线性回归(上)

李俊

问题

Pokemon Go

ID	图标	精灵名称	属性	最大CP	攻击	防御
001		妙蛙种子	草 毒	1071.54	126	126
002		妙蛙草	草 毒	1632.19	156	158
003		妙蛙花	草 毒	2580.49	198	200
004		小火龙	火	955.24	128	108
005		火恐龙	火	1557.48	160	140
006		喷火龙	火 飞行	2602.20	212	182

问题

假设我们有一个经调查得到的数据集，内容是关于Pokemon Go的进化前CP值与进化后CP值之间的关系表：

进化前CP值	进化后CP值
338	640
333	633
328	619
207	393
226	428
……	……

有了这样的数据，我们怎样才能预测CP值，比如推导出一个从进化前CP值得出进化后CP值的函数。

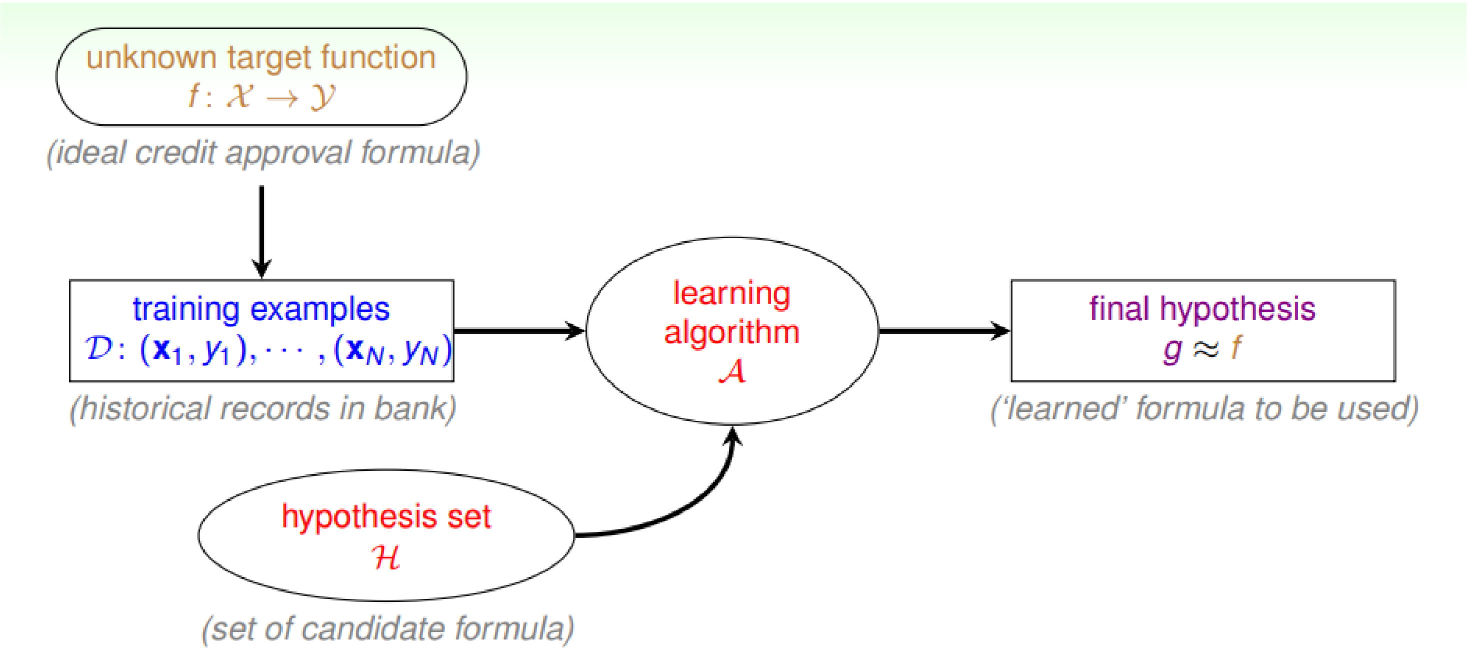
定义

为了今后的课程书写方便，我们约定一些未来将要用到的符号：

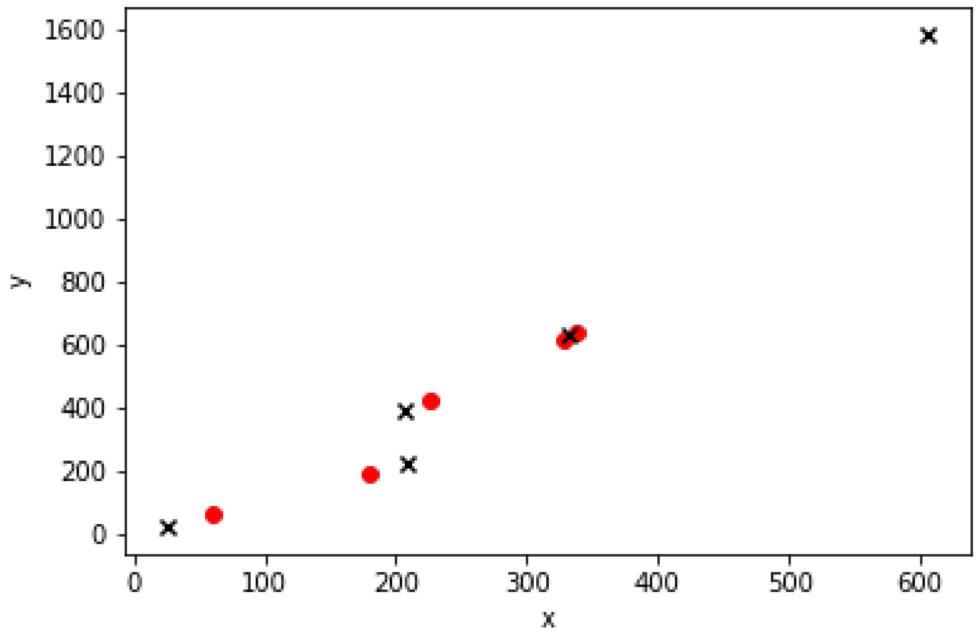
- 使用$x^{(\mathrm{i})}$来表示**输入变量**，也称作**输入特征**（feature），即输入数据；
- 使用$y^{(i)}$来表示**输入变量**或**目标变量**（target），也就是我们尝试做出预测的值；
- 一对$(x^{(i)},y^{(i)})$称作一个**训练样本**（training example）；
- 用于做学习的数据集，也就是由m个$(x^{(i)},y^{(i)})$组成的列表$\{(x^{(i)},y^{(i)});i=1,\ldots,m\}$，称作**训练集**（**training set**）。
- X表示**输入值空间**，Y表示**输出值空间**，在本例中$X=Y=\mathbb{R}$（即都是一维实向量空间）。

问题

更加正式的描述监督学习问题：我们的目标是，通过一个给定的训练集，训练一个函数$h: X \rightarrow Y$，如果$h(x)$能够通过较为准确的预测而得到结果y，则我们称这个$h(x)$是"好的"。因为一些历史原因，函数h被称为假设（hypothesis），监督学习用流程图表示如下：



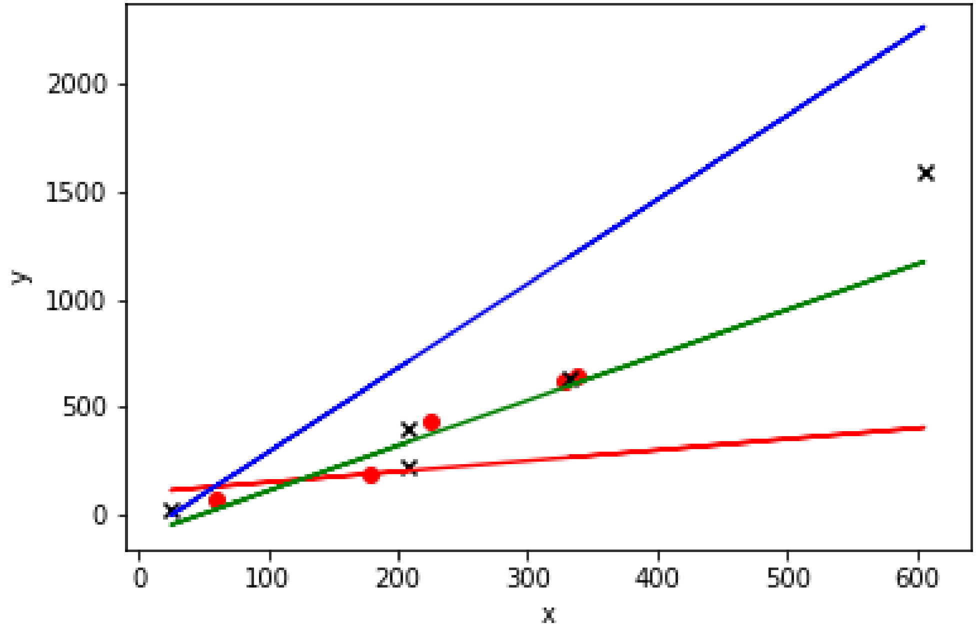
线性回归



我们有10组数据：$\{(x^1, y^1), \{(x^2, y^2)\}, \ldots\ldots, \{(x^{10}, y^{10})\}\}$

x^i表示输入变量，y^i表示输出变量。

线性回归



模型h(x)是x的线性回归函数：

$$y = wx + b$$

其中是$h(x)$预测值，w和b是参数，w是权重，b是偏差。

线性回归

损失函数（Loss Function）：

$$L(w,b)=\frac{1}{2}(h\left(x^{(1)}\right)-y^{(1)})^2$$

代价函数（Cost Function）：

$$C(w,b)=\frac{1}{2n}\sum_{i=1}^{n}(h\left(x^{(i)}\right)-y^{(i)})^2$$

线性回归

目标函数:

$$\min_{w,b} C(w,b) = \frac{1}{2n}\sum_{i=1}^{n}\left(h\left(x^{(i)}\right) - y^{(i)}\right)^2$$
$$= \frac{1}{2n}\sum_{i=1}^{n}\left(w * x^{(i)} + b - y^{(i)}\right)^2$$

线性回归

线性回归模型的一般形式：

$$
\begin{aligned}
h(X) &= \sum_{i=1}^{d} w_i x_i + b \\
&= \sum_{i=1}^{d} w_i x_i + \underbrace{b}_{w_0} * \underbrace{(+1)}_{x_0} \\
&= \sum_{i=0}^{d} w_i x_i \\
&= W^T X
\end{aligned}
$$

线性回归

损失函数的一般形式：

$$L(W)=\frac{1}{2}(h\underbrace{\left(X^{(i)}\right)}_{W^TX^{(i)}}-y^{(1)})^2$$

代价函数的一般形式:

$$C(W)=\frac{1}{2n}\sum_{i=1}^{n}(\underbrace{h\left(X^{(i)}\right)}_{W^TX^{(i)}}-y^{(i)})^2$$

线性回归

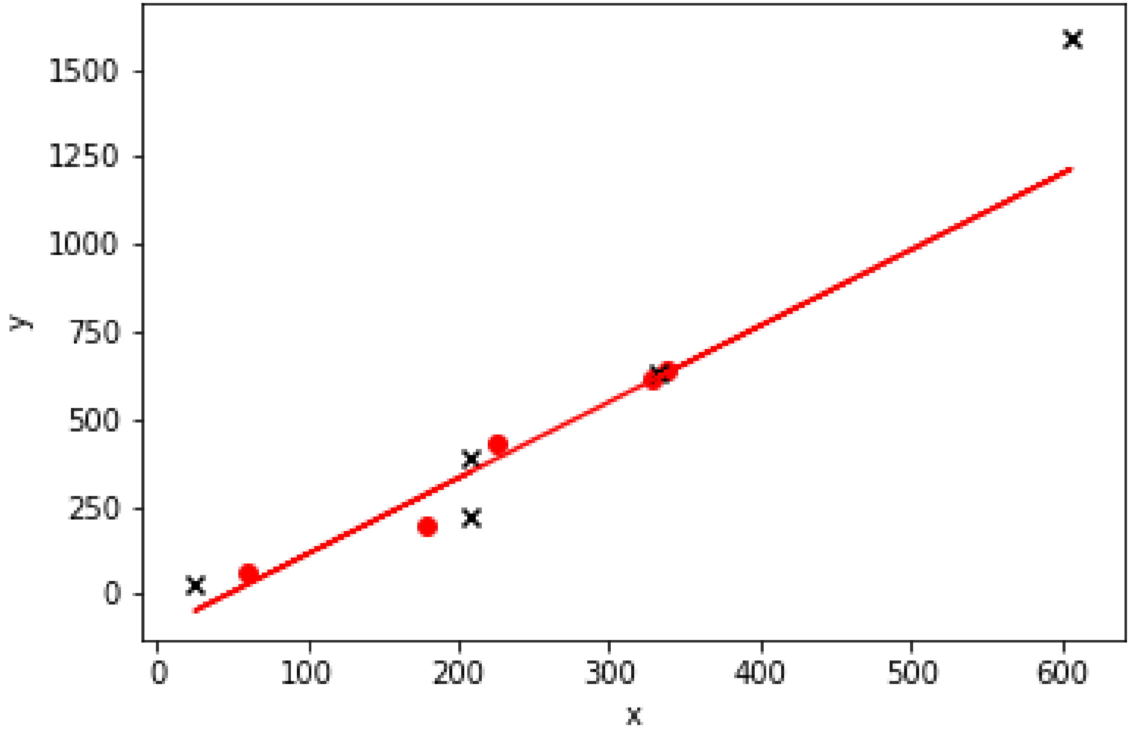
目标函数的一般函数：

$$W^* = \underset{W}{\operatorname{argmin}}\, C(\mathrm{W}) = \frac{1}{2n}\sum_{i=1}^{n}\left(h\left(X^{(i)}\right) - y^{(i)}\right)^2$$

$$= \frac{1}{2n}\sum_{i=1}^{n}\left(W^T X^{(i)} - y^{(i)}\right)^2$$

线性回归

$$y = \alpha_1 x + \alpha_0$$

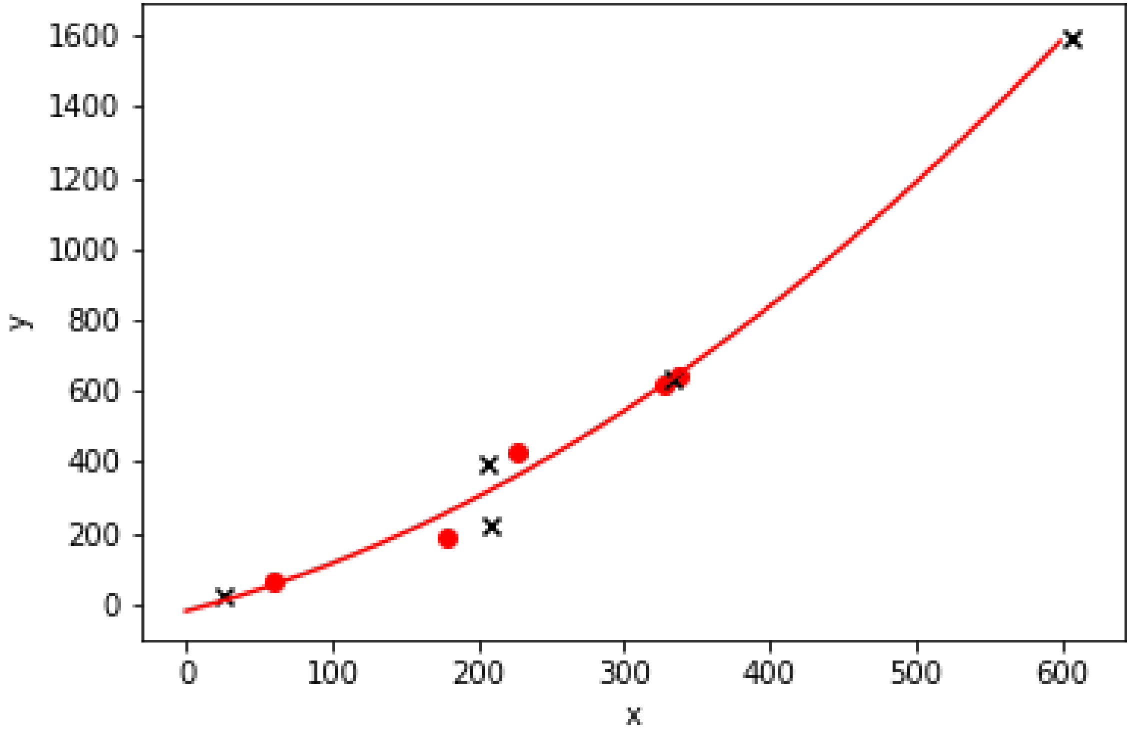


$$error_{train} = 1188.229$$
$$error_{test} = 16558.387$$

$$y = 2.17 * x - 101.718$$

线性回归

$$y = \alpha_1 x + \alpha_2 x^2 + \alpha_0$$

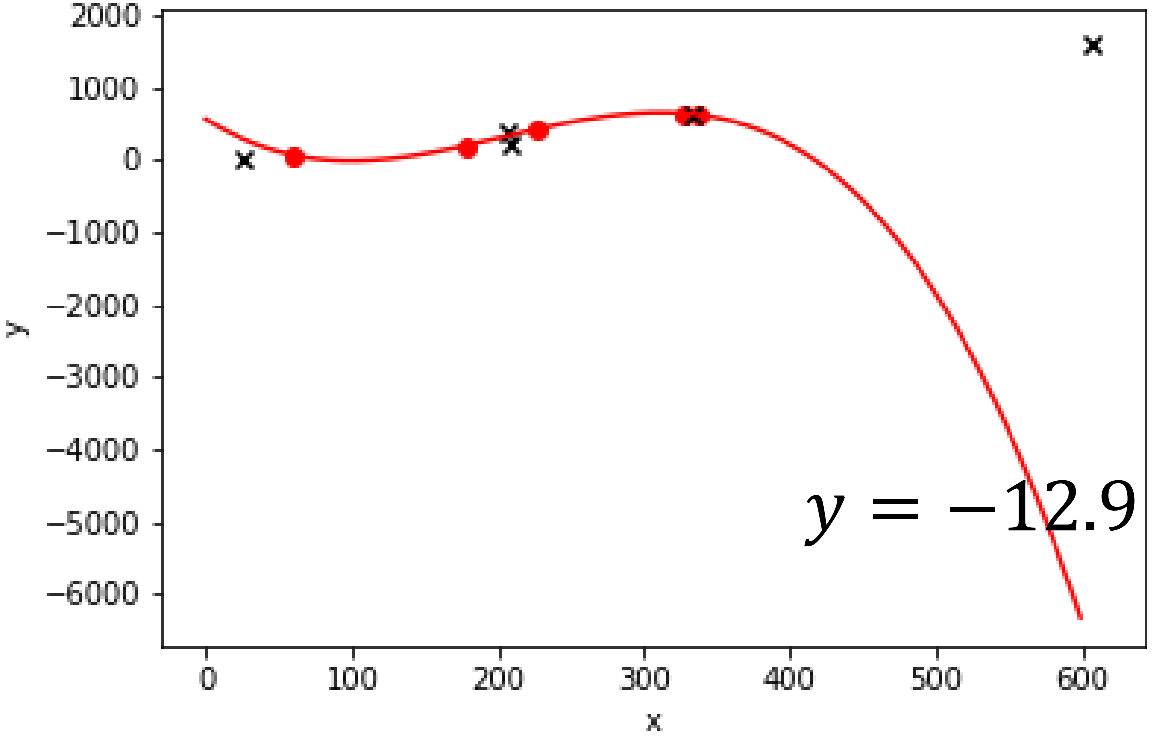


$$error_{train} = 914.23$$
$$error_{test} = 1540.624$$

$$y = 1.07 * x + 0.003 * x^2 - 18.526$$

线性回归

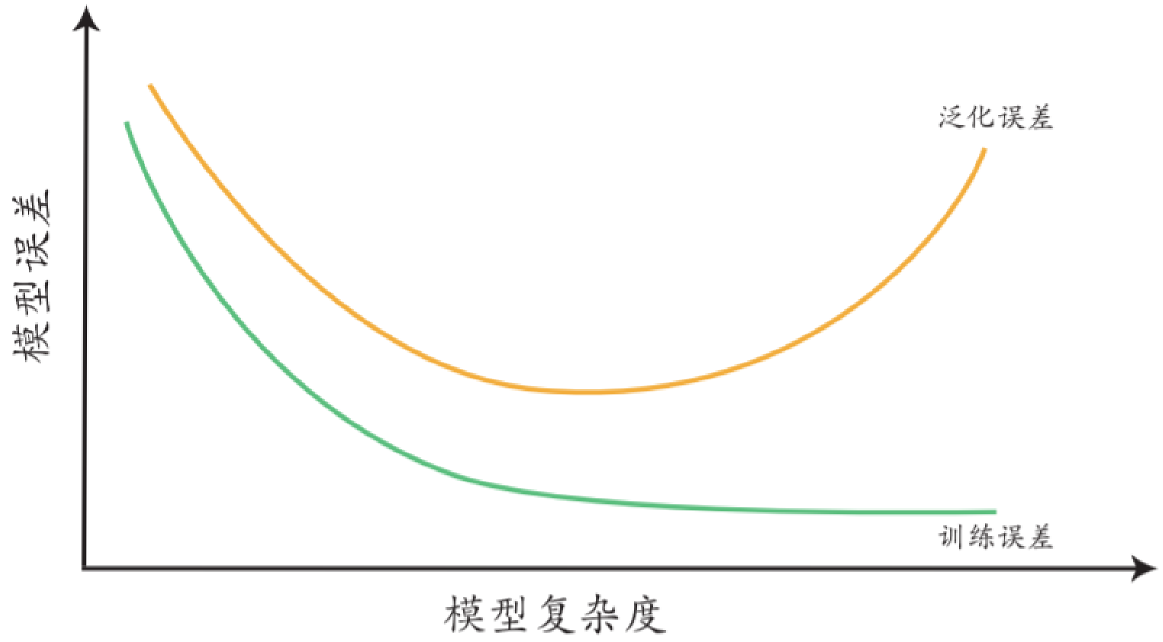
$$y = \alpha_1 x + \alpha_2 x^2 + \alpha_3 x^3 + \alpha_0$$



$$error_{train} = 105.147$$
$$error_{test} = 6955811.737$$

$$y = -12.9 * x + 0.085 * x^2 - 0.00014 * x^3 - 561.35$$

过度拟合问题



线性回归

我们比较一下三个函数：

$$y = 2.17 * x - 101.718$$
$$y = 1.07 * x + 0.003 * x^2 - 18.526$$
$$y = -12.9 * x + 0.085 * x^2 - 0.00014 * x^3 - 561.35$$

使用多项式回归，如果多项式最高次项比较大，模型就容易出现过拟合。

线性回归

正则化是一种常见的防止过拟合的方法，原理是在代价函数后面加上一个对参数的约束项（惩罚项），这个约束项被叫做**正则化项**。

在线性回归模型中，通常有两种不同的正则化项：

- 加上所有参数的绝对值之和，即$L1$范数，此时叫做$\boldsymbol{Lasso}$**回归**。
- 加上所有参数的平方和，即$L2$范数，此时叫做$\boldsymbol{Ridge}$**回归（岭回归）**。

线性回归

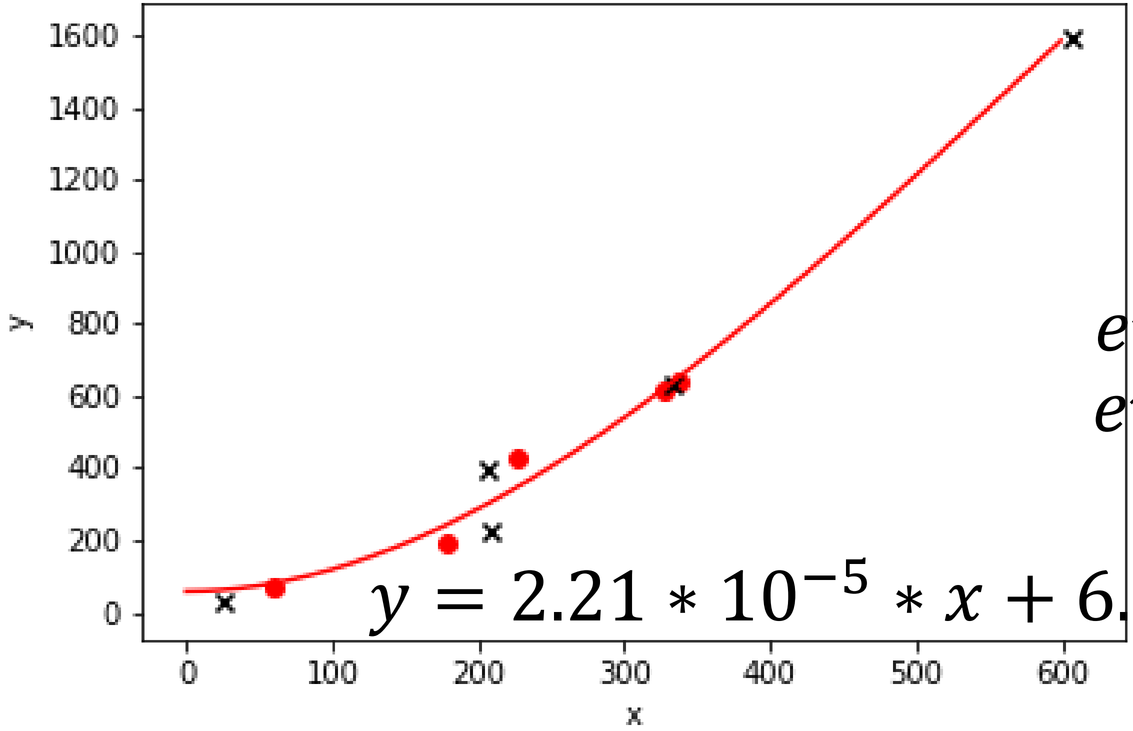
岭回归通过对系数的大小施加惩罚来解决普通最小二乘法的一些问题。 岭系数最小化的是带罚项的残差平方和：

$$W^* = \underset{W}{\operatorname{argmin}}\, C(\mathrm{W}) = \frac{1}{2n}\sum_{i=1}^{n}(W^T X^{(i)} - y^{(i)})^2 + \alpha\|W\|_2^2$$

其中，$\alpha \geq 0$是控制系数收缩量的复杂性参数：α的值越大，收缩量越大，这样系数对共线性的鲁棒性也更强。

线性回归

$$y = \alpha_1 x + \alpha_2 x^2 + \alpha_3 x^3 + \alpha_0$$



$$error_{train} = 983.406$$
$$error_{test} = 1613.133$$

$$y = 2.21 * 10^{-5} * x + 6.45 * 10^{-3} * x^2 - 3.65 * 10^{-6} * x^3 + 58.127$$

线性回归

$Lasso$回归的全称是“最小绝对缩减和选择算子”（Least Absolute Shrinkage and Selection Operator），由加拿大学者罗伯特·提布什拉尼于1996 年提出。与岭回归不同的是，LASSO 回归选择了待求解参数的一范数项作为惩罚项：

$$W^* = \underset{W}{\operatorname{argmin}}\, C(\mathrm{W}) = \frac{1}{2n}\sum_{i=1}^{n}(W^T X^{(i)} - y^{(i)})^2 + \lambda\|W\|_1$$

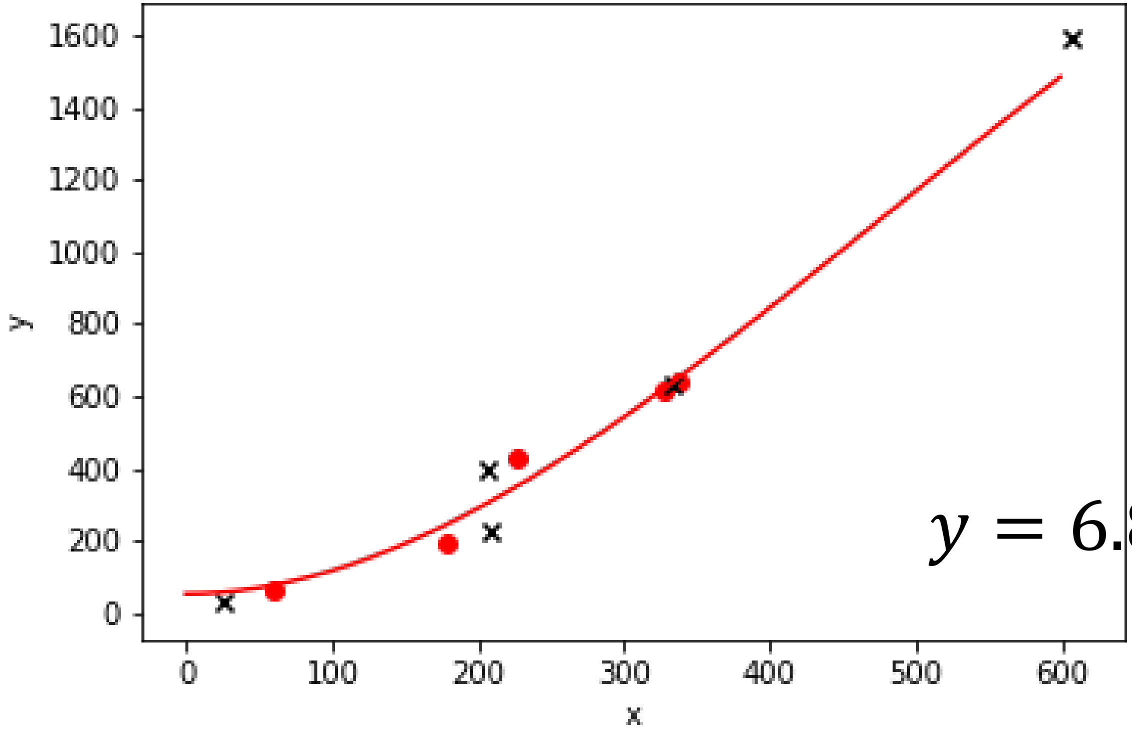
其中，λ是一个常数。

线性回归

与岭回归相比，$Lasso$回归的特点在于稀疏性的引入。它降低了最优解W的维度，也就是将一部分参数的贡献削弱为 0，这就使得W中元素的数目大大小于原始特征的数目。

线性回归

$$y = \alpha_1 x + \alpha_2 x^2 + \alpha_3 x^3 + \alpha_0$$



$$error_{train} = 936.153$$
$$error_{test} = 2269.068$$

$$y = 6.88 * 10^{-3} * x^2 - 4.83 * 10^{-6} * x^3 + 52.858$$

线性回归

但无论岭回归还是$Lasso$回归，其作用都是通过惩罚项的引入抑制过拟合现象，以训练误差的上升为代价，换取测试误差的下降。将以上两种方法的思想结合可以得到新的优化方法**弹性网络**（Elastic Net）。

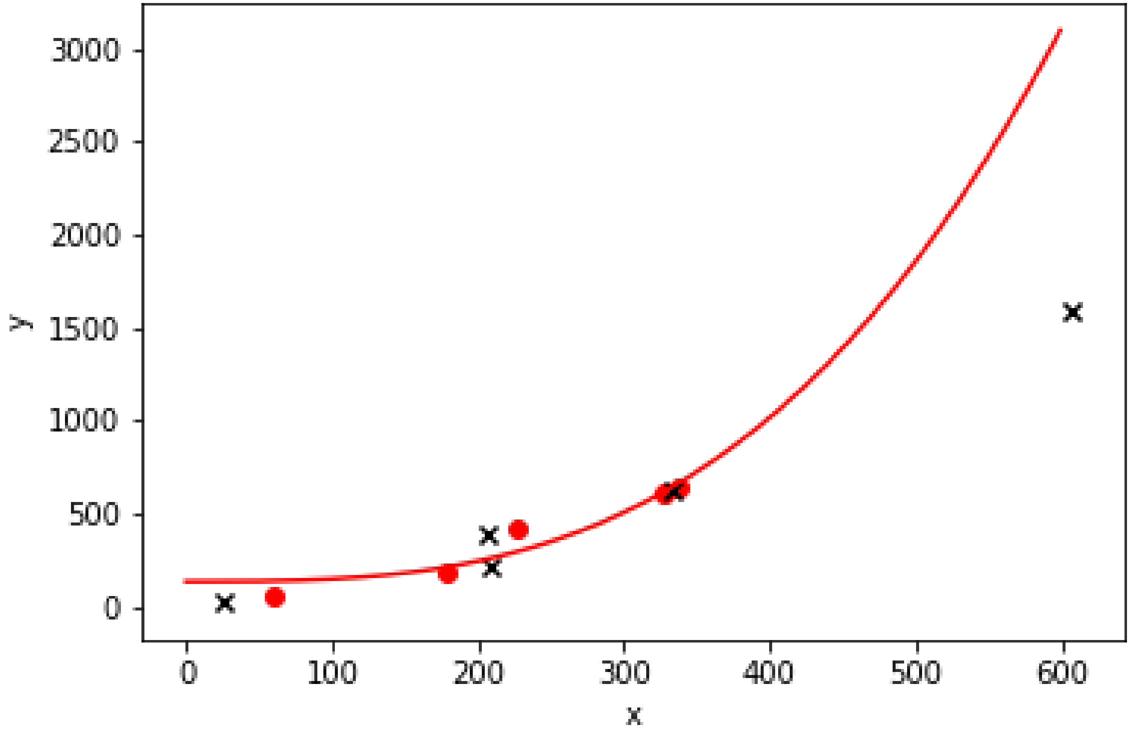
弹性网络 是一种使用$L1, L2$范数作为先验正则项训练的线性回归模型。 这种组合允许学习到一个只有少量参数是非零稀疏的模型，就像$Lasso$回归一样，但是它仍然保持一些像岭回归的正则性质:

$$W^* = \underset{W}{\operatorname{argmin}}\, C(\mathrm{W}) = \frac{1}{2n}\sum_{i=1}^{n}(W^T X^{(i)} - y^{(i)})^2 + \alpha r\|W\|_1 + \frac{\alpha(1-r)}{2}\|W\|_2^2$$

其中r表示$L1$所占的比例。

线性回归

$$y = \alpha_1 x + \alpha_2 x^2 + \alpha_3 x^3 + \alpha_0$$



$$error_{train} = 2427.446$$
$$error_{test} = 263483.786$$
$$y = 1.378 * 10^{-6} * x^3 + 52.858$$

总结

今天和分享了机器学习基本算法之一的线性回归的基本原理，其要点如下：

- 线性回归假设输出变量是若干输入变量的线性组合，并根据这一关系求解线性组合中的最优系数；
- 多元线性回归问题也可以用最小二乘法求解，但极易出现过拟合现象；
- 岭回归和 LASSO 回归分别通过引入二范数惩罚项和一范数惩罚项抑制过拟合。