

Soft computing



K.N. Toosi
University of
Technology


محاسبات نرم



Hasan Ghasemzadeh
<http://wp.kntu.ac.ir/ghasemzadeh>

Soft Computing

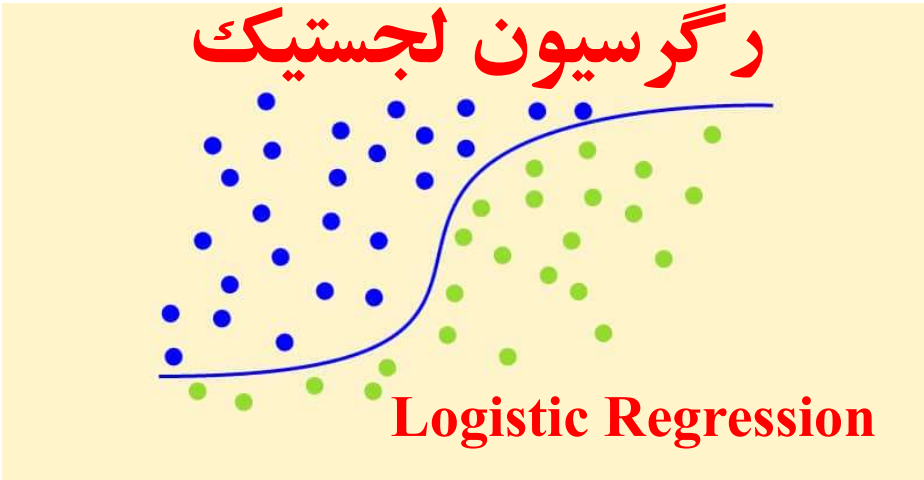
Artificial Intelligence and Soft computing



K.N. TOOSI
University of
Technology

طبقه بندی

رگرسیون لجستیک



Logistic Regression

Soft Computing

2

نمونه مسایل طبقه بندی

Classification- Binary decision

مسایل زیادی تصمیم گیری دودویی هستند و منجر به ایجاد یک طبقه بندی می شوند نظیر:

- طبقه بندی ایمیل ها به دو دسته ای میل های اسپم یا غیر اسپم
- طبقه بندی معاملات آنلاین به دو دسته کلاهبرداری یا غیر کلاهبرداری
- طبقه بندی تومورهای بدخیم یا خوش خیم.

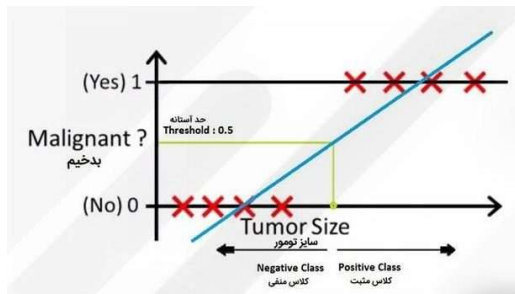
$$y \in \{0, 1\}$$

0: "Negative Class" (e.g., benign tumor)

1: "Positive Class" (e.g., malignant tumor)

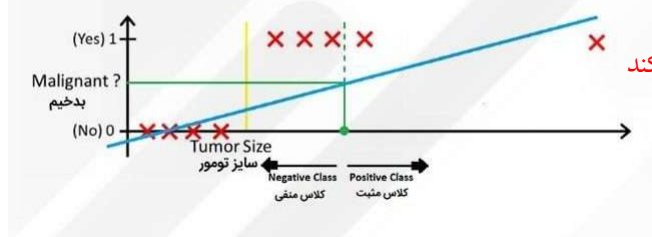
Artificial Intelligence

طبقه بندی و رگرسیون



If $h_{\theta}(x) \geq 0.5$, predict "y = 1"

If $h_{\theta}(x) < 0.5$, predict "y = 0"



داده ی پرت (Outlier)
حدس را دچار خطا می کند

Soft Computing

4

طبقه بندی و رگرسیون

$P(Y = 0|x) + P(Y = 1|x) = 1 \quad \text{or} \quad P(Y|x) = 1$

احتمال بین صفر و یک است و خارج از آن مفهوم ندارد

Soft Computing 5

طبقه بندی و رگرسیون

برچسب

$$P(Y = 1|x, \theta) = \sigma(\theta^T x)$$

$$P(Y = 0|x, \theta) = 1 - \sigma(\theta^T x)$$

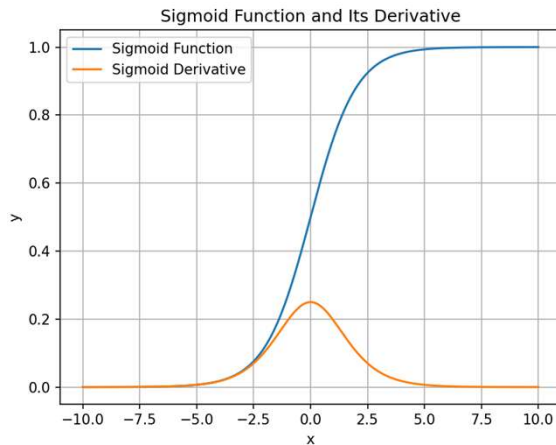
$$P(Y|x, \theta) = 1$$

$$\sigma(\theta^T x) = ?$$

تابع سیگموئید یک تابع مناسب مسایل تصمیم گیری دوگانه است و در رگرسیون لجستیک از این تابع استفاده می شود.

Soft Computing 6

تابع سیگموئید



$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

$$\sigma'(x) = \frac{e^{-x}}{(1 + e^{-x})^2} = \sigma(x)(1 - \sigma(x))$$

مثال: توزیع برنولی یک توزیع گسسته است که مقادیر یک (در صورت موفقیت آزمایش) و صفر را (در صورت شکست) می‌گیرد. احتمال موفقیت آزمایش برابر p و احتمال شکست آن برابر $q=1-p$ است (مثال احتمال در شیر یا خط).

تابع سیگموئید برای توزیع برنولی مناسب است

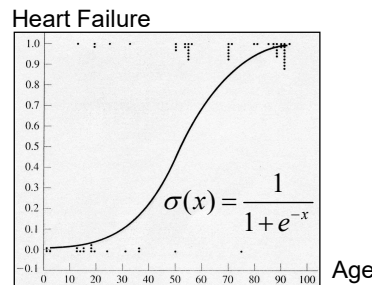
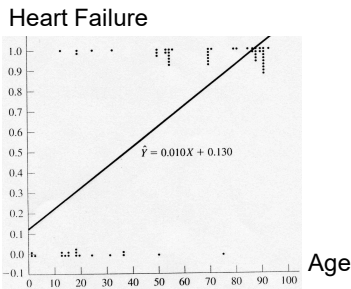
$$\hat{y} = P(Y = 1|x) = p(x) = \sigma(x)$$

Soft Computing

7

مثال رگرسیون لجستیک

احتمال مرگ ناشی از سکته قلبی



- Linear regression Extrapolates outside 0-1 and not as good empirically
- It is not appropriate for probability
- Sigmoidal is bounded between 0 and 1
- It is differentiable

Soft Computing

8

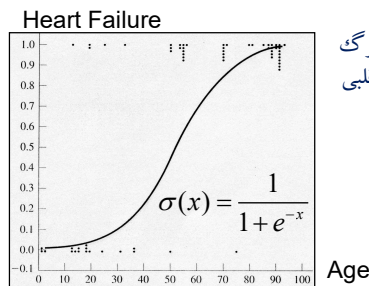
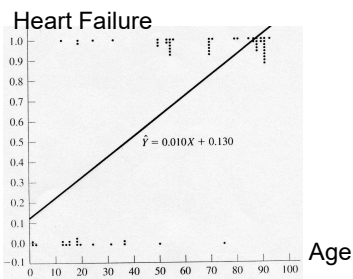
رگرسیون لجستیک

تابع فرضی

$$h_{\theta}(x) = \theta_0 + \theta_1 x$$

$$h_{\theta}(x) = \frac{1}{1 + e^{-(\theta_0 + \theta_1 x)}}$$

رگرسیون خطی با یک ویژگی
رگرسیون لجستیک با یک ویژگی



مثال: احتمال مرگ ناشی از سکته قلبی

- Linear regression Extrapolates outside 0-1 and not as good empirically
- It is not appropriate for probability
- Sigmoidal is bounded between 0 and 1
- It is differentiable

9

نسبت احتمال - بخت

نسبت احتمال - بخت (Odds)

محاسبه بخت بر اساس نسبت احتمال وقوع به احتمال عدم وقوع انجام می شود. بنابراین مقدار بخت، بر خلاف مقدار احتمال می تواند بزرگتر از یک نیز باشد. بطور مثال فرض کنید احتمال خوش خیم بودن غده سرطانی برای یک شخص $\frac{4}{7}$ برآورد شده است، بنابراین مقدار بخت برای این فرد $\frac{4}{3}$ می باشد.

Probability vs Odds

	Risk	Odds
Mathematically	$P(p) = \frac{p}{p+q}$	$O(p) = \frac{\frac{p}{p+q}}{\frac{q}{p+q}} = \frac{p(p+q)}{q(p+q)} = \frac{p}{q}$
Graphically		

$$\hat{y} = P(Y = 1|x) = p(x)$$

$$p(x) = \frac{1}{1 + e^{-(\theta_0 + \theta_1 x)}}$$

$$Odds(\pi(x)) = \frac{p(x)}{1 - p(x)} = e^{(\theta_0 + \theta_1 x)}$$

Soft Computing

10

لگاریتم طبیعی بخت

تبدیل لوجیت (Logit Transformation)

لگاریتم طبیعی بخت به عنوان تبدیل لوجیت یا (Log-Odds) شناخته می شود. در رگرسیون لجستیک با استفاده از تبدیل لوجیت بر روی احتمال وقوع X (برآورد مقدار Y) می توان رابطه خطی رگرسیون را به شکل زیر نمایش داد.

$$\text{Logit}(p(x)) = \text{Ln}(\text{Odds}(p(x))) = \text{Ln}\left(\frac{p(x)}{1-p(x)}\right) = \text{Ln}\left(e^{(\theta_0 + \theta_1 x_1)}\right) = \theta_0 + \theta_1 x_1$$

با تبدیل لوجیت یک رابطه خطی می توان نوشت

$$\hat{y} = E(Y|X=x) = P(Y=1|X=x) = p(x) = \frac{1}{1 + e^{-(\theta_0 + \theta_1 x_1)}}$$

مقدار احتمال از لوجیت می تواند محاسبه شود

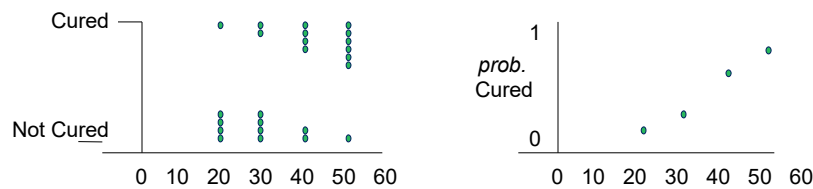
$$\text{Logit}(p(x)) = \text{Ln}\left(\frac{p(x)}{1-p(x)}\right) \Rightarrow \frac{p(x)}{1-p(x)} = e^{\text{Logit}(p(x))} \Rightarrow p(x) = \frac{e^{\text{Logit}(p(x))}}{1 + e^{\text{Logit}(p(x))}}$$

Soft Computing

11

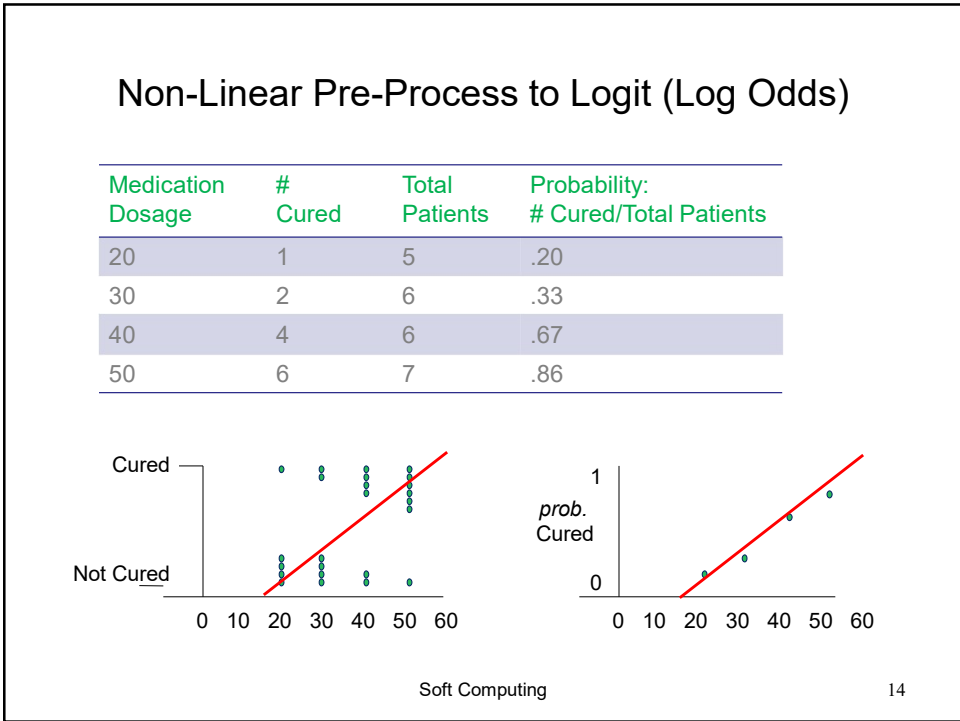
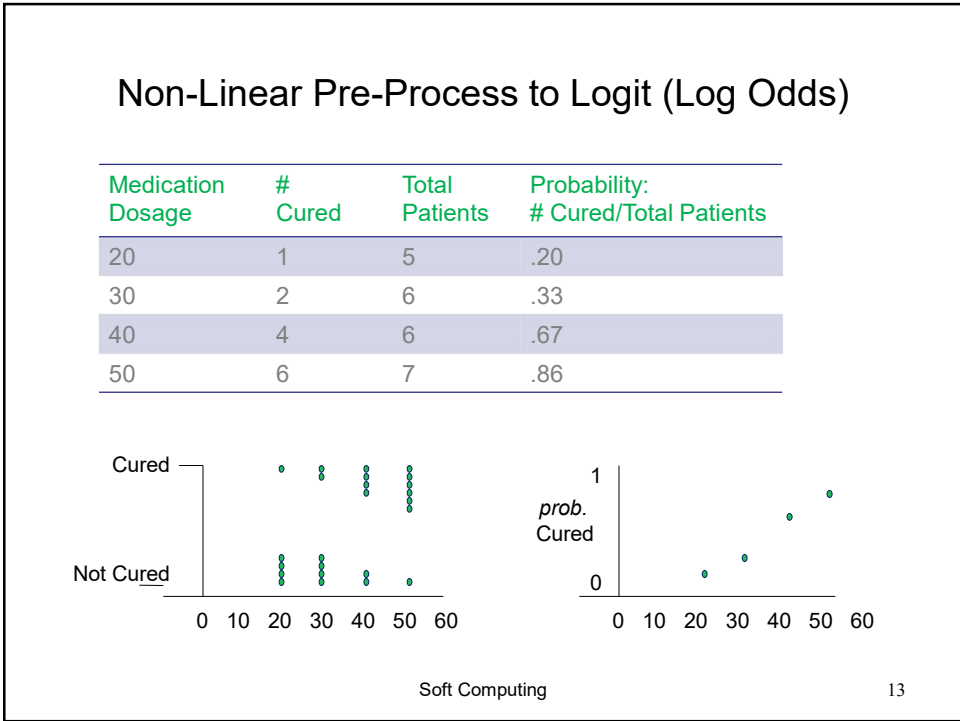
Non-Linear Pre-Process to Logit (Log Odds)

Medication Dosage	# Cured	Total Patients	Probability: # Cured/Total Patients
20	1	5	.20
30	2	6	.33
40	4	6	.67
50	6	7	.86



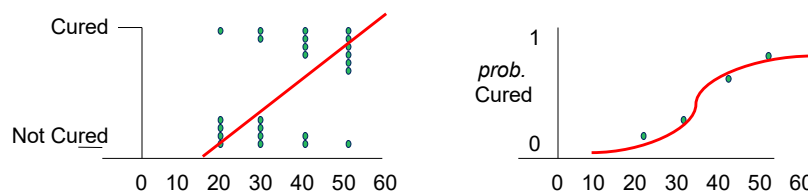
Soft Computing

12



Non-Linear Pre-Process to Logit (Log Odds)

آیا از رگرسیون خطی بدون برون یابی نامطلوب می توان استفاده کرد؟
 مشابه رگرسیون غیر خطی می توان از رگرسیون خطی بر روی لوجیت استفاده نمود.

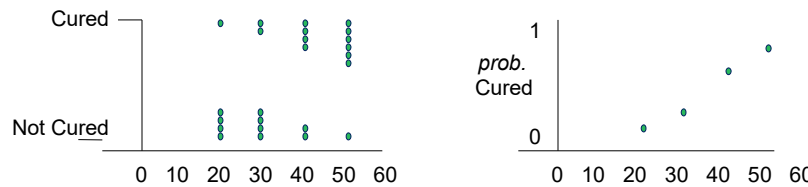


Soft Computing

15

Non-Linear Pre-Process to Logit (Log Odds)

Medication Dosage	# Cured	Total Patients	Probability: # Cured/Total Patients	Odds: $p/(1-p) = \frac{\# \text{ cured}}{\# \text{ not cured}}$	Logit Log Odds: $\ln(\text{Odds})$
20	1	5	.20	.25	-1.39
30	2	6	.33	.50	-0.69
40	4	6	.67	2.0	0.69
50	6	7	.86	6.0	1.79

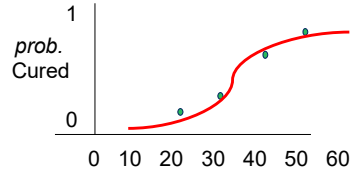


Soft Computing

16

Non-Linear Pre-Process to Logit (Log Odds)

Medication Dosage	# Cured	Total Patients	Probability: # Cured/Total Patients	Odds: p/(1-p) = # cured/# not cured	Logit Log Odds: ln(Odds)
20	1	5	.20	.25	-1.39
30	2	6	.33	.50	-0.69
40	4	6	.67	2.0	0.69
50	6	7	.86	6.0	1.79

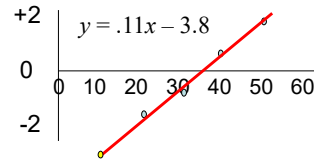


مثال: مقدار احتمال p برای میزان مصرف ۱۰ (دقت کنید که مساله برون یابی است)

$$\text{Logit}(10) = .11(10) - 3.8 = -2.7$$

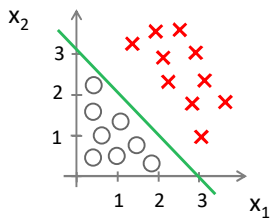
$$p(x) = \frac{e^{\text{Logit}(p(x))}}{1 + e^{\text{Logit}(p(x))}} \rightarrow p(x) = \frac{e^{-2.7}}{1 + e^{-2.7}} = 0.06$$

با استفاده از فرمول مقدار احتمال از طریق لوجیت قابل محاسبه است



مرز خطی تصمیم گیری

Linear Decision Boundary

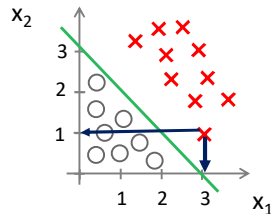


Predict $y = 1$ if $-3 + x_1 + x_2 \geq 0$

$$h(\theta) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 = \theta^T x$$

$$x_2 = 3 - x_1 \Rightarrow 3 - x_1 - x_2 = 0 \quad \text{معادله خط:}$$

$$\theta^T = [-3 \quad 1 \quad 1] \quad x = \begin{bmatrix} 1 \\ x_1 \\ x_2 \end{bmatrix} \quad h(\theta) = -3 + x_1 + x_2$$

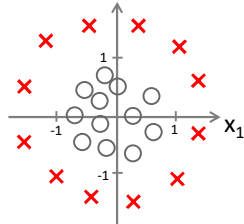


مثال: برای نقطه نمایش داده شده کلاس را مشخص کنید

$$x = \begin{bmatrix} 1 \\ 3 \\ 1 \end{bmatrix} \rightarrow h(\theta) = -3 + 3 + 1 = 1 > 0 \Rightarrow y = 1$$

مرزهای غیر خطی تصمیم گیری

Non-linear decision boundaries



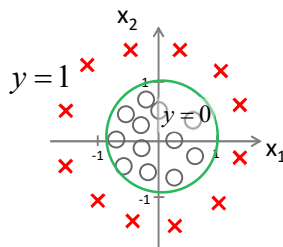
اگر مرز داده ها خطی نباشد چگونه عمل کنیم؟

Soft Computing

19

مرزهای غیر خطی تصمیم گیری

مثال:



Predict $y=1$ if $-1+x_1^2+x_2^2 \geq 0$

استفاده از تابع چند جمله ای با مرتبه دو

$$h(\theta) = \theta^T x$$

$$h(\theta) = \theta^T x = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2$$

$$x_1^2 + x_2^2 = 1 \Rightarrow -1 + x_1^2 + x_2^2 = 0 \quad \text{معادله دایره:}$$

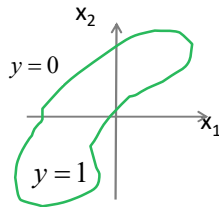
$$h(\theta) = \theta^T x = -1 + 0 \times x_1 + 0 \times x_2 + 1 \times x_1^2 + 1 \times x_2^2$$

$$\theta^T = [-1 \quad 0 \quad 0 \quad 1 \quad 1] \quad x = \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ x_1^2 \\ x_2^2 \end{bmatrix}$$

Soft Computing

20

مرزهای غیرخطی تصمیم گیری



استفاده از تابع چند جمله ای با مرتبه های بالاتر

$$h_{\theta}(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_1^2 x_2 + \theta_5 x_1^2 x_2^2 + \theta_6 x_1^3 x_2 + \dots)$$

مساله تعیین مرز تصمیم گیری یک مساله طبقه بندی است.
این مساله برچسب دار است و یادگیری ماشین با نظارت است.

$$h(\theta) = \theta^T x$$

θ چگونه تعیین می شود؟

Soft Computing

21

رگرسیون لجستیک

رگرسیون لجستیک با چند ویژگی

$$h_{\theta}(x) = \frac{1}{1 + e^{-(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \dots)}}$$

تابع فرضی

تابع هزینه: (اثبات از ماکزیمم احتمال)

$$J(\theta) = \frac{1}{m} \sum_{i=1}^m \text{Cost}(h_{\theta}(x^{(i)}), y^{(i)}) = -\frac{1}{m} \left[\sum_{i=1}^m y^{(i)} \log h_{\theta}(x^{(i)}) + (1 - y^{(i)}) \log (1 - h_{\theta}(x^{(i)})) \right]$$

To fit parameters: $\theta \min_{\theta} J(\theta)$

حداقل کردن تابع هزینه و تعیین ضرایب

Soft Computing

22

رگرسیون لجستیک

رگرسیون لجستیک با چند ویژگی

$$h_{\theta}(x) = \frac{1}{1 + e^{-(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \dots)}}$$

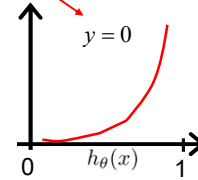
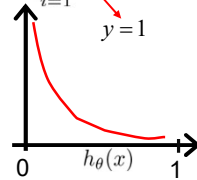
تابع فرضیه

تابع هزینه

$$J(\theta) = \frac{1}{m} \sum_{i=1}^m \text{Cost}(h_{\theta}(x^{(i)}), y^{(i)}) = -\frac{1}{m} \left[\sum_{i=1}^m y^{(i)} \log h_{\theta}(x^{(i)}) + (1 - y^{(i)}) \log (1 - h_{\theta}(x^{(i)})) \right]$$

حداقل کردن تابع هزینه و تعیین ضرایب

To fit parameters: $\theta \min_{\theta} J(\theta)$



To make a prediction given new: x

تخمین برای شرایط جدید

Output $h_{\theta}(x) = \frac{1}{1 + e^{-\theta^T x}}$

Soft Computing

23

رگرسیون لجستیک

در رگرسیون لجستیک تابع هزینه لگاریتمی انتخاب می شود

- این تابع باعث می شود مدل، پیش بینی های خود را برای نمونه هایی که احتمال بالایی دارند به دقت انجام دهد.
- تابع لگاریتمی هزینه، محدب است (مشتق دوم آن یک مقدار مثبت است، به این معنا که تنها یک مینیمم دارد و در نتیجه، الگوریتم های بهینه سازی می توانند به مینیمم مطلق برسند و مدل به خوبی آموزش ببیند

```
from sklearn.linear_model import LogisticRegression
```

```
# Train the logistic regression model
model = LogisticRegression()
model.fit(X, y)
```

Soft Computing

24

روشهای حداقل کردن تابع هزینه

گرادیان کاهشی (Gradient Descent)
 گرادیان کاهشی دسته‌ای (BGD: Batch Gradient Descent)
 گرادیان کاهشی تصادفی (SGD: Stochastic Gradient Descent)
 مختصات کاهشی (Coordinate Descent)
 گرادیان مزدوج (CG: Conjugate Gradient)

روش‌های مبتنی بر بهینه‌سازی عددی مانند
 روش نیوتن

روش شبه نیوتنی (Broyden–Fletcher–Goldfarb–Shanno) BFGS
 روش شبه نیوتنی (Limited-memory BFGS) L-BFGS

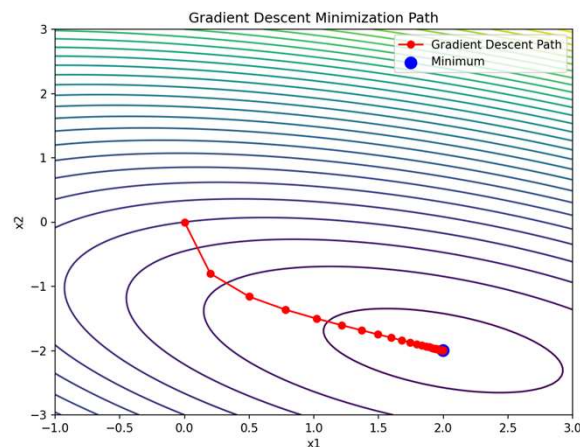
- روش‌های دیگری نیز وجود دارد که در اینجا توضیح داده نمی‌شود و می‌توانید مطالعه نموده و از آنها استفاده کنید مانند:
 Mini-Batch Gradient Descent, Momentum-Based Methods, Adam (Adaptive Moment Estimation), Trust Region Methods, Trust Region Methods
 - در کتابخانه sklearn برای مدل‌های مختلف رگرسیون روش‌های مختلفی از حداقل کردن تابع هزینه گنجانده شده است.

Soft Computing

25

روشهای حداقل کردن تابع هزینه

گرادیان کاهشی (Gradient Descent)



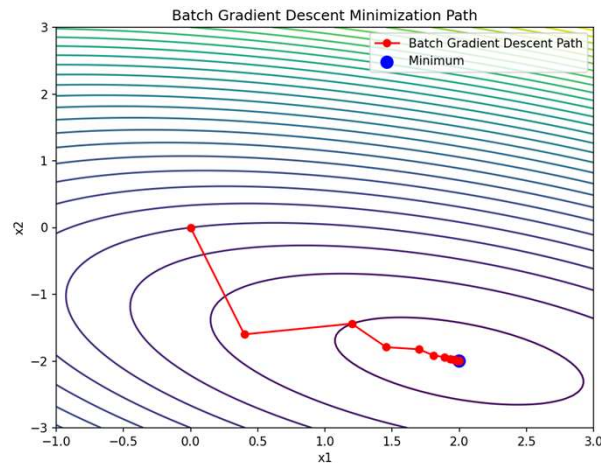
گرادیان کاهشی از تمام داده‌ها یکجا استفاده می‌کند و زمان محاسبات را در داده‌های حجیم خیلی زیاد خواهد شد.

Soft Computing

26

روشهای حداقل کردن تابع هزینه

گرادیان کاهشی دسته‌ای (BGD: Batch Gradient Descent)



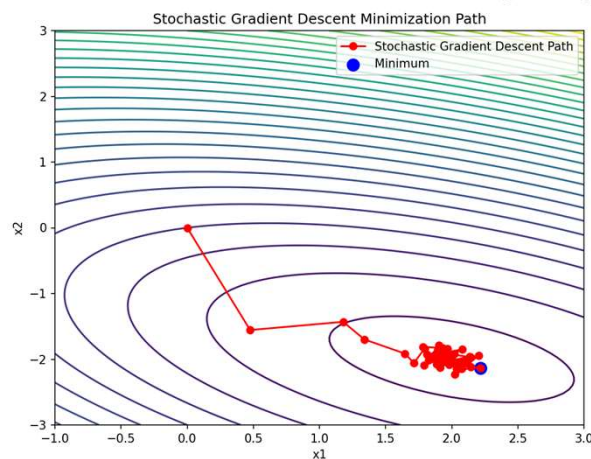
گرادیان کاهشی دسته‌ای از تمام داده ها یکجا استفاده نمی کند و زمان محاسبات را کاهش می دهد

Soft Computing

27

روشهای حداقل کردن تابع هزینه

گرادیان کاهشی تصادفی (SGD: Stochastic Gradient Descent)



روش مشابه

`solver=`
`'sag', 'saga'`

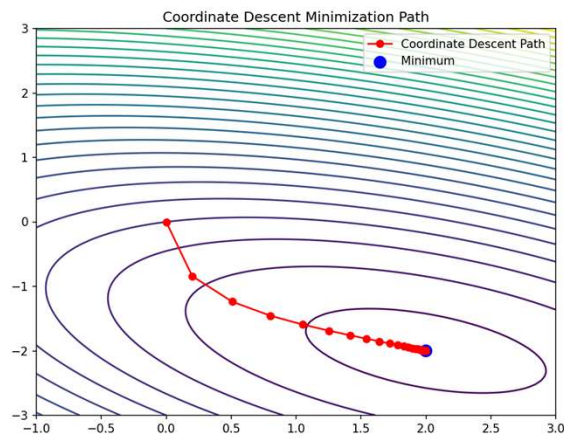
گرادیان کاهشی تصادفی از داده ی تصادفی استفاده می کند و در هر مرحله محاسبات کم است اما تعداد سعی و خطا زیاد است.
برای داده های زیاد مناسب است

Soft Computing

28

روشهای حداقل کردن تابع هزینه

مختصات کاهششی (Coordinate Descent)



روش مشابه

`solver='liblinear'`

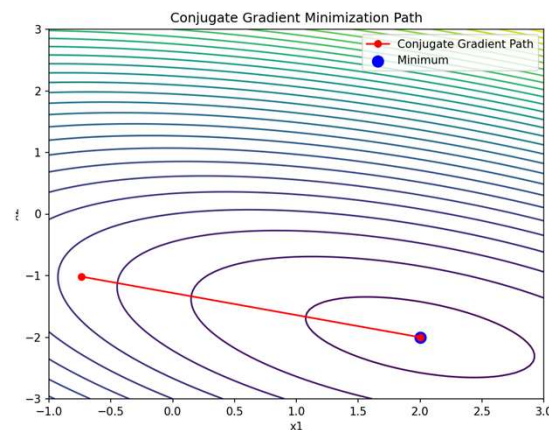
مشابه گرادیان کاهششی است ولی در هر گام تمام متغیر همزمان بروز نمی شوند. در هر گام یک متغیر بروز می شود در حالیکه بقیه متغیرها ثابت فرض می شوند. برای مدل‌های خاصی (مانند رگرسیون لسو) بسیار کارآمد است.

29

Soft Computing

روشهای حداقل کردن تابع هزینه

گرادیان مزدوج (CG: Conjugate Gradient)



`solver='newton-cg'`

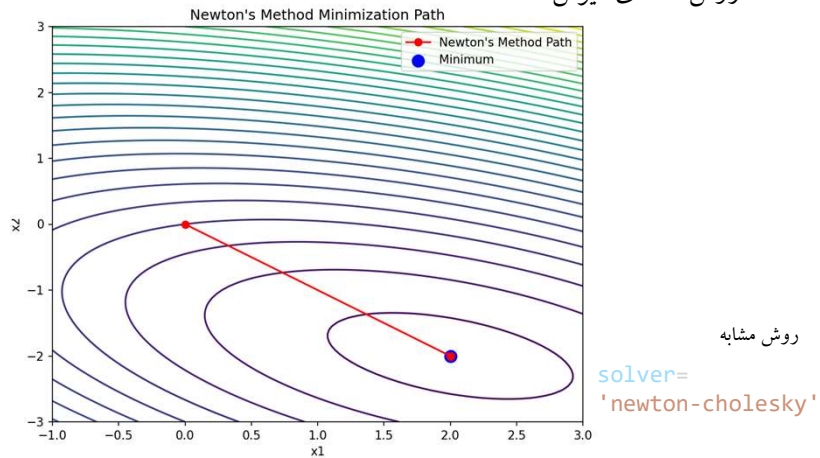
روش گرادیان مزدوج برای توابع کوادراتیک مناسب است. از جهت عمود بر تابع هزینه استفاده می کند و برای توابع کوادراتیک بسیار سریع است

30

Soft Computing

روشهای حداقل کردن تابع هزینه

روش عددی نیوتن



روش مشابه

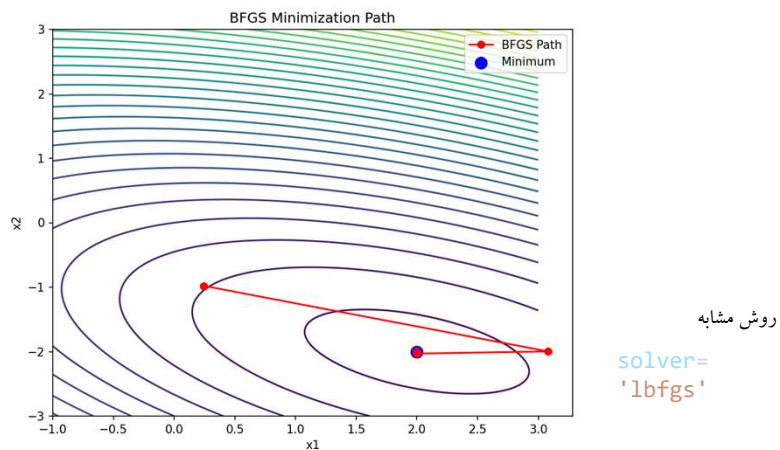
روش عددی نیوتن برای داده های کم مناسب است. این روش در هر مرحله ماتریس مشتقات مرتبه دوم (هسیان) را حساب می کند و زمان زیادی می برد

Soft Computing

31

روشهای حداقل کردن تابع هزینه

روش عددی شبه نیوتنی (Broyden–Fletcher–Goldfarb–Shanno) BFGS



روش مشابه

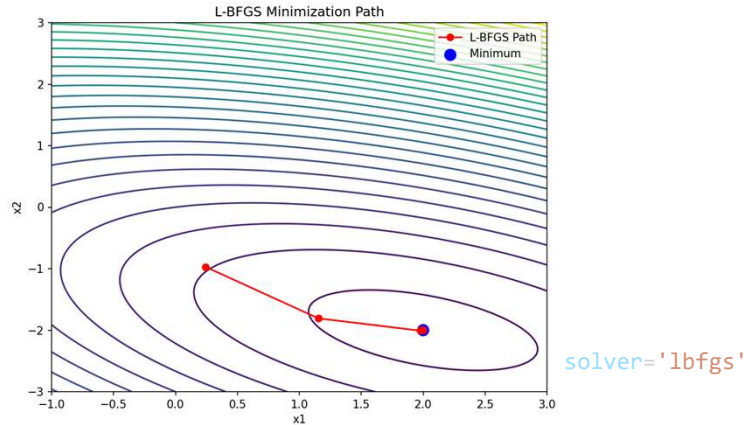
روش عددی شبه نیوتنی BFGS برای داده های حجیم مناسب است. در این روش ماتریس مشتقات مرتبه دوم (هسیان) تخمین زده می شود و دقیق حساب نمی شود ممکن است تعداد سعی و خطا زیاد شود ولی زمان برای داده های حجیم کاهش می یابد.

Soft Computing

32

روشهای حداقل کردن تابع هزینه

روش عددی شبه نیوتنی (Limited-memory BFGS) L- BFGS



روش عددی شبه نیوتنی L-BFGS مشابه روش BFGS بوده و برای داده های کم و متوسط مناسب است. چون در این روش از حافظه کمتری استفاده می شود برای داده های با ابعاد زیاد نظیر طبقه بندی چند کلاسی مناسب است.

Soft Computing

33

روشهای حداقل کردن تابع هزینه

گرادیان کاهشی (Gradient Descent)

گرادیان کاهشی دسته ای (BGD: Batch Gradient Descent)

گرادیان کاهشی تصادفی (SGD: Stochastic Gradient Descent)

مختصات کاهشی (Coordinate Descent)

گرادیان مزدوج (CG: Conjugate Gradient)

روش نیوتن

روش شبه نیوتنی (Broyden-Fletcher-Goldfarb-Shanno) BFGS

روش شبه نیوتنی (Limited-memory BFGS) L- BFGS

- روش های دیگری نیز وجود دارد که در اینجا توضیح داده نمی شود و می توانید مطالعه نموده و از آنها استفاده کنید مانند:

Mini-Batch Gradient Descent, Momentum-Based Methods, Adam (Adaptive Moment Estimation), Trust Region Methods, Trust Region Methods, Coordinate Descent

- در کتابخانه sklearn برای رگرسیونهای مختلف روشهای متفاوتی از حداقل کردن تابع هزینه گنجانده شده است. برای استفاده بهینه می توانید به توضیحات مربوطه مراجعه نمایید.

Soft Computing

34

نمونه رگرسیون لجستیک

مثال: برای داده های زیر که احتمال حمله قلبی را بر اساس ضربان قلب نشان می دهد رگرسیون لجستیک را بدست آورید
احتمال حمله قلبی شخصی با ضربان قلب ۱۰۰ چقدر است

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LogisticRegression

# Create the dataset
data = {
    'Heart Rate': [50, 50, 50, 50, 70, 70, 90, 90, 90, 90, 90],
    'Heart Attack': [1, 0, 0, 0, 0, 1, 1, 1, 0, 1, 1]
}
df = pd.DataFrame(data)

# Prepare the data
X = df[['Heart Rate']]
y = df['Heart Attack']
```

Heart Rate	Heart Attack
50	Y
50	N
50	N
50	N
70	N
70	Y
90	Y
90	Y
90	N
90	Y
90	Y

فایل: 7 regression logistic.py

Soft Computing

35

نمونه رگرسیون لجستیک

```
# Train the logistic regression model
model = LogisticRegression(solver='lbfgs') # Optimization method L-BFGS
model.fit(X, y)

# Generate values for prediction
x_values = np.linspace(40, 100, 200).reshape(-1, 1)
y_pred = model.predict_proba(x_values)[:, 1]
prediction_100 = model.predict_proba(np.array([[100]]))[0, 1]
print(f"heart attack probability for heart rate 100: {prediction_100:.2f}")

# Retrieve the coefficients for the logistic function formula
coef = model.coef_[0][0] # Teta1
intercept = model.intercept_[0] # Teta0

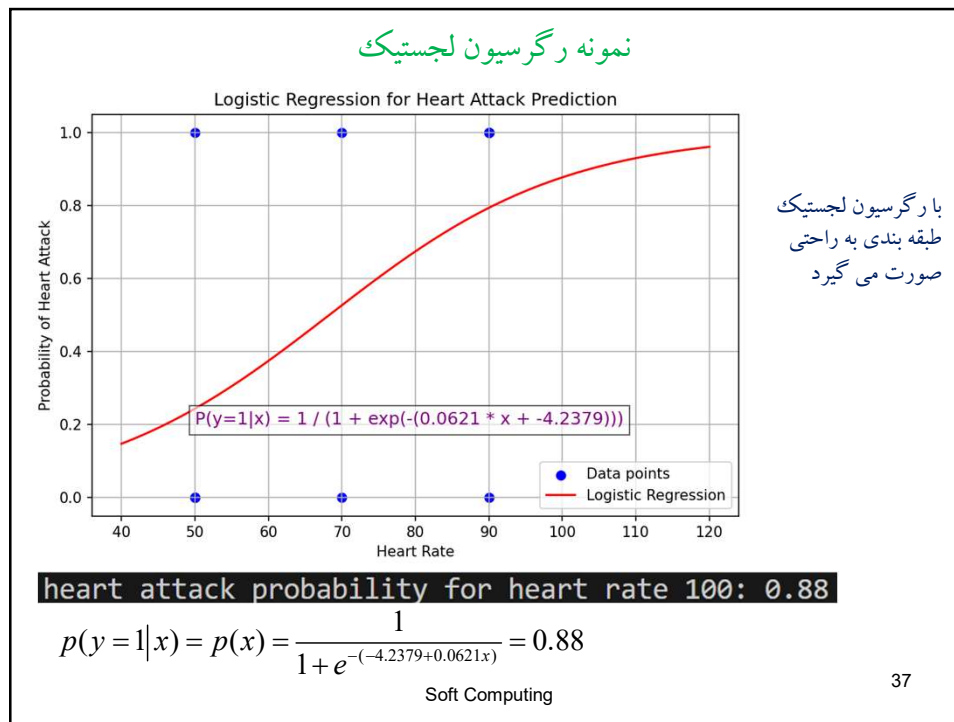
# Plotting
plt.figure(figsize=(8, 5))
plt.scatter(X, y, color='blue', label='Data points')
plt.plot(x_values, y_pred, color='red', label='Logistic Regression')
plt.xlabel('Heart Rate')
plt.ylabel('Probability of Heart Attack')
plt.title('Logistic Regression for Heart Attack Prediction')
plt.legend()

# Display formula on the plot
formula_text = f'P(y=1|x) = 1 / (1 + exp(-({coef:.4f} * x + {intercept:.4f})))'
plt.text(50, 0.2, formula_text, fontsize=12, color='purple',
        bbox=dict(facecolor='white', alpha=0.6))

plt.grid(True)
plt.show()
```

Soft Computing

36



تمرین برنامه نویسی

تمرین هفتم:

یک برنامه به زبان پایتون بنویسید که یک فایل داده را خوانده و رگرسیون لجستیک برای یک ویژگی را بدست آورد.

- ۱- منحنی رگرسیون را روی داده ها رسم کنید
- ۲- از سه روش مختلف حداقل نمودن تابع هزینه استفاده کرده و زمان آنها را با هم مقایسه کنید.
- ۳- برای یک حالت با ویژگی مشخص، احتمال را تعیین کنید

Soft Computing

38

ماتریس مشتقات

ماتریس مشتق مرتبه اول (برای آنهایی که بیشتر می خواهند بدانند)

Jacobian Matrix:

Suppose we have a

vector-valued function $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^m, \mathbf{f}(x_1, x_2, \dots, x_n) =$

$$\begin{bmatrix} f_1(x_1, x_2, \dots, x_n) \\ f_2(x_1, x_2, \dots, x_n) \\ \vdots \\ f_m(x_1, x_2, \dots, x_n) \end{bmatrix}$$

$$\text{Jacobian Matrix: } J(\mathbf{f}) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \frac{\partial f_m}{\partial x_2} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}$$

Example

Consider a simple function

$$\mathbf{f}(x, y) = \begin{bmatrix} f_1(x, y) \\ f_2(x, y) \end{bmatrix} \quad \begin{aligned} f_1(x, y) &= x^2 + y^2 \\ f_2(x, y) &= xy \end{aligned}$$

Jacobian Matrix:

$$J(\mathbf{f}) = \begin{bmatrix} 2x & 2y \\ y & x \end{bmatrix}$$

ماتریس مشتقات (برای آنهایی که بیشتر می خواهند بدانند)

ماتریس مشتق مرتبه دوم (برای آنهایی که بیشتر می خواهند بدانند)

Hessian Matrix:

Suppose we have a scalar-valued function $f(x_1, x_2, \dots, x_n)$

$$\text{Hessian Matrix: } H(f) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}$$

Example

Consider a simple function $f(x, y) = x^2 + y^2$

$$\text{Hessian Matrix: } H(f) = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$