

P2 Regression - Case Study

#deepLearn

#Regression

Regression(数值预测) 可以做什么

- 股票预测系统 $f(\text{关于这支股票的各种资料}) = \text{明天的股价}$
- 自动驾驶汽车 $f(\text{汽车周围的环境}) = \text{接下来的驾驶操作}$
- 推荐算法 $f(\text{你的各种数据}) = \text{你最可能购买的物品}$
- 预测宝可梦的CP值（大雾）
CP值 为一只宝可梦的战斗力
就可以决定是否要进化这支宝可梦（大雾）

任务： $f(\text{一只宝可梦的信息}) = \text{他进化的CP值}$

分类：Regression Problem

输入：一只宝可梦所有的信息

输出：这只宝可梦进化后的战斗力

步骤列表

- 找一个model = function set 模型 = 函数集
- 定一个function set 里面的 function
- 找一个最好的function

Step 1: Model

找一个Model 或者也可以称为 Function set

随便写一个 Model

注意： a_b 表示 b 是 a 的下标

$$Y = b + w * x_{cp}$$

Y : 进化后的CP值

b : 常数项

w: 常数项

X_cp: 进化前宝可梦的CP值

其中 w, b 是参数

(可以为任何数)

这个Model (function set) 里面有无穷无尽个函数 (因为 w 和 b 可以随便取值)

形如 $Y = b + w * x$ 这样的模型 (model) 叫做线性模型 (linear model)

#linear

$$Y = b + \sum w_i * x_i$$

x_i: 称为 feature (特征)

w_i: weight (权重)

b: bias(偏差)

Step 2: Goodness of Function

类型: Supervise Learning Test

我们用 a^b 表示 b 是 a 的上标

Function input: $x^1 \quad X^2 \dots$

(注: 用上标来表示一个完整的Component (元件))

Function output (Value): $Y^1H \quad Y^2H \dots$

(注: 在一个完整的Component 后面 跟一个大写的H表示他是一个输出)

(注: 这不是一个推荐的写法, 只不过是因在markdown方便)

注意: 在本次试验中, 我们都是输入输出值都是单个数值, 所以可能不需要上标下标, 但是在后面的试验中, 这是一个推荐的方法

Training Data:

10 Pokemons

x^1, y^1H

x^2, y^2

...

This is the real data(老师注：这是真实的数据) (大雾)

Goodness of Function (定义一个函数的好坏)

如何去定义：定一个损失函数(Loss function) 记为 Loss 或者 L (大写)

Input : a function (输入是一个函数)

Output: how bad it is (这个函数怎么样)

$$L(f) = L(w, b)$$

衡量一个函数的好坏 等价于 衡量一组参数的好坏

如何去求这个Loss (常见求法：写均方差，如下图所示)

$$= \sum_{n=1}^{10} \left(\hat{y}^n - (b + w \cdot x_{cp}^n) \right)^2$$

y^H : 这是理想值 (真正的数值)

里面的小括号是：这个函数预测出来的值 (预测的数值)

两者求差就是 预测的预测

最后再把预测误差的求和

Step 3: Best Function (选出最好的函数)

Pick The Best Function !

$$f^* = \arg \min_f L(f)$$

要找到一个 f 使得 $L(f)$ 最小

$$w^*, b^* = \arg \min_{w, b} L(w, b)$$

穷举所有的 w 和 b 来找到最小的 $L(w, b)$

$$= \arg \min_{w, b} \sum_{n=1}^{10} \left(\hat{y}^n - (b + w \cdot x_{cp}^n) \right)^2$$

b ，可以給我們最好的結果。

找到所有 w b 带进去 损失函数 看看那个最小

Gradient Descent 梯度下降算法

#GradientDescent

该方法不只适用于解这么一个函数,这个函数只要是 L 损失函数是可以微分的，就可以求。

简化题目：

Consider loss function $L(w)$ with one parameter w ;

(只考虑有一个参数的损失函数)

暴力方法：穷举所有 w 可能的取值，全部带进去看那个最小。

Gradient Descent 如何解：

- (Randomly) Pick an initial value w^0 (随机取起始点 记为 w^0)

