拟合算法

插值算法中，得到的多项式f(x)要经过所有样本点。但是如果样本点太多，那么这个多项式次数过高，会造成龙格现象。

尽管我们可以选择分段的方法避免这种现象，但是更多时候我们更倾向于得到一个确定的曲线，尽管这条曲线不能经过每一个样本点，但只要保证误差足够小即可，这就是拟合的思想。**(拟合的结果是得到一个确定的曲线)**

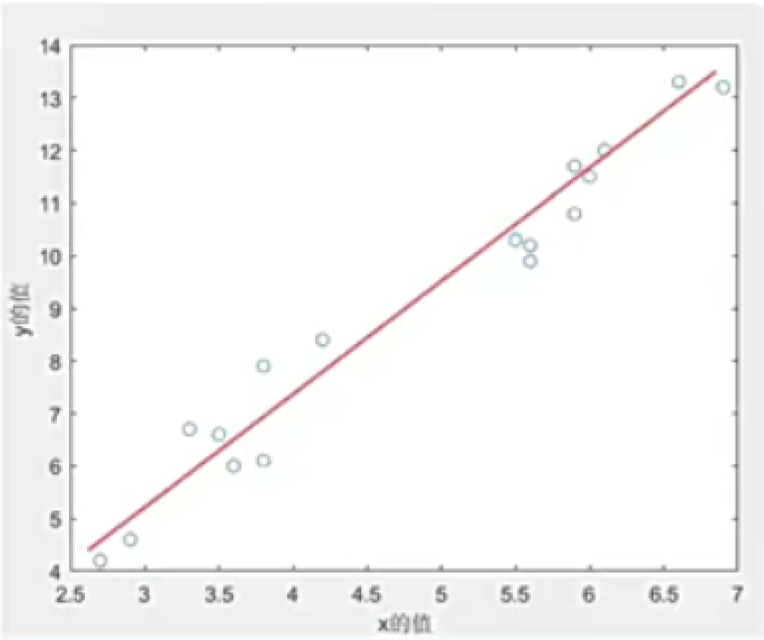
基本概念

1.

设这些样本点为(x_i, y_i), $i = 1, 2, \cdots, n$

我们设置的拟合曲线为$y = kx + b$

问题：k和b取何值时，样本点和拟合曲线最接近。



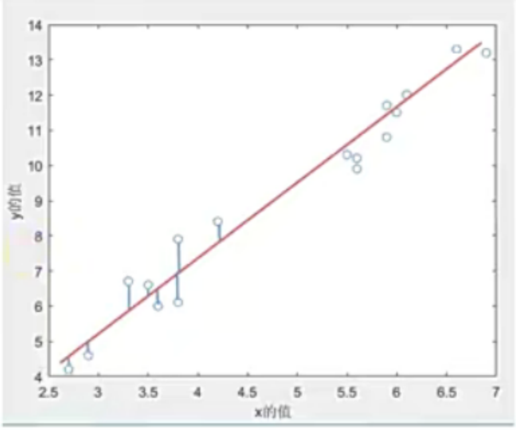
2.

最小二乘法的几何解释

设这些样本点为(x_i, y_i), $i = 1, 2, \cdots, n$

我们设置的拟合曲线为$y = kx + b$

问题：k和b取何值时，样本点和拟合曲线**最接近**。



第一种定义：$\hat{y}_i = kx_i + b$

$$\hat{k}, \hat{b} = \underset{k,b}{arg\min}\left(\sum_{i=1}^{n} |y_i - \hat{y}_i|\right)$$

第二种定义：$\hat{y}_i = kx_i + b$

$$\hat{k}, \hat{b} = \underset{k,b}{arg\min}\left(\sum_{i=1}^{n} (y_i - \hat{y}_i)^2\right)$$

第一种定义有绝对值，不容易求导，因此计算比较复杂。
所以我们往往使用第二种定义，这也正是最小二乘的思想。

为什么不用四次方？

（1）避免极端数据对拟合曲线的影响。

（2）最小二乘法得到的结果和MLE极大似然估计一致。

不用奇数次方的原因：误差会正负相抵。

求解最小二乘法：

设这些样本点为(x_i, y_i), $i = 1, 2, \cdots, n$，我们设置的拟合曲线为$y = kx + b$

令拟合值$\hat{y}_i = kx_i + b$

$$那么\hat{k}, \hat{b} = \underset{k,b}{arg\min}\left(\sum_{i=1}^{n} (y_i - \hat{y}_i)^2\right) = \underset{k,b}{arg\min}\left(\sum_{i=1}^{n} (y_i - kx_i - b)^2\right)$$

令$L = \sum_{i=1}^{n} (y_i - kx_i - b)^2$，现要找$k, b$使得$L$最小。

（L在机器学习中被称为损失函数，在回归中也常被称为残差平方和）

$$\begin{cases} \dfrac{\partial L}{\partial k} = -2\sum_{i=1}^{n} x_i(y_i - kx_i - b) = 0 \\ \dfrac{\partial L}{\partial b} = -2\sum_{i=1}^{n} (y_i - kx_i - b) = 0 \end{cases} \Rightarrow \begin{cases} \sum_{i=1}^{n} x_i y_i = k\sum_{i=1}^{n} x_i^2 + b\sum_{i=1}^{n} x_i \\ \sum_{i=1}^{n} y_i = k\sum_{i=1}^{n} x_i + bn \end{cases} \Rightarrow \begin{cases} n\sum_{i=1}^{n} x_i y_i = kn\sum_{i=1}^{n} x_i^2 + bn\sum_{i=1}^{n} x_i \\ \sum_{i=1}^{n} y_i \sum_{i=1}^{n} x_i = k\sum_{i=1}^{n} x_i \sum_{i=1}^{n} x_i + bn\sum_{i=1}^{n} x_i \end{cases}$$

$$n\sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} y_i \sum_{i=1}^{n} x_i = kn\sum_{i=1}^{n} x_i^2 - k\sum_{i=1}^{n} x_i \sum_{i=1}^{n} x_i \Rightarrow \hat{k} = \frac{n\sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} y_i \sum_{i=1}^{n} x_i}{n\sum_{i=1}^{n} x_i^2 - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} x_i}$$

$$同理我们可得到：\hat{b} = \frac{\sum_{i=1}^{n} x_i^2 \sum_{i=1}^{n} y_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} x_i y_i}{n\sum_{i=1}^{n} x_i^2 - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} x_i}$$

3.如何评价拟合的好坏

拟合优度（可决系数）R^2

总体平方和SST：$Total\ sum\ of\ squares$：$SST=\sum_{i=1}^{n}(y_i-\bar{y})^2$

误差平方和SSE：$The\ sum\ of\ squares\ due\ to\ error$：$SSE=\sum_{i=1}^{n}(y_i-\hat{y}_i)^2$

回归平方和SSR：$Sum\ of\ squares\ of\ the\ regression$：$SSR=\sum_{i=1}^{n}(\hat{y}_i-\bar{y})^2$

可以证明：$SST=SSE+SSR$（要用到我们求导得到的两个等式）

拟合优度：$0\le R^2=\frac{SSR}{SST}=\frac{SST-SSE}{SST}=1-\frac{SSE}{SST}\le 1$

R^2越接近1，说明误差平方和越接近0，误差越小说明拟合的越好。

（注意：R^2只能用于拟合函数是线性函数时，拟合结果的评价）

线性函数和其他函数（例如指数函数）比较拟合的好坏，直接看SSE即可

（未来你可能有机会看到R^2是个负数）

R^2 只能用于拟合函数是线性函数时，拟合结果的评价。

4.

思考：$y=a+bx^2$是线性函数吗？

是的，因为我们这里说的线性函数是指**对参数为线性（线性于参数）**。

由于本书主要讨论像方程（2.2.2）那样的线性模型，所以我们必须知道线性一词的真正含义，因为对它可作两种解释。

☐ **对变量为线性**

对线性的第一种并且也许是更“自然”的一种解释是，Y 的条件期望值是 X_i 的线性函数，比如说，方程（2.2.2）。[①] 从几何意义上说，这时回归曲线是一条直线。按照这种解释，诸如 $E(Y\mid X_i)=\beta_1+\beta_2X_i^2$ 的回归函数，由于变量 X 以幂或指数 2 出现，就不是线性的。

☐ **对参数为线性**

对线性的第二种解释是，Y 的条件期望 $E(Y\mid X_i)$ 是参数 β 的一个线性函数；它可以是或不是变量 X 的线性函数。[②] 对于这种解释，$E(Y\mid X_i)=\beta_1+\beta_2X_i^2$ 就是一个线性（于参数）回归模型。为了看出这一点，让我们假设 X 取值为 3。因此，$E(Y\mid X=3)=\beta_1+9\beta_2$，显然它是 β_1 和 β_2 的线性函数。图 2—3 中所示的所有模型因此也都是线性回

在函数中，参数仅以一次方出现，且不能乘以或除以其他任何的参数，并不能出现参数的复合函数形式。

matlab中的曲线拟合工具箱

在matlab中，点击界面上方的APP选项，在数学与优化选项中，选择曲线拟合器。

工具箱提供的拟合函数有多种：

- **Custom Equations**：用户自定义的函数类型；
- **Exponential**：指数逼近，有2种类型， $a*\exp(b*x)$ 、 $a*\exp(b*x)+c*\exp(d*x)$；
- **Fourier**：傅立叶逼近，有7种类型，基础型是 $a0+a1*\cos(x*w)+b1*\sin(x*w)$；
- **Gaussian**：高斯逼近，有8种类型，基础型是 $a1*\exp(-((x-b1)/c1)^2)$；
- **Interpolant**：插值逼近，有4种类型，Nearest neighbor、Linear、Cubic、Shape-preserving（PCHIP）；
- **Linear Fitting**：线性拟合；
- **Polynomial**：多形式逼近；
- **Power**：幂逼近，有2种类型，$a*x^b$ 、 $a*x^b+c$；
- **Rational**：有理数逼近；
- **Smoothing Spline**：平滑逼近；
- **Sum of Sin Functions**：正弦曲线逼近，有8种类型，基础型是 $a1*\sin(b1*x+c1)$；
- **Weibull**：只有一种，$a*b*x^(b-1)*\exp(-a*x^b)$；

可以在得到拟合函数的值以后，使用工具箱生成函数文件，在自己的代码中使用。生成的函数文件，可以直接复制函数function后面的内容到代码的主文件中使用。

一般题目中会给出需要拟合的函数。