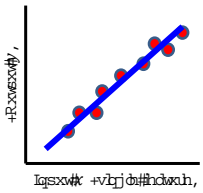
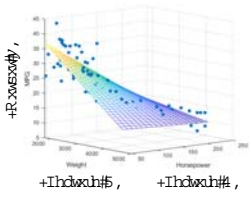
Hồi quy tuyến tính

- Đặt vấn đề
- Hồi quy tuyến tính đơn biến
- Hàm mất mát
- Hồi quy tuyến tính đa biến
- Ưu nhược điểm
- Các ứng dụng

Đặt vấn đề

- Linear regression - tìm phương trình đường thẳng (mặt phẳng/siêu phẳng) "xấp xỉ" với data huấn luyện



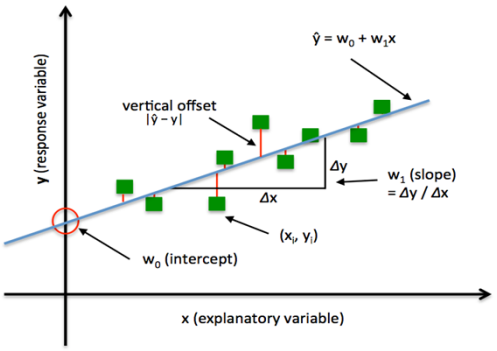


- Dựa vào phương trình để dự báo output ứng với đầu vào (input)
- So sánh hồi quy với phân lớp

Hồi quy tuyến tính đơn biến

- Dựa vào học thống kê
- x là một đặc trưng đầu vào
- y là giá trị chúng ta cần dự báo
- Mô hình hồi quy có 2 tham số: slope (w1) và y-intercept (w0)
- ε thành phần lỗi

$$y = w_0 + w_1 x + \varepsilon$$



3

Hồi quy tuyến tính đơn biến

- Xét phương trình hồi quy đơn biến có *n* quan sát, trong đó
- $y = \{y_1, y_2, \ldots, y_n\}$ là biến phụ thuộc và $x = \{x1, x2, \ldots, x_n\}$ là biến đầu vào
- Phương trình hồi quy tuyến tính đơn biến có dạng

$$\hat{y}_i = f(x_i) = w_0 + w_1 * x_i$$

4

Hàm mất mát

- Mục tiêu của mô hình học có giám sát là tìm ra một hàm số dự báo mà giá trị của chúng sai khác so với giá trị thực tế là nhỏ nhất.
- Sai khác này được đo lường thông qua các hàm mất mát (loss function).
- Huấn luyện mô hình học máy thực chất là qui về tìm cực trị của hàm mất mát. Tuỳ thuộc vào bài toán mà chúng ta có những dạng hàm mất mát khác nhau.

5

Hàm mất mát

- Hàm mất mát (loss function): Giả sử dung hàm MSE (Mean Square Error) làm hàm mất mát. Hàm này còn gọi là hàm sai số trung bình bình phương

$$\mathcal{L}(\mathbf{w}) = \frac{1}{2n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = \frac{1}{2n}\sum_{i=1}^{n}(y_i - w_0 - w_1 * x_i)^2$$

- Mục tiêu là tìm véc tơ w(w0, w1) sao cho sai số giữa giá trị dự báo và thực tế là nhỏ nhất, tức là cần tìm w0 và w1 để hàm mất mát L(w) có giá trị nhỏ nhất
- Phương pháp: Đạo hàm riêng của hàm L(w) theo w0, w1 bằng 0

6

Xác định tham số PT hồi quy TT

- Đạo hàm L(w) theo w0 bằng 0:

$$\begin{aligned}\frac{\delta \mathcal{L}(\mathbf{w})}{\delta w_0} &= \frac{-1}{n}\sum_{i=1}^{n}(y_i - w_0 - w_1 * x_i) \\ &= \frac{-1}{n}\sum_{i=1}^{n} y_i + w_0 + w_1\frac{1}{n}\sum_{i=1}^{n} x_i \\ &= -\bar{\mathbf{y}} + w_0 + w_1\bar{\mathbf{x}} \\ &= 0\end{aligned}$$

- Đạo hàm L(w) theo w1 bằng 0:

$$\begin{aligned}\frac{\delta \mathcal{L}(\mathbf{w})}{\delta w_1} &= \frac{-1}{n}\sum_{i=1}^{n} x_i(y_i - w_0 - w_1 * x_i) \\ &= \frac{-1}{n}\sum_{i=1}^{n} x_i y_i + w_0\frac{1}{n}\sum_{i=1}^{n} x_i + w_1\frac{1}{n}\sum_{i=1}^{n} x_i^2 \\ &= -\bar{\mathbf{xy}} + w_0\bar{\mathbf{x}} + w_1\bar{\mathbf{x^2}} \\ &= 0\end{aligned}$$

7

Xác định tham số PT hồi quy TT

- Tìm w0, w1

Từ phương trình (1) ta suy ra: $w_0 = \bar{\mathbf{y}} - w_1\bar{\mathbf{x}}$. Thế vào phương trình (2) ta tính được:

$$\begin{aligned}-\bar{\mathbf{xy}} + w_0\bar{\mathbf{x}} + w_1\bar{\mathbf{x^2}} &= -\bar{\mathbf{xy}} + (\bar{\mathbf{y}} - w_1\bar{\mathbf{x}})\bar{\mathbf{x}} + w_1\bar{\mathbf{x^2}} \\ &= -\bar{\mathbf{xy}} + \bar{\mathbf{y}}\bar{\mathbf{x}} - w_1\bar{\mathbf{x}}^2 + w_1\bar{\mathbf{x^2}} \\ &= 0\end{aligned}$$

Từ đó suy ra:

$$w_1 = \frac{\bar{\mathbf{x}}\bar{\mathbf{y}} - \bar{\mathbf{xy}}}{\bar{\mathbf{x}}^{\mathbf{2}} - \bar{\mathbf{x^2}}}$$

Sau khi tính được w_1 thế vào ta tính được:

$$w_0 = \bar{\mathbf{y}} - w_1\bar{\mathbf{x}}$$

8

Ví dụ 1

- ***Diện tích:*** 73.5,75.,76.5,79.,81.5,82.5,84.,85.,86.5,87.5,89.,90.,91.5
- ***Giá nhà:*** 1.49,1.50,1.51,1.54,1.58,1.59,1.60,1.62,1.63,1.64, 1.66,1.67,1.68
- Yêu cầu: Xây dựng mô hình hồi quy tuyến tính dự báo giá nhà với hàm mất mát là trung bình bình phương (MSE)
- Mô hình hồi quy đơn biến **Y= f(x) = w0 +w1.x,** trong đó f(x) là giá nhà, x là diện tích

9

Ví dụ

- Áp dụng công thức:
  - Mean(x) =?
  - Mean(y) =?
  - Mean(x*y) = ?
  - $(\text{Mean}(x))^2$ = ?
  - Mean(x^2) = ?

$$w_1 = \frac{\bar{x}\bar{y} - \bar{xy}}{\bar{x}^2 - \bar{x^2}}$$

- Áp dụng công thức

$$w_0 = \bar{y} - w_1\bar{x}$$

10

Ví dụ - code python

```
import numpy as np
# dien tich
x = np.array([73.5,75.,76.5,79.,81.5,82.5,84.,85.,86.5,87.5,89.,90.,91.5])
# gia nha (tỷ đồng)
y = np.array([1.49,1.50,1.51,1.54,1.58,1.59,1.60,1.62,1.63,1.64,1.66,1.67,1.68])
# tính trung bình
xbar = np.mean(x)
ybar = np.mean(y)
x2bar = np.mean(x**2)
xybar = np.mean(x*y)
# tính w0, w1
w1 = (xbar*ybar-xybar)/(xbar**2-(x2bar))
w0 = ybar-w1*xbar
print('w1: ', w1)
print('w0: ', w0)
#dự báo
dientich=76.5
print('giá nhà với diện tích:', w0+w1*76.5)
```

11

Ví dụ - code python với scikit-learn

```
from sklearn import linear_model
import numpy as np
# dien tich
x = np.array([[73.5,75.,76.5,79.,81.5,82.5,84.,85.,86.5,87.5,89.,90.,91.5]]).T
# gia nha (tỷ đồng)
y =
np.array([[1.49,1.50,1.51,1.54,1.58,1.59,1.60,1.62,1.63,1.64,1.66,1.67,1.68]]).T
# fit the model by Linear Regression
regr = linear_model.LinearRegression(fit_intercept=True) # fit_intercept = False
for calculating the bias
regr.fit(x, y)
# Hệ số của phương trình hồi quy
print( 'hệ số w1 : ', regr.coef_ )
print( 'hệ số w0  : ', regr.intercept_ )

#dự báo
dientich=76.5
print('giá nhà với diện tích:', w0+w1*76.5)
```

12

Hồi quy tuyến tính đa biến

- Hồi qui tuyến tính đa biến là hồi qui tuyến tính với nhiều hơn một biến đầu vào.
- Hồi qui tuyến tính đa biến phổ biến hơn so với đơn biến trong thực tế
- Để xây dựng mô hình hồi quy đa biến (*p* biến), ta cần tính được w_0, w_1, …., w_p

$$\hat{y}_i = f(x_1, x_2, \ldots, x_p) = w_0 + w_1 x_{i1} + \cdots + w_p x_{ip} = \mathbf{w}^T \mathbf{x}_i$$

- Áp dụng sklearn từ thư viên scikitlearn để tìm w_0, w_1, …., w_p

13

Ví dụ 2

- ***Diện tích: x1=*** 73.5,75.,76.5,79.,81.5,82.5,84.,85.,86.5,87.5,89.,90.,91.5
- ***Khoảng cách đến trung tâm thành phố: x2 =*** 20, 18, 17, 16, 15, 14, 12, 10, 8, 7, 5, 2, 1
- ***Giá nhà: y*** 1.49,1.50,1.51,1.54,1.58,1.59,1.60,1.62,1.63,1.64,1.66,1.67,1.68
- Yêu cầu: Xây dựng mô hình hồi quy tuyến tính dự báo giá nhà với hàm mất mát là trung bình bình phương (MSE)
- Mô hình hồi quy đa biến **Y= f(x) = w0 +w1.x1 +w2.x2,** trong đó f(x) là giá nhà, x1 là diện tích, x2 là khoảng cách tới trung tâm thành phố

14

Ví dụ 2

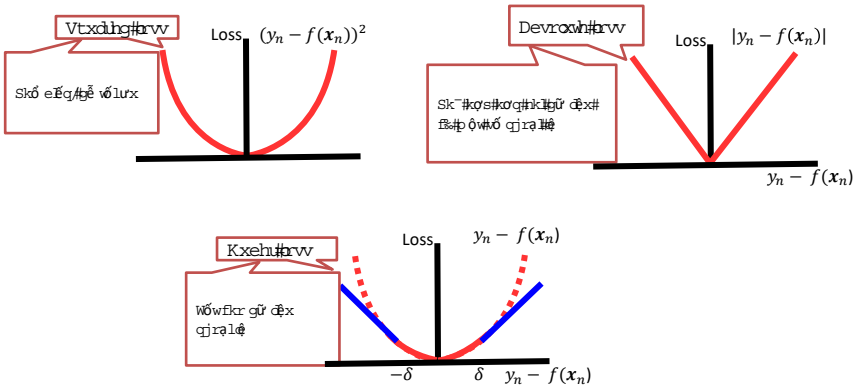
```
from sklearn import linear_model
import numpy as np
# Dien tich
x1 = np.array([[73.5,75.,76.5,79.,81.5,82.5,84.,85.,86.5,87.5,89.,90.,91.5]]).T
# Khoang cach toi trung tam
x2 = np.array([[20, 18, 17, 16, 15, 14, 12, 10, 8, 7, 5, 2, 1]]).T
# Gop cac bien dau vao X = [x1, x2] - ĐA BIẾN
X = np.concatenate([x1, x2], axis = 1)
# Giá nhà tỷ đồng
y = np.array([[1.49,1.50,1.51,1.54,1.58,1.59,1.60,1.62,1.63,1.64,1.66,1.67,1.68]]).T
# fit the model by Linear Regression
regr = linear_model.LinearRegression(fit_intercept=True) # fit_intercept = False for
calculating the bias
regr.fit(X, y)
# Hệ số của phương trình hồi quy
print( 'Coefficient : ', regr.coef_ )
print( 'Interception  : ', regr.intercept_ )
#Dự báo tự viết
```

15

Một số hàm mất mát

16

- Việc lựa chọn hàm mất mát thường phụ thuộc vào bản chất của dữ liệu. Ngoài ra, một số hàm mất mát dẫn đến vấn đề tối ưu hóa dễ dàng hơn các hàm khác.



16

Ưu nhược điểm

- Dễ hiểu, cài đặt dễ
- Chi phí tính toán thấp
- Giả định dữ liệu là độc lập
- Rất nhạy cảm với dữ liệu ngoại lệ (outlier), giải pháp Hồi quy Huber
- Không biểu diễn được các mô hình dữ liệu phức tạp
- …

17

Ứng dụng

- Dich tễ học
- Tài chính
- Kính tế
- Khoa học môi trường
- …

18