



TRƯỜNG ĐẠI HỌC
VĂN LANG

Đạo đức - Ý chí - Sáng tạo

KHOA CÔNG NGHỆ THÔNG TIN

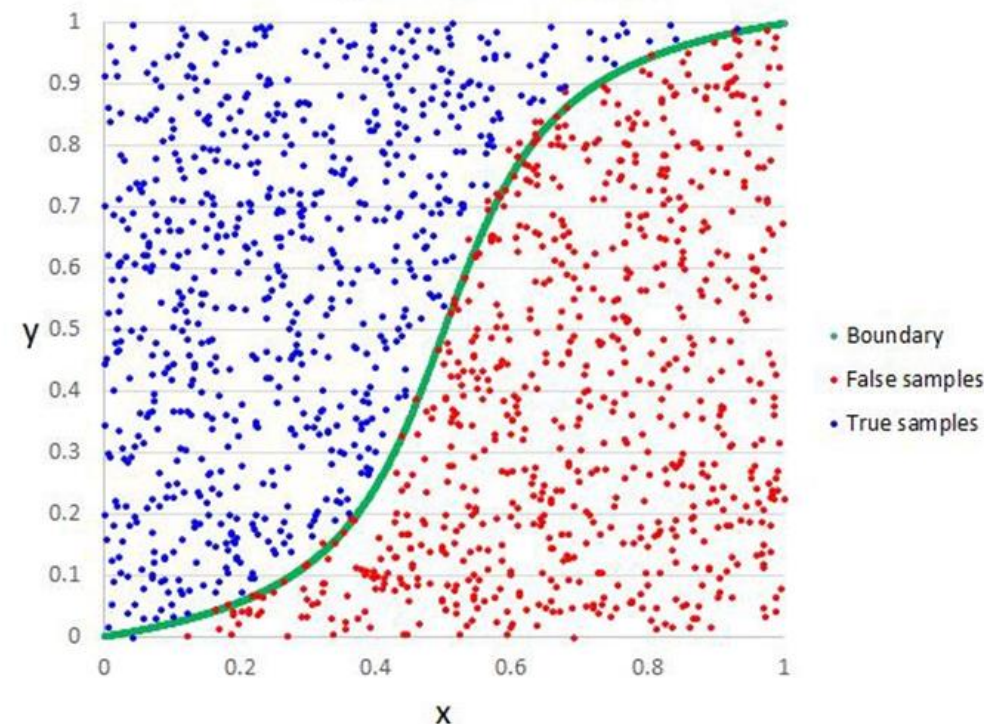
THỰC HÀNH NHẬP MÔN HỌC MÁY

THỰC HÀNH – HỒI QUY LOGISTIC

BỘ MÔN KHOA HỌC DỮ LIỆU

THỰC HÀNH – HỒI QUY LOGISTIC

- Hồi quy logistic là một thuật toán khác được mượn bởi machine learning từ lĩnh vực thống kê. Đây là phương thức tốt nhất cho các vấn đề phân loại nhị phân (hai lớp giá trị).
- Hồi quy logistic giống như hồi quy tuyến tính với mục đích là để tìm ra các giá trị cho các hệ số mà trọng lượng mỗi biến đầu vào. Không giống như hồi quy tuyến tính, dự đoán đầu ra được chuyển đổi bằng cách sử dụng một hàm không tuyến tính được gọi là hàm logistic.
- Hàm logistic trông giống như một S lớn và sẽ biến đổi bất kỳ giá trị nào thành 0 -1. Điều này rất hữu ích bởi vì chúng ta có thể áp dụng một quy tắc cho đầu ra của hàm logistic để tăng giá trị cho 0 và 1 (ví dụ nếu nhỏ hơn 0.5 sau đó đầu ra 1) và dự đoán một lớp giá trị.

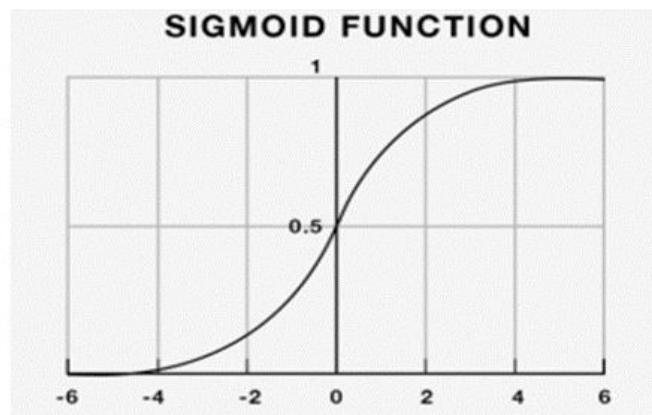


THỰC HÀNH – HỒI QUY LOGISTIC

- Vì cách mô hình được học, các dự đoán thực hiện bởi hồi quy logistic cũng có thể được sử dụng như là xác suất của một ví dụ dữ liệu nhất định thuộc lớp 0 hoặc lớp 1. Điều này có thể hữu ích cho các vấn đề khi bạn cần đưa ra nhiều lý do cho một dự đoán.
- Giống như hồi quy tuyến tính, hồi quy logistic hoạt động tốt hơn khi bạn loại bỏ các thuộc tính không liên quan đến biến đầu ra cũng như các thuộc tính tương tự nhau (correlated). Đó là một mô hình có thể học hỏi nhanh và có hiệu quả với các vấn đề phân loại nhị phân.
- Giá trị đầu ra (y) sẽ được tổng hợp từ các giá trị x thông qua phép biến đổi hàm $\log h(x) = \frac{1}{1+e^{-x}}$.
Sau đó, ta sẽ áp dụng ngưỡng để biến xác suất này thành dạng phân loại nhị phân.
- Mô hình hồi quy logistic sẽ tính toán một tổng trọng số của các biến đầu vào (tương tự như mô hình hồi quy tuyến tính), sau đó chạy kết quả qua một phương trình phi tuyến tính đặc biệt. Sử dụng một phương trình logistic hoặc sigmoid để tính đầu ra y . Trong đó, đầu ra sẽ ở dạng nhị phân, hoặc 0/1 hay -1/1.

THỰC HÀNH – HỒI QUY LOGISTIC

- Phương trình sigmoid/logistic được biểu diễn như sau: $y = \frac{1}{1+e^{-x}}$
- Biểu đồ có một đường cong dạng S tiệm cận giá trị 1 khi giá trị của biến đầu vào tăng vượt mức 0, và tiệm cận giá trị 0 khi giá trị biến đầu vào giảm thấp hơn 0. Ngoài ra, đầu ra của phương trình sigmoid sẽ đạt 0,5 khi biến đầu vào có giá trị 0.
- Vậy giá trị đầu ra là lớn hơn 0,5, ta có thể kết quả là 1 (tích cực), còn nếu thấp hơn 0,5, thì có thể coi kết quả là 0 (tiêu cực).
- Hồi quy logistic là sử dụng dữ liệu huấn luyện để tính toán giá trị của các hệ số, qua đó giảm tối đa chênh lệch giữa kết quả dự đoán và kết quả thực tế. Các hệ số trên sẽ được ước tính sử dụng kỹ thuật Ước lượng Hợp lý Cực đại (Maximum Likelihood Estimation).



THỰC HÀNH – HỒI QUY LOGISTIC

- Kỹ thuật Maximum Likelihood Estimation là một phương thức để ước tính các tham số trong mô hình thống kê. Trong đó, độ khả thi có thể được tối đa hóa thông qua nhiều kỹ thuật, ví dụ như thuật toán tối ưu.
- Có thể sử dụng Phương thức Newton để giá trị cực đại và cực tiểu của nhiều phương trình, bao gồm cả phương trình tính độ khả thi. Ngoài ra, phương thức Gradient Descent cũng có thể được sử dụng để thay thế.
- Ví dụ như dự đoán khả năng xảy ra của một sự kiện, sự kiện thật sự xảy ra sẽ được gọi là 1. Chẳng hạn như dự đoán khả năng bị ốm của một người, thì các trường hợp ốm sẽ được gọi là 1. Phương thức này mang tên phương trình chuyển đổi logistic được sử dụng, cụ thể là $h(x) = \frac{1}{1+e^{-x}}$, được biểu hiện bằng một đường cong có dạng S.
- Thuật toán này thường được sử dụng trong đánh giá tín dụng, độ đo xác suất thành công của các chiến lược marketing, hay như dự đoán doanh thu của một sản phẩm cụ thể.



THỰC HÀNH – HỒI QUY LOGISTIC

Bước 1: Thu thập dữ liệu / tập dữ liệu;

Bước 2: import các gói, lớp và thư viện hàm;

Bước 3: Xây dựng khung dữ liệu;

Bước 4: Tạo mô hình(hồi quy logistic);

Bước 5: Dự đoán bằng cách sử dụng Tập dữ liệu kiểm tra
và kiểm tra điểm số;

Bước 6: Dự đoán với Tập dữ liệu mới và đánh giá độ chính
xác;



THỰC HÀNH – HỒI QUY LOGISTIC

#Thực hành 01

#Step 1: Import Packages, Functions, and Classes

```
>>> import matplotlib
>>> import matplotlib.pyplot as plt
>>> import numpy as np
>>> from sklearn.linear_model import LogisticRegression
>>> from sklearn.metrics import classification_report,
confusion_matrix
```

#Step 2: Get Data

```
>>> x = np.arange(10).reshape(-1, 1)
>>> y = np.array([0, 0, 0, 0, 1, 1, 1, 1, 1, 1])
>>> x
>>> y
array([0, 0, 0, 0, 1, 1, 1, 1, 1, 1])
```

THỰC HÀNH – HỒI QUY LOGISTIC

#Step 3: tạo mô hình & huấn luyện

```
>>> model = LogisticRegression(solver='liblinear', random_state=0)
>>> model.fit(x, y)
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                    intercept_scaling=1, max_iter=100, multi_class='warn',
                    n_jobs=None, penalty='l2', random_state=0, solver='liblinear',
                    tol=0.0001, verbose=0, warm_start=False)
>>> model = LogisticRegression(solver='liblinear', random_state=0).fit(x, y)
>>> model.classes_
array([0, 1])
>>> model.intercept_
array([-1.04608067])
>>> model.coef_
array([[0.51491375]])
```


#Step 4: đánh giá mô hình

```
>>> model.predict_proba(x)
array([[0.74002157, 0.25997843],
       [0.62975524, 0.37024476],
       [0.5040632 , 0.4959368 ],
       [0.37785549, 0.62214451],
       [0.26628093, 0.73371907],
       [0.17821501, 0.82178499],
       [0.11472079, 0.88527921],
       [0.07186982, 0.92813018],
       [0.04422513, 0.95577487],
       [0.02690569, 0.97309431]])

>>> model.predict(x)
array([0, 0, 0, 1, 1, 1, 1, 1, 1, 1])

>>> model.score(x, y)
0.9

>>> confusion_matrix(y, model.predict(x))
array([[3, 1],
       [0, 6]], dtype=int64)
```

THỰC HÀNH – HỒI QUY LOGISTIC

```
>>> cm = confusion_matrix(y, model.predict(x))
>>> fig, ax = plt.subplots(figsize=(8, 8))
>>> ax.imshow(cm)
<matplotlib.image.AxesImage object at 0x11EBF990>
>>> ax.grid(False)
>>> ax.xaxis.set(ticks=(0, 1), ticklabels=('Predicted 0s', 'Predicted 1s'))
[[<matplotlib.axis.XTick object at 0x1042D850>, <matplotlib.axis.XTick object
at 0x1042D450>], [Text(0, 0, 'Predicted 0s'), Text(0, 0, 'Predicted 1s')]]
>>> ax.yaxis.set(ticks=(0, 1), ticklabels=('Actual 0s', 'Actual 1s'))
[[<matplotlib.axis.YTick object at 0x1042DF90>, <matplotlib.axis.YTick object
at 0x1042DBF0>], [Text(0, 0, 'Actual 0s'), Text(0, 0, 'Actual 1s')]]
>>> ax.set_ylim(1.5, -0.5)
(1.5, -0.5)
```

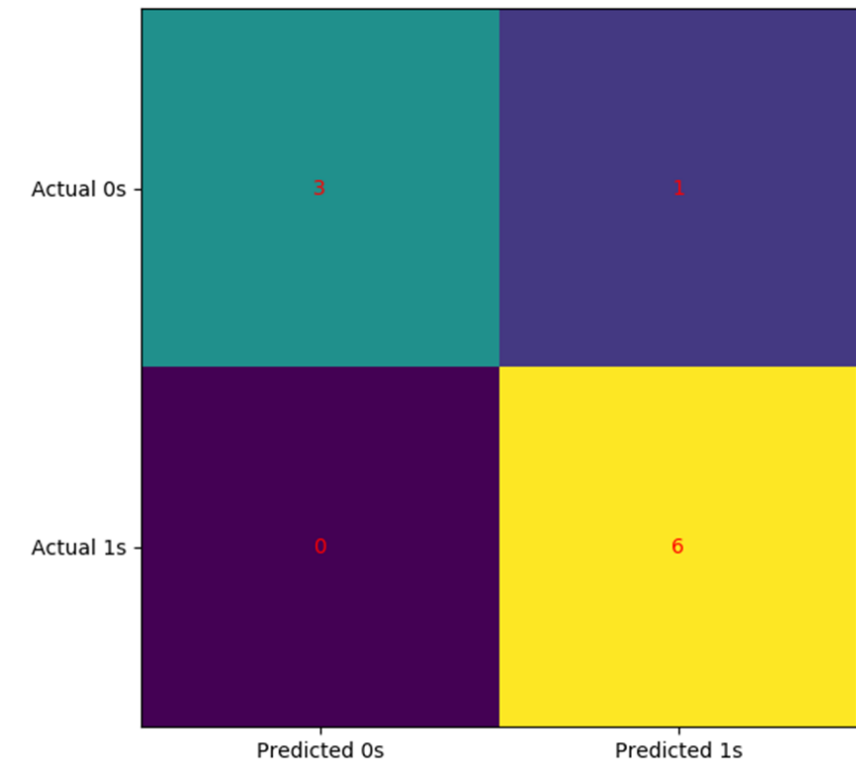
THỰC HÀNH – HỒI QUY LOGISTIC

```
>>> for i in range(2):
    for j in range(2):
        ax.text(j, i, cm[i, j], ha='center',
va='center', color='red')
Text(0, 0, '3')
Text(1, 0, '1')
Text(0, 1, '0')
Text(1, 1, '6')
>>> plt.show()

>>> print(classification_report(y, model.predict(x)))
              precision    recall  f1-score   support

     0           1.00        0.75     0.86         4
     1           0.86        1.00     0.92         6

 micro avg       0.90        0.90     0.90        10
 macro avg       0.93        0.88     0.89        10
weighted avg       0.91        0.90     0.90        10
```



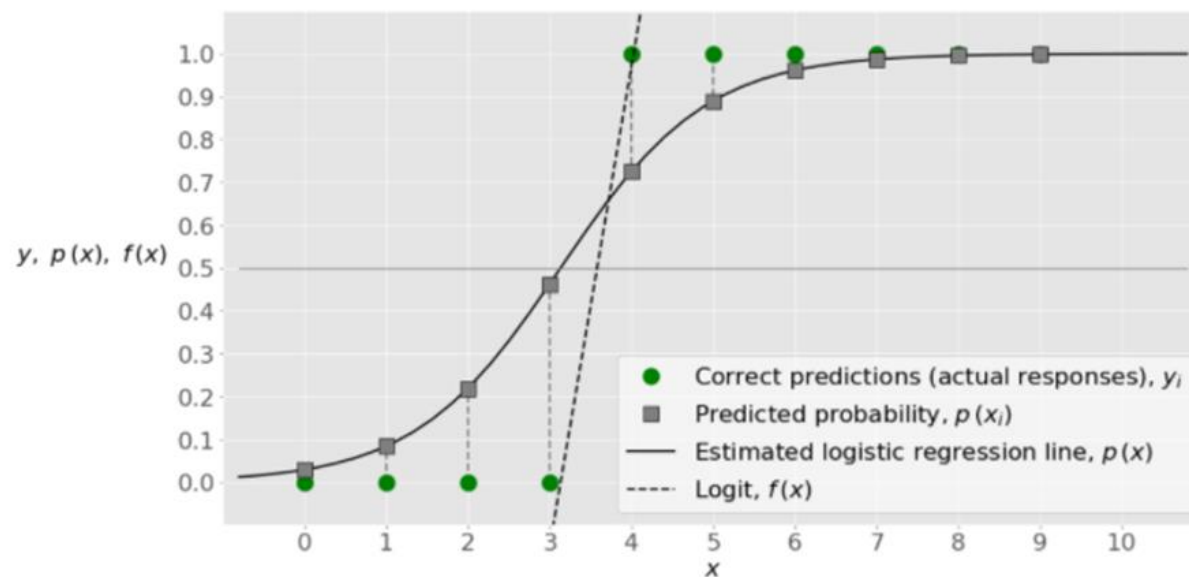
THỰC HÀNH – HỒI QUY LOGISTIC

```
>>> model = LogisticRegression(solver='liblinear', C=10.0,
random_state=0)
>>> model.fit(x, y)
LogisticRegression(C=10.0, class_weight=None, dual=False,
fit_intercept=True,
                    intercept_scaling=1, max_iter=100, multi_class='warn',
                    n_jobs=None, penalty='l2', random_state=0,
solver='liblinear',
                    tol=0.0001, verbose=0, warm_start=False)
>>> model.intercept_
array([-3.51335372])
>>> model.coef_
array([[1.12066084]])
>>> model.predict_proba(x)
array([[0.97106534, 0.02893466],
       [0.9162684 , 0.0837316 ],
       [0.7810904 , 0.2189096 ],
       [0.53777071, 0.46222929],
       [0.27502212, 0.72497788],
       [0.11007743, 0.88992257],
       [0.03876835, 0.96123165],
       [0.01298011, 0.98701989],
       [0.0042697 , 0.9957303 ],
       [0.00139621, 0.99860379]])
>>> model.predict(x)
array([0, 0, 0, 0, 1, 1, 1, 1, 1])
>>> model.score(x, y)
1.0
```

THỰC HÀNH – HỒI QUY LOGISTIC

```
>>> confusion_matrix(y, model.predict(x))
array([[4, 0],
       [0, 6]], dtype=int64)
>>> print(classification_report(y, model.predict(x)))
```

	precision	recall	f1-score	support
0	1.00	1.00	1.00	4
1	1.00	1.00	1.00	6
micro avg	1.00	1.00	1.00	10
macro avg	1.00	1.00	1.00	10
weighted avg	1.00	1.00	1.00	10



THỰC HÀNH – HỒI QUY LOGISTIC

#Thực hành 02

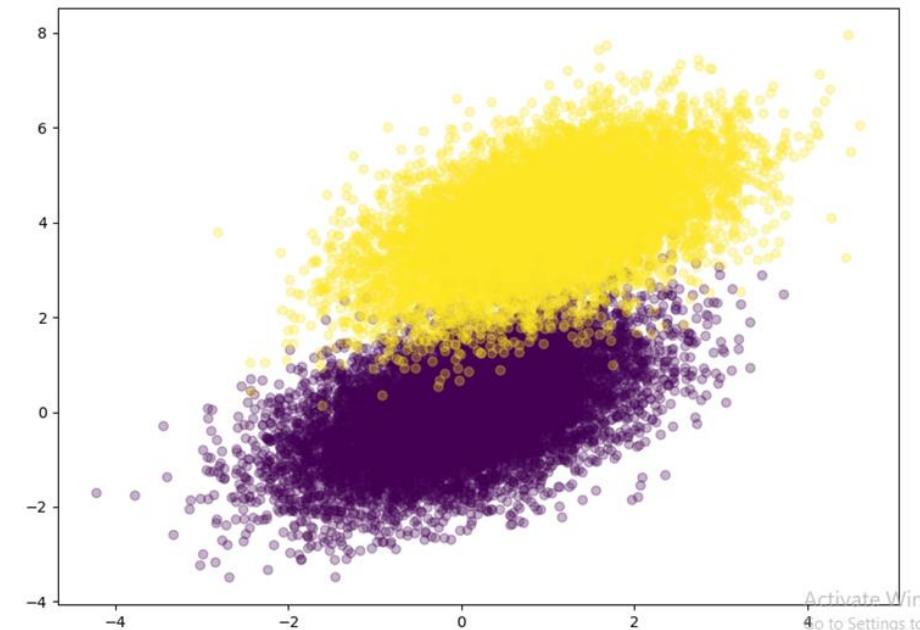
```
>>> import numpy as np
>>> import matplotlib.pyplot as plt
>>> def sigmoid(scores):    #hàm sigmoid
    return 1 / (1 + np.exp(-scores))
>>> def log_likelihood(features, target, weights):
    #hàm Ước lượng Cực đại
    scores = np.dot(features, weights)
    ll = np.sum(target * scores - np.log(1 + np.exp(scores)))
    return ll
>>> def logistic_regression(features, target, num_steps,
learning_rate, add_intercept=False):
    if add_intercept:
        intercept = np.ones((features.shape[0], 1))
        features = np.hstack((intercept, features))
        weights = np.zeros(features.shape[1])

    for step in range(num_steps):
        scores = np.dot(features, weights)
        predictions = sigmoid(scores)
        output_error_signal = target - predictions
        gradient = np.dot(features.T, output_error_signal)
        weights += learning_rate * gradient

    if step % 10000 == 0:
        print(log_likelihood(features, target, weights))
    return weights
```

THỰC HÀNH – HỒI QUY LOGISTIC

```
>>> np.random.seed(10)
>>> num_observations = 10000
>>> x1 = np.random.multivariate_normal([0, 0], [[1, 0.5], [0.5, 1]],
num_observations)
>>> x2 = np.random.multivariate_normal([1, 4], [[1, 0.5], [0.5, 1]],
num_observations)
>>> simulated_separableish_features = np.vstack((x1,
x2)).astype(np.float32)
>>> simulated_labels = np.hstack((np.zeros(num_observations),
np.ones(num_observations)))
>>> plt.figure(figsize=(10, 8))
<Figure size 1000x800 with 0 Axes>
>>> plt.scatter(simulated_separableish_features[:,
0], simulated_separableish_features[:, 1],
c=simulated_labels, alpha=0.3,)
<matplotlib.collections.PathCollection object at 0x012A5070>
>>> plt.show()
```





TRƯỜNG ĐẠI HỌC
VĂN LANG

Đạo đức - Ý chí - Sáng tạo

KHOA CÔNG NGHỆ THÔNG TIN

