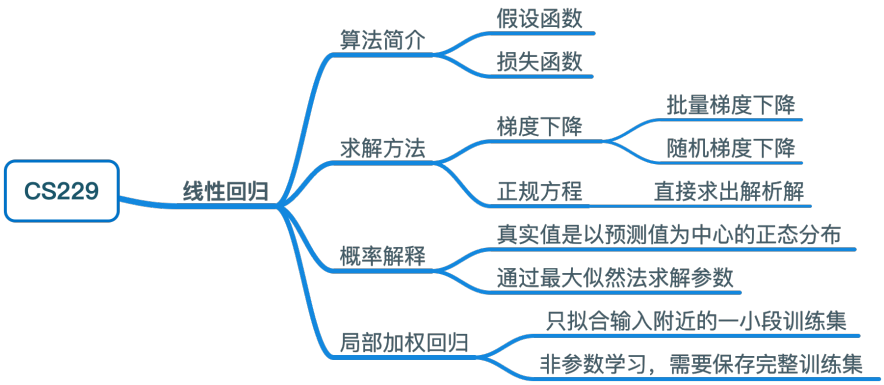
第一章 线性回归



算法简介

- 线性回归是一种监督学习算法，即给定一个训练集，去学习一个假设函数，用来尽量精确地预测每个样本对应的输出
  - 线性回归属于回归算法，其输出变量连续
  - 另一类监督学习算法是分类算法，其输出变量离散
- 线性回归的假设函数为：

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

- 线性回归的代价函数为：

$$J(\theta)=\frac{1}{2}\sum_{i=1}^{m}\left(h_\theta(x^{(i)})-y^{(i)}\right)^2$$

- 线性回归的目的：通过训练集找出使代价函数最小的一组参数 θ（又称最小二乘法）

求解方法

- 对于线性回归代价函数的求解，有两种可选方法：**梯度下降**与**正规方程**

梯度下降

- 梯度下降是一种求解最优化问题的迭代方法，具体步骤为：
  1. 随机选取初始的 θ
  2. 不断地以梯度的方向修正 θ
  3. 最终使 $J(\theta)$ 收敛至局部最优（在最小二乘中，局部最优即全局最优）

$$\theta_j:=\theta_j-\alpha\frac{\partial}{\partial\theta_j}J(\theta) \tag{1}$$

- α 称为学习速率，太小会导致收敛缓慢，太大会导致错过最优点，需要谨慎选择
- 对公式进一步推导（假设只有一个样本点），得到：

$$
\begin{aligned}
\frac{\partial}{\partial\theta_j}J(\theta) &= \frac{\partial}{\partial\theta_j}\frac{1}{2}(h_\theta(x)-y)^2 \\
&= 2\cdot\frac{1}{2}(h_\theta(x)-y)\cdot\frac{\partial}{\partial\theta_j}(h_\theta(x)-y) \\
&= (h_\theta(x)-y)\cdot\frac{\partial}{\partial\theta_j}\left(\sum_{i=0}^{n}\theta_i x_i - y\right) \\
&= (h_\theta(x)-y)x_j
\end{aligned}
$$

- 将上述结果代入 (1) 式得到：

$$
\theta_j := \theta_j + \alpha\left(y^{(i)} - h_\theta(x^{(i)})\right)x_j^{(i)} \tag{2}
$$

  - 上述的结果通过数学变换将减号变成了加号， 方便之后与逻辑回归的结果作比较

分类

- 梯度下降主要可以分为两类：**批量梯度下降**和**随机梯度下降**
- 批量梯度下降：每次计算梯度都需要遍历所有的样本点，当样本量很大时， 计算速度会十分缓慢
- 随机梯度下降：每次只考虑一个样本点，而不是所有样本点，计算速度会提高， 但是收敛过程会比较曲折， 可能无法精确收敛至最优值
  - 随机梯度下降的优化：小批量梯度下降，利用矩阵并行运算，一次处理小批量的样本点，有时可以比随机梯度下降速度更快

梯度方向的选择

- 选择梯度方向的原因是它是使代价函数减小（下降）最大的方向
  - 我们可以利用柯西不等式对这一结论进行证明
- 当 θ 改变一个很小的量时，利用泰勒公式，忽略一阶导数之后的项，得：

$$
\Delta J \approx \frac{\partial J}{\partial\theta_0}\Delta\theta_0 + \frac{\partial J}{\partial\theta_1}\Delta\theta_1 + \cdots + \frac{\partial J}{\partial\theta_n}\Delta\theta_n \tag{3}
$$

- 定义如下变量：

$$
\begin{aligned}
\Delta\theta &\equiv (\Delta\theta_0, \Delta\theta_1, \ldots, \Delta\theta_n)^T \\
\nabla J &\equiv (\frac{\partial J}{\partial\theta_0}, \frac{\partial J}{\partial\theta_1}, \ldots, \frac{\partial J}{\partial\theta_n})^T
\end{aligned}
$$

- 将其代回 (3) 式，得：

$$
\Delta J \approx \nabla J\cdot\Delta\theta
$$

- 根据柯西不等式，有（等号当且仅当 $\Delta\theta$ 与 ∇J 线性相关时成立）：

$$
|\Delta J| \approx |\nabla J\cdot\Delta\theta| \le \|\nabla J\|\cdot\|\Delta\theta\|
$$

- 因此，要使 ΔJ 最小，即 $|\Delta J|$ 最大且 $\Delta J<0$，而当且仅当 $\Delta\theta=-\alpha\nabla J \quad (\alpha>0)$ 时满足条件，即沿着梯度方向调整 θ

正规方程

- 我们可以通过正规方程直接求出 θ 的解析解。推导过程如下：

矩阵导数

- 对一个将 $m\times n$ 的矩阵映射至实数的函数，定义其导数为：

$$\nabla_A f(A) = \begin{bmatrix} \frac{\partial f}{\partial A_{11}} & \cdots & \frac{\partial f}{\partial A_{1n}} \\ \vdots & \ddots & \vdots \\ \frac{\partial f}{\partial A_{m1}} & \cdots & \frac{\partial f}{\partial A_{mn}} \end{bmatrix}$$

- 对一个 $n \times n$ 方阵，它的迹定义为对角线元素之和：

$$\operatorname{tr} A = \sum_{i=1}^{n} A_{ii}$$

- 易证明迹操作具有如下性质（各矩阵为方阵）：

$$\begin{gathered} \operatorname{tr} AB = \operatorname{tr} BA, \\ \operatorname{tr} ABC = \operatorname{tr} CAB = \operatorname{tr} BCA, \\ \operatorname{tr} ABCD = \operatorname{tr} DABC = \operatorname{tr} CDAB = \operatorname{tr} BCDA \end{gathered}$$

- 同样易证明如下性质（a 为实数）：

$$\begin{gathered} \operatorname{tr} A = \operatorname{tr} A^T \\ \operatorname{tr} (A + B) = \operatorname{tr} A + \operatorname{tr} B \\ \operatorname{tr} aA = a \operatorname{tr} A \end{gathered}$$

- 基于以上定义，可以证明一些关于矩阵导数的性质（等式 (7) 只针对非奇异矩阵）：

$$\nabla_A \operatorname{tr} AB = B^T \tag{4}$$
$$\nabla_{A^T} f(A) = (\nabla_A f(A))^T \tag{5}$$
$$\nabla_A \operatorname{tr} ABA^T C = CAB + C^T AB^T \tag{6}$$
$$\nabla_A |A| = |A|(A^{-1})^T \tag{7}$$

最小二乘重现

- 对于训练集，可以写成如下的形式：

$$X = \begin{bmatrix} — (x^{(1)})^T — \\ — (x^{(2)})^T — \\ \vdots \\ — (x^{(m)})^T — \end{bmatrix}, \ \vec{y} = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(m)} \end{bmatrix}$$

- 因为 $h_\theta(x^{(i)}) = (x^{(i)})^T \theta$，我们可以得出：

$$\begin{aligned} X\theta - \vec{y} &= \begin{bmatrix} (x^{(1)})^T \theta \\ (x^{(2)})^T \theta \\ \vdots \\ (x^{(m)})^T \theta \end{bmatrix} - \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(m)} \end{bmatrix} \\ &= \begin{bmatrix} h_\theta(x^{(1)}) - y^{(1)} \\ h_\theta(x^{(2)}) - y^{(2)} \\ \vdots \\ h_\theta(x^{(m)}) - y^{(m)} \end{bmatrix} \end{aligned}$$

- 此外，对于一个向量z，我们有 $z^T z = \sum_i z_i^2$，因此综上可以得出：

$$\begin{aligned} \frac{1}{2}(X\theta - \vec{y})^T (X\theta - \vec{y}) &= \frac{1}{2} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right)^2 \\ &= J(\theta) \end{aligned}$$

- 所以为了使 $J(\theta)$ 最小，即只需找出其导数为 0 时 θ 的值。下面给出详细的求解过程：
  - 首先，将 (5) 式与 (6) 式结合，得：

$$\begin{aligned}
\nabla_{A^T} f(A) &= (\nabla_A f(A))^T && (5)\\
\nabla_A \operatorname{tr} ABA^TC &= CAB + C^TAB^T && (6)\\
\nabla_{A^T} \operatorname{tr} ABA^TC &= (CAB + C^TAB^T)^T \\
&= (CAB)^T + (C^TAB^T)^T \\
&= B^TA^TC^T + BA^TC && (8)
\end{aligned}$$

    - 其中最后两步的推导基于矩阵转置的下列性质：

$$\begin{aligned}
(A+B)^T &= A^T + B^T \\
(AB)^T &= B^TA^T
\end{aligned}$$

  - 基于以上所述，有：

$$\begin{aligned}
\nabla_\theta J(\theta) &= \nabla \frac{1}{2}(X\theta - \vec{y})^T(X\theta - \vec{y}) \\
&= \frac{1}{2}\nabla_\theta(\theta^TX^T - \vec{y}^T)(X\theta - \vec{y}) \\
&= \frac{1}{2}\nabla_\theta \underbrace{(\theta^TX^TX\theta - \theta^TX^T\vec{y} - \vec{y}^TX\theta + \vec{y}^T\vec{y})}_{a=\operatorname{tr} a,\, a\in R} \\
&= \frac{1}{2}\nabla_\theta \underbrace{\operatorname{tr}\,(\theta^TX^TX\theta - \theta^TX^T\vec{y} - \vec{y}^TX\theta + \vec{y}^T\vec{y})}_{\operatorname{tr}\,(A+B)=\operatorname{tr} A+\operatorname{tr} B} \\
&= \frac{1}{2}\nabla_\theta \underbrace{(\operatorname{tr}\theta^TX^TX\theta - \operatorname{tr}\theta^TX^T\vec{y} - \operatorname{tr}\vec{y}^TX\theta + \operatorname{tr}\vec{y}^T\vec{y})}_{\operatorname{tr} A=\operatorname{tr} A^T} \\
&= \frac{1}{2}\nabla_\theta(\operatorname{tr}\theta^TX^TX\theta - 2\operatorname{tr}\vec{y}^TX\theta + \underbrace{\operatorname{tr}\vec{y}^T\vec{y}}_{\text{don't depend on }\theta}) \\
&= \frac{1}{2}\nabla_\theta(\operatorname{tr}\theta^TX^TX\theta - 2\operatorname{tr}\vec{y}^TX\theta) \\
&= \frac{1}{2}(\underbrace{\nabla_\theta\operatorname{tr}\theta^TX^TX\theta}_{\text{use (8), } A^T=\theta, B=B^T=X^TX, C=I} - 2\underbrace{\nabla_\theta\operatorname{tr}\vec{y}^TX\theta}_{\operatorname{tr} ABC=\operatorname{tr} CAB, \text{then use (4)}}) \\
&= \frac{1}{2}(X^TX\theta + X^TX\theta - 2X^T\vec{y}) \\
&= X^TX\theta - X^T\vec{y}
\end{aligned}$$

- 因此，正规方程如下：

$$\begin{aligned}
X^TX\theta &= X^T\vec{y} \\
\theta &= (X^TX)^{-1}X^T\vec{y}
\end{aligned}$$

  - 不可逆问题可以通过伪逆计算或正则化处理解决

概率解释

- 在线性回归中，为什么要选择最小二乘函数作为代价函数？我们可以用概率模型来对其进行解释

概率模型

- 假设真实值与输入之间满足如下等式：

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

  - 其中 $\epsilon^{(i)}$ 是误差项，表示没有被建模的因素或是随机噪声
- 进一步假设误差项是独立同分布的，那么根据中心极限定理，大量相互独立的随机变量符合以 0 为中心的正态分布（可以理解为大量独立随机变量的大部分误差会相互抵消），即 $\epsilon^{(i)} \sim \mathcal{N}(0, \sigma^2)$，那么有：

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

- 而误差的概率和预测出真实值的概率是一样的，因此：

$$p(y^{(i)} \mid x^{(i)}; \theta) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}\right)$$

  - 注意，这里 $p(y^{(i)} \mid x^{(i)}; \theta)$ 不同于 $p(y^{(i)} \mid x^{(i)}, \theta)$，这里指给定 $x^{(i)}$，以 θ 为参数的 $y^{(i)}$ 的分布，因为对于训练集，θ 是客观存在的，只是当前还不确定（这是一种频率学派的观点，之后会细说）
- 因此，我们得出真实值是以预测值为中心的一个正态分布：

$$y^{(i)} \mid x^{(i)}; \theta \sim \mathcal{N}(\theta^T x^{(i)}, \sigma^2)$$

似然函数

- 给定训练集 X 和参数 θ，预测结果等于真实结果的概率，将其看作 θ 的函数，可以理解为 θ 为真实 θ 的可能性（似然性），即：

$$L(\theta) = L(\theta; X, \vec{y}) = p(\vec{y} \mid X; \theta)$$

- 因为每个样本是独立同分布的，所以有：

$$\begin{aligned} L(\theta) &= \prod_{i=1}^{m} p(y^{(i)} \mid x^{(i)}; \theta) \\ &= \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}\right) \end{aligned}$$

- 现在，我们可以通过**最大似然法**，即找出使 $L(\theta)$ 最大的那个 θ，作为对参数 θ 的最佳取值
- 实际应用中，为了简化计算，通常不直接求似然函数的最大值，而是采用**对数似然**函数：

$$\begin{aligned} \ell(\theta) &= \log L(\theta) \\ &= \log \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}\right) \\ &= \sum_{i=1}^{m} \log \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}\right) \\ &= m\log\frac{1}{\sqrt{2\pi}\sigma} - \frac{1}{\sigma^2} \cdot \frac{1}{2}\sum_{i=1}^{m}(y^{(i)} - \theta^T x^{(i)})^2 \end{aligned}$$

- 因此，最大化 $l(\theta)$ 就是最小化：

$$\frac{1}{2}\sum_{i=1}^{m}(y^{(i)} - \theta^T x^{(i)})^2$$

  - 这正是我们之前提出的**最小二乘代价函数**
  - 可以看出 θ 的选择并不依赖于 σ^2

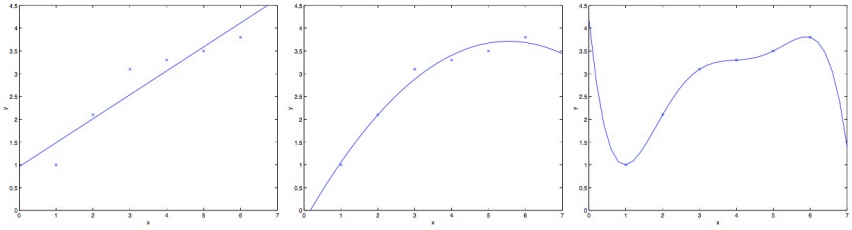
- 概率解释只是对最小二乘法的一种合理解释，其实还有其他的解释方法

局部加权线性回归

- 本节将介绍一种特殊的线性回归算法

欠拟合与过拟合

- 对于传统的线性回归，特征的选择极为重要，对于下面三幅图，我们称第一幅图的模型是**欠拟合**，第三幅图的模型则是**过拟合**（之后会详细介绍）



  - 可以看出，找到一个全局的线性模型去拟合整个训练集，并不是一件简单的事情，往往会引起欠拟合或是过拟合的发生
  - 对于这种情况之后会给出解决方案，而这里我们提出了另外一种思路，即**局部线性加权回归**，这种方案可以使特征的选择的重要性降低

算法思路

- 局部线性加权回归的思路是并不去拟合整个训练集来产生全局的模型，而是在每次预测时，只去拟合给定输入 x 附近的一小段训练集，无论全局训练集是怎样的一条分布曲线，在局部小段数据上，都可以用线性去逼近。具体步骤如下：
  1. 拟合 θ 来最小化 $\sum_i \omega^{(i)}\left(y^{(i)}-\theta^T x^{(i)}\right)^2$
  2. 输出 $\theta^T x$
- 这里 $\omega^{(i)}$ 是**非负权重**，一般取值如下：

$$\omega^{(i)}=exp\left(-\frac{(x^{(i)}-x)^2}{2\tau^2}\right)$$

  - 当 x 为向量时表达式有所不同
  - 可以看出，离给定输入越近的样本点权重越大，拟合程度越高
- $\omega^{(i)}$ 的定义与高斯分布类似，但并没有关系，分布曲线同为钟型
  - τ 称为**带宽参数**，用来控制钟型曲线的顶峰下降速度，即权重变化的快慢，需要根据具体情况作出调整

参数学习与非参数学习

- 局部加权线性回归本质上是一种**非参数学习算法**，而传统的线性回归是一种**参数学习算法**
- 两者的区别在于：
  - 参数学习算法有一组有限的、固定的参数，一旦完成拟合，只需要保存下参数值做预测，而不需要保存完整的训练集
  - 非参数学习算法由于参数不固定，所以需要保存完整的训练集来进行预测，而不仅仅是保存参数
- **非参数学习**导致的结果：为了表达假设 h 而保存的数据将随着训练集的大小而线性增长