第二章、线性回归

赵涵

2022 年 3 月 10 日

1 线性模型的建立

我们考虑一个问题，房价预测。对一个房子来说，它有很多特征，其中包括面积，地段，楼层，是否学区，是否有电梯等等，假设它有d个特征，我们用一个列向量来表示一个房子的所有特征，即$x = (x_1, x_2, ..., x_d)^T$，这里$x$代表房子，向量里面的$x_i$代表它的特征，所以$x$是一个$d$维的列向量，用$y$表示房子的价格。我们基于此，建立线性回归模型，即我们假设，存在一组参数，这种参数与房子的特征相乘，得到的结果就应该是房子价格。即：

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots \theta_d x_d \tag{1}$$

我们记$\theta = (\theta_0, \theta_1, ..., \theta_d)^T$ 为一个参数向量，这里θ_0是一个**偏置**（bias）项，它的作用是为了增加模型的灵活性，否则模型始终会通过坐标原点（在坐标系上看，很显然）。我们还可以认为存在一个$x_0 = 1$，这样就可以通过向量来表示公式（1），即：

$$h_\theta(x) = \sum_{i=0}^{d} \theta_i x_i = \theta^T x \tag{2}$$

等式右边表示的是两个向量做内积。以上就称为**线性回归**（Linear Regression）模型。对于一个模型，参数θ 可以有无穷多个取值，所以下一步，我们要找到一个策略，找到最优的那一组θ，最常用的方法是**最小均方估计**（Least Square Estimation）。我们让模型的预测值，与真实值比较，然后最小它们之间的差异。因为对一个样本集来说，不止一个数据，假设有n个数据，在x的右上角添加角标来标记第几个数据，即$(x^{(i)}, y^{(i)})$，i表示是第几个数据。那么就是这n个数据的误差的和，由于误差有正有负，所以我们采用平方来消除掉这种由于正负引起的影响。我们定义函数：

$$J(\theta) = \frac{1}{2} \sum_{i=1}^{n} (h_\theta(x^{(i)}) - y^{(i)})^2 \tag{3}$$

这个函数我们把它称为**损失函数**（loss function）。我们的目标就是最小化这个函数值，进而求出θ。至此我们建立了模型，制定了策略，那么接下来如何去实现目标，我们把这一步称为算法。最后我们总结一下，这几个有关机器学习的名词。

模型（model）：模型在未进行训练前，其可能的参数是多个甚至无穷的，故可能的模型也是多个甚至无穷的，这些模型构成的集合就是假设空间。

策略（strategy）：即从假设空间中挑选出参数最优的模型的准则。模型的分类或预测结果与实际情况的误差（损失函数）越小，模型就越好。那么策略就是误差最小。

算法（algorithm）：即从假设空间中挑选模型的方法（等同于求解最佳的模型参数）。机器学习的参数求解通常都会转化为最优化问题，故学习算法通常是最优化算法，例如最速梯度下降法、牛顿法以及拟牛顿法等。

2 梯度下降

梯度下降算法可以说是机器学习的灵魂，没有这种优化方法，今天我们所有的人工智能，可以说都是空谈。接下来我们通过线性回归，详细讨论一下，梯度下降算法的主要内容。

2.1 批量梯度下降

我们想通过最小化$J(\theta)$去选择最优的θ。首先让我们初始化参数θ，然后改变θ的值，去使得$J(\theta)$变得更小，直至$J(\theta)$不再发生变化，我们称已经收敛到了$J(\theta)$的最小值。具体来说，我们考虑如下的更新规则：

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta) \tag{4}$$

角标j表示对参数向量的所有参数都要进行更新，α是学习率，是一个超参数（不同于参数θ，θ可以在更新过程中发生改变），需要手动调节。:= 表示赋值，把右边的值，替换掉左边的值。这个我们先假设只存在一个样本，求出等式右边的梯度：

$$\begin{aligned}
\frac{\partial}{\partial \theta_j} J(\theta) &= \frac{\partial}{\partial \theta_j} \frac{1}{2}(h_\theta(x) - y)^2 & (5)\\
&= 2 \times \frac{1}{2}(h_\theta(x) - y)\frac{\partial}{\partial \theta_j}(h_\theta(x) - y) & (6)\\
&= (h_\theta(x) - y)\frac{\partial}{\partial \theta_j}(\sum_{i=0}^{d} \theta_i x_i - y) & (7)\\
&= (h_\theta(x) - y)x_j & (8)
\end{aligned}$$

所以对于一个训练样本，我们有：

$$\theta_j := \theta_j - \alpha(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)} \tag{9}$$

这个更新规则称为最小均方（Least Mean Squares）更新规则，文献也称为Widrow-Hoff 学习规则。这个更新规则我们很自然就能得到一些信息，比如括号里面的项为误差项，我们可以看到，误差越大的样本，可以对参数做出一个更大的改变，而误差较小的样本，参数只能得到较小的改变。现在我们推广到n个样本，那么右边的梯度，应该为n个样本的梯度和，即：

$$\theta_j := \theta_j - \alpha \sum_{i=1}^{n}(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)} \tag{10}$$

然后直到算法收敛。我们的模型有$d+1$个参数，所以如果按照以上的更新规则，我们每次更新所有参数，需要进行一个循环，循环的次数为$d+1$。如果采用向量的形式，会简化掉这个循环：

$$\theta := \theta - \alpha \sum_{i=1}^{n}(h_\theta(x^{(i)}) - y^{(i)})x^{(i)} \tag{11}$$

我们通过以上的更新规则，我们可以看出，每次更新就使用了所有的样本，我们把这种方法称为**批量梯度下降**（batch gradient descent）。这里我们需要强调一下，梯度下降是会寻找到局部最优解，我们这里定义的线性回归的损失函数不存在这种情况，因为是凸二次型函数，这里涉及到一些凸优化有关的内容，我们不在这过多涉及，我们只需要知道，只有凸函数是全局最优解，非凸函数，可能找不到全局最优解。这里我们通过一个图片给出梯度下降的一个解释： 最右边的点是初始化的，然后每更新一次，我们就离中心就更进一步，随着迭代次数的增加，达到全局最优解。

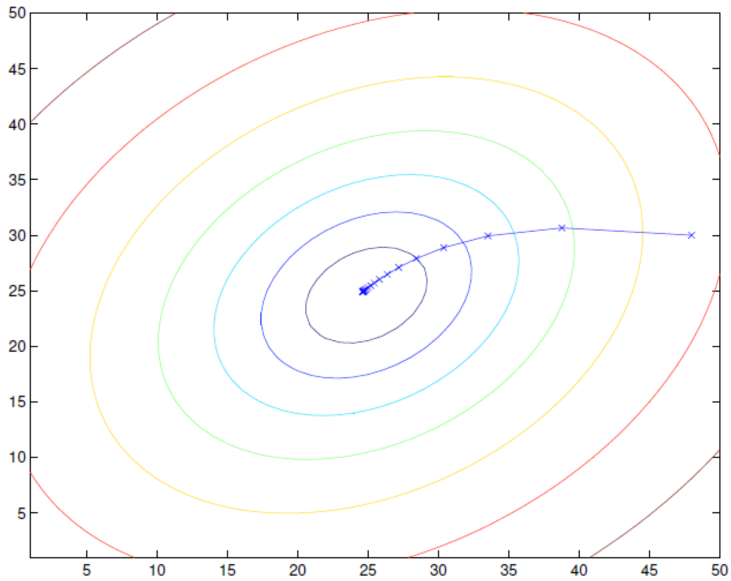

Figure 1: 可视化梯度下降的过程。

2.2 随机梯度下降

目前我们的世界是一个数据大爆炸的时代，数据是以指数级的量增加。当数据集非常大时，每次批量梯度下降如果把所有样本都进行更新是不现实的，所以应运而生，每次随机挑选一个样本进行更新：

$$\theta := \theta - \alpha(h_\theta(x^{(i)}) - y^{(i)})x^{(i)} \tag{12}$$

这里对i进行循环，直到收敛。我们把这种梯度下降，称为**随机梯度下降**（stochastic gradient descent）。这种方法其实时目前在机器学习领域比较常用的。

2.3 小批量梯度下降

在随机梯度下降和批量梯度下降之间，有一种折中的方法，也是因为数据集数量最大，但是又想加快更新速度，所以把数据集进行划分，假设划分为k份。每次更新参数时，只选择其中一份进行更新。这一份的数据量，可大可小，视计算机的性能而定。我们把这种更新方式，称为**小批量梯度下降**（mini-batch gradient descent）。目前计算机科学发展迅猛，并行计算广泛使用，即通过集群分散运算，也是小批量梯度下降的一种体现，也可以加快更新速度。

3 正规化方程

以上我们讨论了数值方法求解优化问题，对于线性回归模型，存在解析解，即闭式解。求解方法就是偏导数为0，解出θ。首先我们把损失函数，用矩阵的形式表达出来。先改写数据集，把每个样本写成行向量的形式，然后堆叠在一起，记为X，具有如下的形式：

$$X = \begin{bmatrix} - & (x^{(1)})^T & - \\ - & (x^{(2)})^T & - \\ & \vdots & \\ - & (x^{(n)})^T & - \end{bmatrix} \tag{13}$$

那么每个样本对应的房价数值，也可以写成一个列向量，记为Y，即：

$$Y = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(n)} \end{bmatrix} \tag{14}$$

通过以上的改写，很容易可以得到：

$$\begin{aligned} X\theta - Y &= \begin{bmatrix} - & (x^{(1)})^T\theta & - \\ - & (x^{(2)})^T\theta & - \\ & \vdots & \\ - & (x^{(n)})^T\theta & - \end{bmatrix} - \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(n)} \end{bmatrix} && (15) \\ &= \begin{bmatrix} h_\theta(x^{(1)}) - y^{(1)} \\ h_\theta(x^{(2)}) - y^{(2)} \\ \vdots \\ h_\theta(x^{(n)}) - y^{(n)} \end{bmatrix} && (16) \end{aligned}$$

因此，损失函数就是上式的平方，即两个向量的内积$z^Tz = \sum_i z_i^2$。

$$\begin{aligned} \frac{1}{2}(X\theta - Y)^T(X\theta - Y) &= \frac{1}{2}\sum_{i=1}^{n}(h_\theta(x^{(i)} - y^{(i)}))^2 && (17) \\ &= J(\theta) && (18) \end{aligned}$$

然后我们令$\nabla_\theta J(\theta) = 0$，得到：

$$\begin{aligned} \nabla_\theta J(\theta) &= \nabla_\theta \frac{1}{2}(X\theta - Y)^T(X\theta - Y) && (19) \\ &= \frac{1}{2}\nabla_\theta((X\theta)^T X\theta - (X\theta)^T Y - Y^T(X\theta) + Y^T Y) && (20) \\ &= \frac{1}{2}\nabla_\theta(\theta^T(X^T X)\theta - Y^T(X\theta) - Y^T(X\theta)) && (21) \\ &= \frac{1}{2}\nabla_\theta(\theta^T(X^T X)\theta - 2Y^T(X\theta)) && (22) \\ &= \frac{1}{2}\nabla_\theta(2(X^T X)\theta - 2X^T Y) && (23) \\ &= X^T X\theta - X^T Y = 0 && (24) \end{aligned}$$

所以我们得到如下的方程：

$$X^T X\theta = X^T Y \tag{25}$$

在左乘$(X^TX)^{-1}$，就得到了参数θ的解析解：

$$\theta = (X^T X)^{-1} X^T Y \tag{26}$$

我们把这个方程，称为**正规化方程**（normal equation）。值得注意的是，并不是所有的矩阵都存在逆，即X^TX对应的矩阵不存在逆时，就要采用奇异值分解（SVD），具体的细节我们不再过多介绍，感兴趣的同学可以查阅相关的书籍与资料。

4 概率解释

这一节，我们从概率的角度出发，我们假设每个样本的估计值是在真实值附近的微小波动，这个波动符合高斯分布，即：

$$y^{(i)} = \theta^T x^{(i)} + \epsilon^{(i)} \tag{27}$$

这里的$\epsilon^{(i)}$就表示这个微小波动，对于每个样本，我们都假设$\epsilon^{(i)}$是独立同分布的（IID），且$\epsilon^{(i)} \sim N(0, \sigma^2)$。我们写出$\epsilon^{(i)}$的表达式：

$$p(\epsilon^{(i)}) = \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{(\epsilon^{(i)})^2}{2\sigma^2}) \tag{28}$$

这意味着：

$$p(y^{(i)}|x^{(i)};\theta)\frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}) \tag{29}$$

我们解释一下等式左边的含义，p表示概率，括号里面的，表示在参数θ是一个未知的确定值情况下（注意，这里θ 不是随机变量），给定样本$x^{(i)}$ 时，$y^{(i)}$ 的概率大小。这里的σ是一个超参数，需要人为的赋值。对于n 个样本，我们写出n 个样本对应的联合概率分布，我们把这个函数，定义为**似然函数**（likelihood）$L(\theta)$：

$$L(\theta) = \prod_{i=1}^{n} p(y^{(i)}|x^{(i)};\theta) \tag{30}$$

在之前我们已经介绍过MLE方法了，这里我们直接使用，求$L(\theta)$的极值，首先对L取对数，因为对数函数并不影响函数的单调性把取对数的似然函数称为对数似然函数$\ell(\theta)$ （log-likelihood function）。得到如下的表达式：

$$\begin{aligned}
\ell(\theta) &= \log L(\theta) &(31)\\
&= \log \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}) &(32)\\
&= \sum_{i=1}^{n} \log \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}) &(33)\\
&= n \log \frac{1}{\sqrt{2\pi}\sigma} - \frac{1}{\sigma^2}\frac{1}{2}\sum_{i=1}^{n}(y^{(i)} - \theta^T x^{(i)})^2 &(34)
\end{aligned}$$

可以看到，最大化$\ell(\theta)$等价于我们最小化之前定义的损失函数：

$$J(\theta) = \frac{1}{2}\sum_{i=1}^{n}(y^{(i)} - \theta^T x^{(i)})^2 \tag{35}$$

5 正则化

我们以上采用的时MLE方法，即认为最优的参数是确定的，我们只不过是不知道，通过求极值得到最优解，这是频率派的观点。贝叶斯派认为，最优参数也是随机变量，也应该有一个先验分布，即我们最大化的目标函数，应该具有如下的形式：

$$\begin{aligned}
\hat{\theta} &= \arg\max_{\theta} p(\theta|Y) &(36)\\
&= \arg\max_{\theta} p(Y|\theta)p(\theta) &(37)\\
&= \arg\max_{\theta} \log p(Y|\theta)p(\theta) &(38)\\
&= \arg\max_{\theta} \log p(Y|\theta) + \log p(\theta) &(39)
\end{aligned}$$

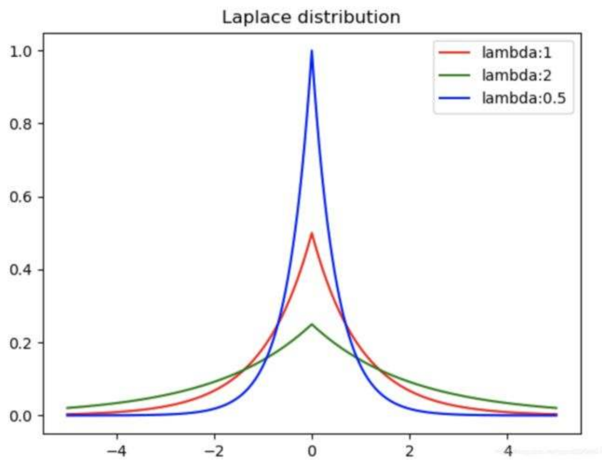

Figure 2: 拉普拉斯分布。

其中$p(\theta)$是一个先验分布，采用不同的先验知识，会得到不同的效果。这种方法我们称为**正则化**（regularization）。正则化是处理模型过拟合的一种手段，所谓过拟合，就是样本容量与样本的维度相比，并不是远远大于，换句话说，就是数据集的样本量不够多，模型的参数存在冗余。那么我们一般有三种方式处理这种情况：

1. 继续收集数据，添加数据。
2. 特征选择，降低数据的维度。
3. 正则化。

有关过拟合相关的内容，我们放到后面详细介绍，我们这里先介绍一下正则化，这里介绍常用的两种先验分布。

5.1 ℓ_1正则化

当我们把先验分布取为拉普拉斯分布，即：

$$p(\theta) = \frac{1}{2\lambda}\exp(-\frac{|\theta-\mu|}{\lambda}) \tag{40}$$

这里λ,μ都是常数，且$\lambda>0$。函数的图形如（2）：

这种情况下的正则化，我们称为ℓ_1正则化，一般情况下，我们简单起见，取$\mu=0$。把最大后验概率可以写成如下的形式：

$$\arg\max_{\theta} L(\theta) + \lambda||\theta||_1,\ \lambda>0 \tag{41}$$

求解这个问题，并不是一个简单问题，需要引入次梯度的概念。最常用的优化方法是近端梯度下降，是梯度下降算法的一种变形，在此我们不在过多介绍。最后我们要说，ℓ_1正则化可以引起稀疏解，即求出来的参数会大多数为0。ℓ_1正则化，又叫做Lasso回归。

5.2 ℓ_2正则化

当我们把先验分布取为高斯分布$\theta\sim N(0,\sigma_0^2)$，即：

$$p(\theta) = \frac{1}{\sqrt{2\pi}\sigma}\exp(-\frac{\theta^2}{2\sigma^2}) \tag{42}$$

把方程带入到公式（39），我们可以得到如下的优化问题：

$$\hat{\theta} = \arg\min_{\theta}[(X\theta - Y)^T(X\theta - Y) + \frac{\sigma^2}{\sigma_0^2}\theta^T\theta] \tag{43}$$

由于σ, σ_0都是超参数，我们可以把前面的系数记为λ。所以求解这个优化问题，同样采用求导数等于0。即：

$$\begin{aligned}
\hat{\theta} = \arg\min_{\theta} J(\theta) + \lambda\theta^T\theta \quad &\rightarrow \quad \frac{\partial}{\partial\theta}J(\theta) + 2\lambda\theta = 0 && (44)\\
&\rightarrow \quad 2X^TX\hat{\theta} - 2X^TY + 2\lambda\hat{\theta} = 0 && (45)\\
&\rightarrow \quad \hat{\theta} = (X^TX + \lambda I)^{(-1)}X^TY && (46)
\end{aligned}$$

这里的I是一个单位阵。并且可以看到，使用2范数进行正则化不仅可以让模型选择θ较小的参数，同时也避免X^TX不可逆的问题。

6 总结

线性回归模型是最简单的模型，但是麻雀虽小，五脏俱全。我们利用最小二乘误差得到了闭式解。同时也发现，在噪声为高斯分布的时候，MLE的解等价于最小二乘误差，而增加了正则项后，最小二乘误差加上ℓ_2正则项等价于高斯噪声先验下的MAP解，加上ℓ_1正则项后，等价于Laplace噪声先验。

传统的机器学习方法或多或少都有线性回归模型的影子：

1. 线性模型往往不能很好地拟合数据，因此有三种方案克服这一劣势：
   (a) 对特征的维数进行变换，例如多项式回归模型就是在线性特征的基础上加入高次项。
   (b) 在线性方程后加一个非线性变换，即引入一个非线性的激活函数，典型的有线性分类模型如感知机。
   (c) 对于一致的线性系数，我们进行多次变换，这样同一个特征不仅仅被单个系数影响，例如多层感知机（深度前馈网络）。
2. 线性回归在整个样本空间都是线性的，我们修改这个限制，在不同区域引入不同的线性或非线性，例如线性样条回归和决策树模型。
3. 线性回归中使用了所有的样本，但是对数据预先进行加工学习的效果可能更好（所谓的维数灾难，高维度数据更难学习），例如PCA算法和流形学习。

References