Aula07

DATA SCIENCE IPT

TURMA 02

LINKED EDUCATION

V2019

CLASSIFICAÇÃO

Conteúdo

Na regressão, fizemos a predição de um valor contínuo, no nosso exemplo, o consumo de um carro em MPG. Agora, trabalharemos com classificação, ou seja, tentaremos prever a classe de uma amostra.

CLASSIFICAÇÃO

Na **Regressão**, obtivemos um modelo que se aproximava ao máximo das amostras (minimizava o erro por nós definido) para prever o consumo do carro em função de potência e peso. O consumo previsto é um valor contínuo.

Na CLASSIFICAÇÃO, o modelo prevê a classe de uma entrada. Exemplos :

a)Classificação Binária (é ou não é)

✓ Em função dos sintomas, classificar em Paciente com Dengue ou não (Dengue=1, não = 0)

✓ Ao fazer o login no celular, classificar se o rosto é ou não de quem está fazendo o login.

b)Multiclassificação (pertence a uma de n classes ou a nenhuma delas)

✓ Classificar uma foto em carro, cão ou casa (nesse caso o input é controlado : só carro, cão ou casa).

✓ Classificar a raça do cachorro em função da foto e um conjunto de raças. O resultado será uma das classes ou nenhuma delas.

Multiclassificação

Como construir multiclassificadores a partir de classificadores binários?

Estratégia OVA (one vs. All)

São criados n classificadores, um para cada classe, sendo a classe o "positivo" (one) e qualquer outra classe, o negativo (all).

Exemplo, para as classes A,B e C, teríamos os classificadores:

1)A x B,C
2)B x A,C
3)C x A,B

Submetendo uma amostra a cada um desses classificadores, o maior valor obtido, indicará a classe. Por exemplo, obtendo 0.3,0.7 e 0.5 em 1,2 e 3, a classe resultado será **B.**

Multiclassificação

Estratégia OVO (one vs. one)

São criados n*(n-1)/2 classificadores, um para cada par de classes. Exemplo, para as classes A,B e C, teríamos os classificadores:

1) A x B
2) A x C
3) B x C

Submetendo uma amostra a cada um desses classificadores, a classe com mais vitórias, será a escolhida. Havendo empate, a soma dos valores das predições fará a definição da classe resultado.

Regressão logística

- Para classificações binárias, usaremos inicialmente o algoritmo regressão logística.

Regressão logística

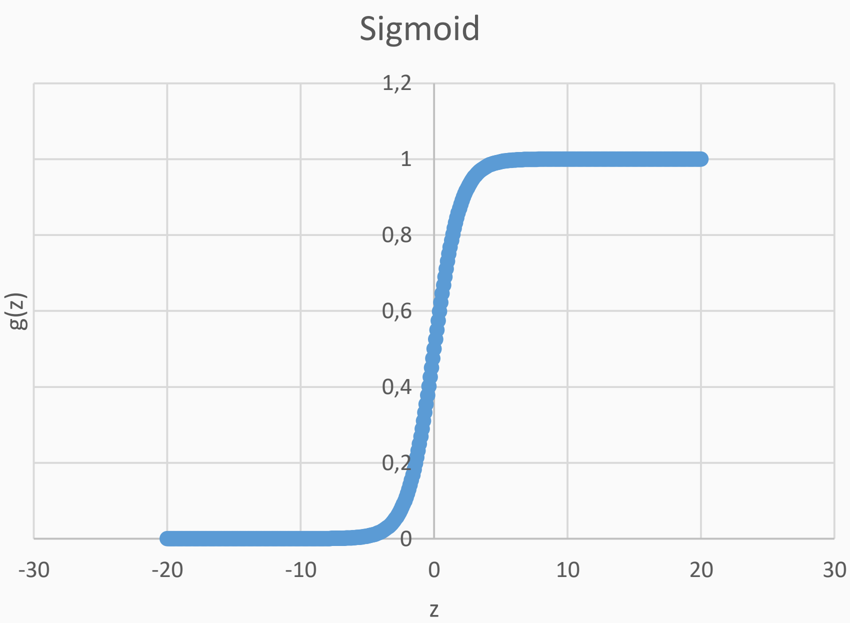
Uma das características da hipótese da regressão logística é utilizar a função sigmoid (também chamada função logística), essa função "empurra" fortemente os valores acima de zero para 1 e os valores abaixo de zero para zero. Em zero, a função tem valor ½.

A função sigmoid é dada por:

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

Sua derivada é
sig'(x)=s(x)(1-s(x))

Vamos traçar o gráfico dessa função em Python.



Regressão logística

Conteúdo

A hipótese (modelo) na função logística é h(x)= g(z) = g($\boldsymbol{\theta}^t$ **x**) =1/(1+$e^{-\boldsymbol{\theta}^t \mathbf{x}}$)...**x** e $\boldsymbol{\theta}$ são vetores

Assim, como na regressão linear, o modelo é definido pelos θs.
g(z)=1/(1+$e^{-\mathbf{z}}$) ...g(z) é a função sigmoid

Exemplo: para 2 features, teremos 3 θs (θ_0, θ_1 e θ_2), x será (1,x1,x2...1 é o x_0) e h(x) será:

$h(x)=1/(1+e^{-(\theta_0 1+ \theta_1 x_1 + \theta_2 x_2)})$

Na regressão logística assumimos que :

P(y=1 | θ ,x)= h(x)
P(y=0 | θ, x)= 1 - h(x)

Regressão logística

Conteúdo

Função Custo J para regressão logística

A não linearidade da função sigmoid não permite que o custo seja dado diretamente da função h(x) como na regressão linear :

J = 1/2m $\sum(\theta^t x - y)^2$ nesse caso a função é convexa (?)

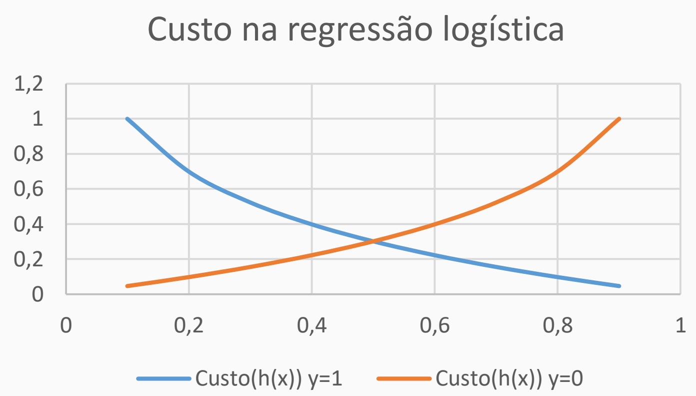
Mas no caso de regressão logística :

J = 1/m $\sum(g(\theta^t x) - y)2$ ... A função custo não é convexa, pode chegar a mínimos locais.

Assim, para o caso da regressão logística, definimos o custo como :

J = 1/m $\sum Custo(h(x), y)$

Custo(h(x,y))= -log(h(x))   se y=1
Custo(h(x,y))= -log(1-h(x))   se y=0



Regressão logística

Conteúdo

Se x= sigmoid()..então x ficará entre 0 e 1

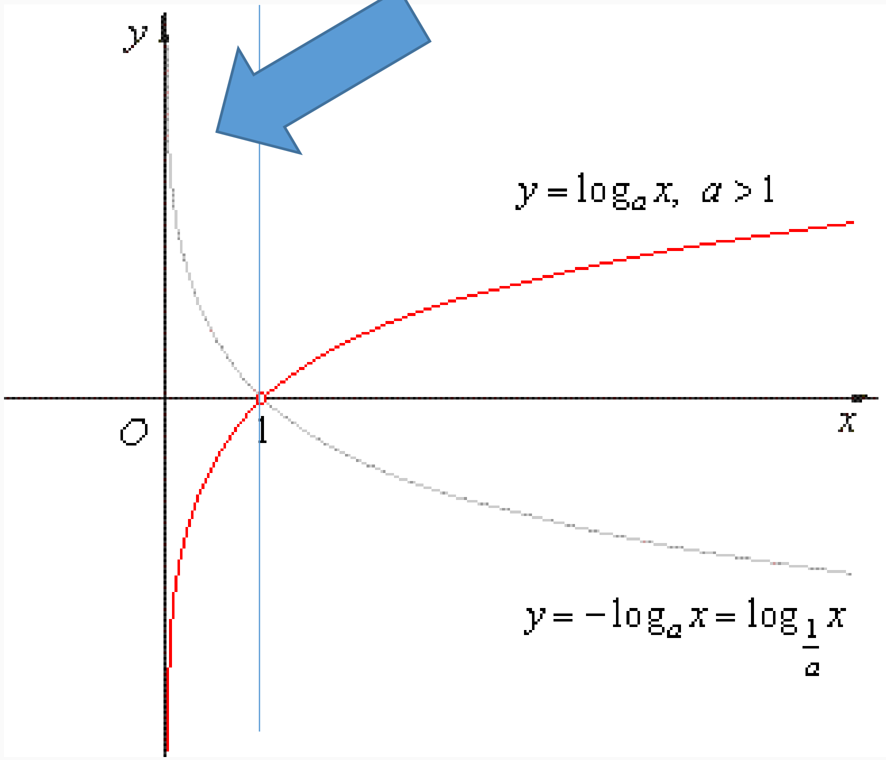
Um pouco da função $\log_a x$

$\log_{10} 100 = 2$, porque $10^2 = 100$

Algumas propriedades :

$\log_a bc = \log_a b + \log_a c$

$-\log_a x = \log_{1/a} x$



Regressão logística

Ainda .... a Função Custo J para regressão logística

Podemos escrever uma única função custo para y=0 ou y=1

J = $1/m\sum(\ y(\ -\log(h(x)))\ + (1\text{-}y)\ (-\log(1\text{-}h(x))\ )\ \ )$

J=$-1/m\sum(\ y(\log\big(h(x)\big) + (1-y)(\log\big(1-h(x)\big)))$

J= $-1/m\sum_{i=1}^{m}(yi(\log\big(h(x^i)\big) + (1-yi)\log\big(1-h(x^i)\big))$

Matricialmente..fazendo toda a somatória com produto interno de vetores:

$$J = 1/m(-\mathbf{y}^t\log(\mathbf{h})-(\mathbf{1}\text{-}\mathbf{y})^t\log(\mathbf{1}\text{-}\mathbf{h}))$$

Esse custo é chamado cross-entropy

h(x)= g(**X**θ)....ou seja, sigmoid aplicada a todo o vetor ye....o 1 em bold é um vetor de uns...

Regressão logística

Conteúdo

Para minimizarmos a função custo com o Gradient descent, temos que obter o gradiente dessa função, ou seja, as derivadas parciais da função custo em relação aos thetas.

O gradiente $\nabla J = -1/m \sum \left(y^i - h(x^i) \right) x_j^i$

A versão matricial é $\nabla J = 1/m.\mathbf{X}^t.(g(\mathbf{X}.\theta)-\mathbf{y})$

Obs: O Gradiente foi obtido pelas derivadas parciais em relação a θ. (veja último slide)

Regressão logística

Atividade 1 (Classificação Binária) :

Classificação de documentos bancários (1=fraude, 0=normal)

Base de dados da UCI :

https://archive.ics.uci.edu/ml/datasets/banknote+authentication#

As features vêm de parâmetros da imagem transformada (wavelets transforms(?))

Partir do notebook: reg_logistic.ipynb

Acurácia é a métrica : número de amostras classificadas corretamente/número total de amostras..usar Threshold=0.5 (output>0.5=> 1....caso contrário..0)

Regressão logística

Atividade 2 (Classificação Binária) :

Efetuar Classificação (mesmo exemplo anterior) por Regressão Logística usando uma biblioteca Python (scikit)

Tarefa1: Obter o índice de acertos

Partir do notebook reg_logistic_2.ipynb

Observar que matriz X no scikit não usa a coluna de 1´s na primeira coluna (faz automaticamente)

Regressão logística

Classificaremos uma flor (em função de suas dimensões) em 3 classes :

Íris-setosa
Iris-versicolor
Iris-virginica

Criaremos 3 classificadores binários :

1) **Íris-setosa ou não**
2) **Iris-versicolor ou não**
3) **Íris-virgínica ou não**

E analisaremos a acurácia de cada um deles (threshold 0.5 é padrão no scikit)

Partir de log_reg_iris.ipynb

Regressão logística

Faça uma cópia de log_reg_iris.ipynb e crie ova.ipynb. Crie a estratégia de multiclassificação OVA.

Regressão logística

Analisar o notebook : nova_doenca_res.ipynb

Explicar por que a acurácia com regressão logística é bem maior com o algoritmo regressão logística no caso do dataset nova_doenca.csv do que no dataset nova_doenca2.csv.

Cross-Validation

Uma técnica usual de validação á a cross “Cross-Validation”. A ideia é treinar o modelo em diversas partições da amostra e testá-lo nas amostras de teste de cada partição. Com os resultados das diversas partições, é possível compará-lo a outros modelos.

K-fold Cross-Validation

K-fold cross validation é a divisão do conjunto de amostras em “k” partes, com treinamento nas “k-1” e teste em 1 delas. Esse processo é repetido “k” vezes. Veja o exemplo, com k=10 (valor usual).

K-fold Cross-Validation

Partindo do notebook valida.ipynb,
Obtenha acurácia média em cross-validation k-fold com k=10

Gradiente do custo

$$
\begin{aligned}
\frac{\partial J(\theta)}{\partial \theta_j} &= \frac{\partial}{\partial \theta_j} \frac{-1}{m} \sum_{i=1}^{m} \left[y^{(i)} \left(\log(h_\theta\left(x^{(i)}\right)) \right) + (1 - y^{(i)}) \left(\log(1 - h_\theta\left(x^{(i)}\right)) \right) \right] \\
&\underset{\text{linearity}}{=} \frac{-1}{m} \sum_{i=1}^{m} \left[y^{(i)} \frac{\partial}{\partial \theta_j} \log\left(h_\theta\left(x^{(i)}\right)\right) + (1 - y^{(i)}) \frac{\partial}{\partial \theta_j} \left(\log(1 - h_\theta\left(x^{(i)}\right)) \right) \right] \\
&\underset{\text{chain rule}}{=} \frac{-1}{m} \sum_{i=1}^{m} \left[y^{(i)} \frac{\frac{\partial}{\partial \theta_j} (h_\theta\left(x^{(i)}\right)}{h_\theta\left(x^{(i)}\right)} + (1 - y^{(i)}) \frac{\frac{\partial}{\partial \theta_j} (1 - h_\theta\left(x^{(i)}\right))}{1 - h_\theta\left(x^{(i)}\right)} \right] \\
&\underset{h_\theta(x) = \sigma(\theta^\top x)}{=} \frac{-1}{m} \sum_{i=1}^{m} \left[y^{(i)} \frac{\frac{\partial}{\partial \theta_j} \sigma\left(\theta^\top x^{(i)}\right)}{h_\theta\left(x^{(i)}\right)} + (1 - y^{(i)}) \frac{\frac{\partial}{\partial \theta_j} (1 - \sigma\left(\theta^\top x^{(i)}\right))}{1 - h_\theta\left(x^{(i)}\right)} \right] \\
&\underset{\sigma'}{=} \frac{-1}{m} \sum_{i=1}^{m} \left[y^{(i)} \frac{\sigma\left(\theta^\top x^{(i)}\right)\left(1 - \sigma\left(\theta^\top x^{(i)}\right)\right) \frac{\partial}{\partial \theta_j} \left(\theta^\top x^{(i)}\right)}{h_\theta\left(x^{(i)}\right)} \right. \\
&\qquad \left. - (1 - y^{(i)}) \frac{\sigma\left(\theta^\top x^{(i)}\right)\left(1 - \sigma\left(\theta^\top x^{(i)}\right)\right) \frac{\partial}{\partial \theta_j} \left(\theta^\top x^{(i)}\right)}{1 - h_\theta\left(x^{(i)}\right)} \right] \\
&\underset{\sigma(\theta^\top x) = h_\theta(x)}{=} \frac{-1}{m} \sum_{i=1}^{m} \left[y^{(i)} \frac{h_\theta\left(x^{(i)}\right)\left(1 - h_\theta\left(x^{(i)}\right)\right) \frac{\partial}{\partial \theta_j} \left(\theta^\top x^{(i)}\right)}{h_\theta\left(x^{(i)}\right)} \right. \\
&\qquad \left. - (1 - y^{(i)}) \frac{h_\theta\left(x^{(i)}\right)\left(1 - h_\theta\left(x^{(i)}\right)\right) \frac{\partial}{\partial \theta_j} \left(\theta^\top x^{(i)}\right)}{1 - h_\theta\left(x^{(i)}\right)} \right] \\
&\underset{\frac{\partial}{\partial \theta_j}\left(\theta^\top x^{(i)}\right) = x_j^{(i)}}{=} \frac{-1}{m} \sum_{i=1}^{m} \left[y^{(i)} \left(1 - h_\theta\left(x^{(i)}\right)\right) x_j^{(i)} - (1 - y^{i})\, h_\theta\left(x^{(i)}\right) x_j^{(i)} \right] \\
&\underset{\text{distribute}}{=} \frac{-1}{m} \sum_{i=1}^{m} \left[y^{i} - y^{i} h_\theta\left(x^{(i)}\right) - h_\theta\left(x^{(i)}\right) + y^{(i)} h_\theta\left(x^{(i)}\right) \right] x_j^{(i)} \\
&\underset{\text{cancel}}{=} \frac{-1}{m} \sum_{i=1}^{m} \left[y^{(i)} - h_\theta\left(x^{(i)}\right) \right] x_j^{(i)} \\
&= \frac{1}{m} \sum_{i=1}^{m} \left[h_\theta\left(x^{(i)}\right) - y^{(i)} \right] x_j^{(i)}
\end{aligned}
$$

Fonte: Andrew NG

LINKED EDUCATION

Cursos com Alta Performance de Aprendizado