 0.46386 = 0.34014$

So after Iteration 2:

$$\beta^{(2)} \approx [0.28860, 1.00686, 0.25098, 0.34014]^T.$$