机器学习实验 3. 线性回归

线性回归原理概述

回归分析特指对预测变量与响应变量之间的联系进行建模的过程。在机器学习中，狭义的回归分析仅指预测变量为连续值的情况，即某种感兴趣的属性或特征。但回归也可以用来进行分类，即将离散的类标签作为回归的预测变量。鉴于线性回归在回归中的基础作用，因此首先讲述线性回归。

线性回归学习一个线性模型以拟合预测变量与响应变量之间的联系，从而尽可能准确地预测实际输出。一般来说，线性回归主要应用于预测连续输出，如价格、经济指数、气象指标等，因此广泛应用于统计学领域。

一元线性回归

给定数据集 $D=\{(x_1,y_1),(x_2,y_2),...,(x_m,y_m)\}$，其中，$x_i=(x_{i1};x_{i2};...;x_{id}),y_i\in R$。考虑最简单的形式：输入属性的数目只有一个，即 $D=\{(x_i,y_i)\}_{i=1}^m$，其中 $x_i\in R$，这种形式一般称为即一元线性回归。此时，回归方程为：$f(x_i)=wx_i+b$，使得 $f(x_i)\approx y_i$。为了确定 w 和 b，使得 $f(x_i)$ 与 y_i 均方误差最小，即：

$$(w^*,b^*)=\arg\min_{(w,b)}\sum_{i=1}^m(f(x_i)-y_i)^2=\arg\min_{(w,b)}\sum_{i=1}^m(y_i-wx_i-b)^2\ ,$$

求解其最小值情况下的参数 w 和 b。均方误差有非常好的意义，他对应了常用的欧几里得距离（Euclidean distance）。基于均方误差最小化来进行模型求解的方法称为最小二乘法（least square method）。在线性回归中，最小二乘法就是试图找到一条直线，使得所有样本到直线上的欧几里得距离之和最小。求解 w 和 b 使 $E_{(w,b)}=\sum_{i=1}^m(y_i-wx_i-b)^2$ 最小化的过程，称为线性回归的最小二乘参数估计。因为 E 是关于 w 和 b 的凸函数，所以分别对 w 和 b 求偏导，令偏导数为 0，则有 w 和 b 的最优解的闭式（closed-form）解：

$$w=\frac{\sum_{i=1}^{m}y_i(x_i-\bar{x})}{\sum_{i=1}^{m}x_i^2-\frac{1}{m}(\sum_{i=1}^{m}x_i)^2},b=\frac{1}{m}\sum_{i=1}^{m}(y_i-wx_i)。$$

多元线性回归

假设想要使用两个特征（今天的股票价格和昨天的股票价格）来预测明天的股票价格。把今天的股票价格作为第一个特征f_1，昨天的股票价格作为特征f_2。接下来，线性回归的目标就是学习两个权重系数：w_1,w_2。这样就可以使用一个简单的方程来表示明天的股票价格：$\hat{y}=w_1f_1+w_2f_2$。其中$\hat{y}$是明天真实股票价格y的预测值。最终，线性回归的目标就是学习一组权重参数进而在预测时尽可能准确的接近真实值。根据上述思路，考虑线性回归有n个不同的响应变量，可以看作n个不同的输入特征、属性。则回归模型构造为n个响应变量线性组合与输出属性之间的线性关系。为简化符号，这里采用矩阵方式进行表示。输入m个样本，每个样本都具有n个特征，则输入样本表示为矩阵X，而回归参数表示为向量W:

$$X=\begin{pmatrix}x_{11}&x_{12}&\dots&x_{1n}&1\\x_{21}&x_{21}&\dots&x_{2n}&1\\\dots&\dots&\dots&\dots&1\\x_{m1}&x_{m2}&\dots&x_{mn}&1\end{pmatrix}_{m\times(n+1)}=\begin{pmatrix}\vec{x}_1&1\\\vec{x}_2&1\\\dots&\dots\\\vec{x}_m&1\end{pmatrix}_{m\times(n+1)},\quad W=\begin{pmatrix}w_1\\w_2\\\dots\\w_n\\b\end{pmatrix}_{(n+1)\times1}$$

线性回归的参数一般包括响应变量的系数w和常量（截距）b。为了矩阵计算的方便，在上述两个矩阵表达中，将b吸收到系数向量W中，并在数据矩阵X中相应扩展了一列，由此可将线性回归直接表示为数据矩阵X与系数向量相乘的形式:

$$XW=\begin{pmatrix}\vec{x}_1&1\\\vec{x}_2&1\\\dots&\dots\\\vec{x}_m&1\end{pmatrix}\bullet\begin{pmatrix}w_1\\w_2\\\dots\\w_n\\b\end{pmatrix}=\begin{pmatrix}\sum_{i=1}^{n}w_ix_{1,i}+b\\\sum_{i=1}^{n}w_ix_{2,i}+b\\\dots\\\sum_{i=1}^{n}w_ix_{n-1,i}+b\\\sum_{i=1}^{n}w_ix_{n,i}+b\end{pmatrix}_{m\times1}$$

令样本实际值为：

$$Y=\begin{pmatrix} y_1 \\ y_2 \\ \cdots \\ y_m \end{pmatrix}_{m\times 1}$$

则对每一个样本的预测损失为：

$$Y-XW=\begin{pmatrix} y_1 \\ y_2 \\ \cdots \\ y_m \end{pmatrix}_{m\times 1}-\begin{pmatrix} \sum_{i=1}^{n} w_i x_{1,i}+b \\ \sum_{i=1}^{n} w_i x_{2,i}+b \\ \cdots \\ \sum_{i=1}^{n} w_i x_{n-1,i}+b \\ \sum_{i=1}^{n} w_i x_{n,i}+b \end{pmatrix}_{m\times 1}=\begin{pmatrix} loss_1 \\ loss_2 \\ \cdots \\ loss_m \end{pmatrix}_{m\times 1}$$

利用矩阵的方式求解最小二乘法，则对于 m 个样本，预测模型的总损失的平方和为：

$$E_W=(Y-XW)^T(Y-XW)=\sum_{i=1}^{m} loss_i^2$$

求导数，$\frac{\partial E_W}{\partial W}=0$。可得 $X^TXW=X^TY$，根据伪逆的形式可得闭合解形式为：

$$W=(X^TX)^{-1}X^TY$$

所以最终多元线性模型所得的预测值为可以表示为：

$$f(\vec{x}_i)=\begin{pmatrix} x_{i1} & x_{i2} & \cdots & x_{in} & 1 \end{pmatrix}\times W=\vec{x}_i(X^TX)^{-1}X^TY$$

线性回归算法描述

1）首先通过闭合解法或梯度下降法寻找最优参数 w 和 b

2）利用评估指标来表征这组参数的好坏

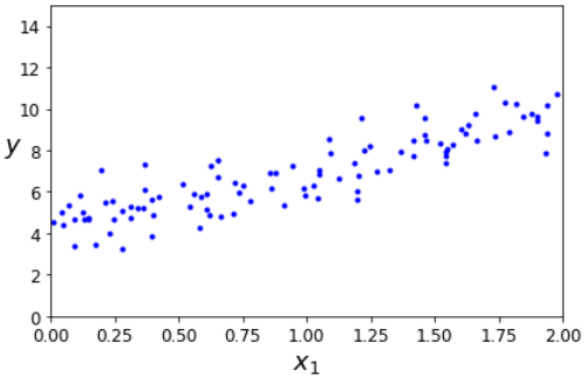
线性回归一般实现

1）生成一组随机实验数据

```
import numpy as np

X = 2 * np.random.rand(100, 1)
y = 4 + 3 * X + np.random.randn(100, 1)
```

得到如下图



2）求解 w，并对 w 做出预测

```
X_b = np.c_[np.ones((100, 1)), X]  # add x0 = 1 to each instance
theta_best = np.linalg.inv(X_b.T.dot(X_b)).dot(X_b.T).dot(y)
```

```
X_new = np.array([[0], [2]])
X_new_b = np.c_[np.ones((2, 1)), X_new]  # add x0 = 1 to each instance
y_predict = X_new_b.dot(theta_best)
y_predict
```

3）得到线性回归模型预测