多元统计分析—R 与 Python 的实现

回归

吴喜之

June 28, 2019

Example

波士顿住房数据 (BostonHousing2.csv) 这是一个非常经典的数据, 为 1970 年人口普查的 506 个人口普查区住房数据.[a] 该数据包含在程序包 `mlbench`[b] 中, 名为 `BostonHousing`, 包含 Harrison 和 Rubinfeld(1979) 的原始数据, 而带有额外空间信息的数据 `BostonHousing2` 是校正版本.

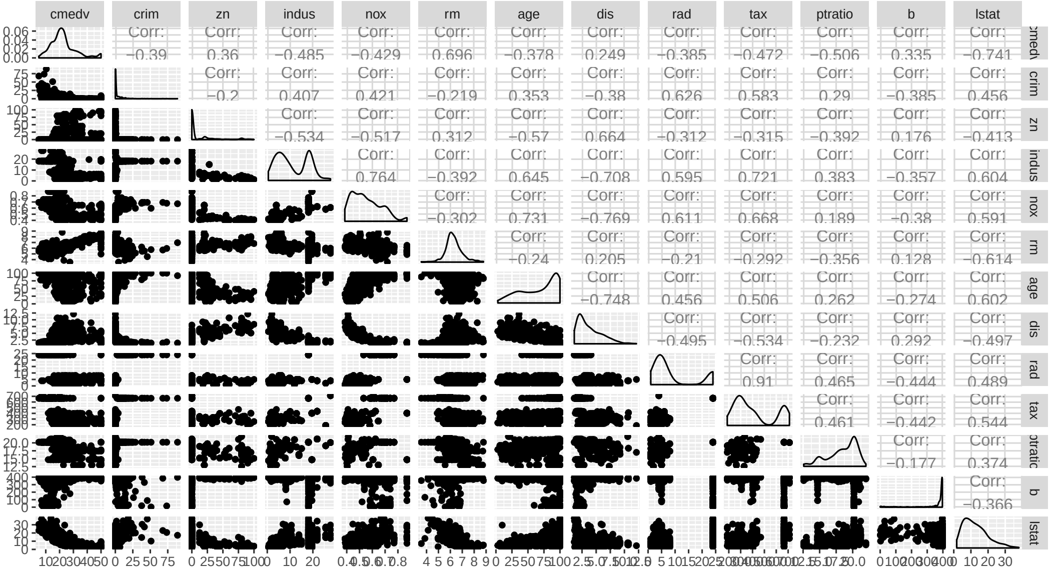
[a] Newman, D.J. & Hettich, S. & Blake, C.L. & Merz, C.J. (1998). UCI Repository of machine learning databases [http://www.ics.uci.edu/~mlearn/MLRepository.html]. Irvine, CA: University of California, Department of Information and Computer Science.

[b] Friedrich Leisch & Evgenia Dimitriadou (2010). mlbench: Machine Learning Benchmark Problems. R package version 2.1-1.

原始数据有 14 个变量的 506 个观察值，其中，medv(自住房屋房价中位数，单位：千美元) 是原始的目标变量，其他变量包括：crim (城镇的人均犯罪率)、zn (占地面积超过 25000 平方英尺的住宅用地的比例)、indus (每个镇的非零售业务比例，单位：英亩)、chas (有关查尔斯河的虚拟变量，如果挨着河为 1，否则为 0)、nox (一氧化氮浓度，单位：ppm)、rm (平均每间住房的房间数量)、age (1940 年以前建成的自住单位的房龄比例)、dis (五个波士顿就业中心的加权距离)、rad (高速公路的可达性指数)、tax (每万美元全价物业值的财产税率)、ptratio (城镇学生与教师的比例)、b $(=1000(B-0.63)^2$，其中的 B 是城镇黑人的比例)、lstat (低收入人口比例)；更正过的数据集有以下附加变量：cmedv (修正了的自住房价中位数，单位：千美元)、town (镇名称)、tract (人口普查区)、lon (人口普查区的经度)、lat (人口普查区的纬度).
我们将用 cmedv (修正了的自住房屋房价中位数) 作为因变量，舍弃原来的 medv，而其他多数变量作为自变量 (除了 town(镇名称)，tract(人口普查区)，lon(人口普查区的经度)，lat(人口普查区的纬度) 等变量).

该数据有一个用哑元 (0-1) 表示的定性变量 (chas)，在 R 中，哑元表示的定性变量在处理时需要用函数 `factor` 做标记 (“因子化”). 比如，在做回归之前使用下面的语句来确定，否则会被计算机当成数量.

数据的描述：例1波士顿住房数据的变量相关图



还可以输出各个变量的汇总，下面是代码及输出:

```
> summary(w)
     cmedv            crim                 zn             indus        chas
 Min.   : 5.00   Min.   : 0.00632   Min.   :  0.00   Min.   : 0.46   0:471
 1st Qu.:17.02   1st Qu.: 0.08204   1st Qu.:  0.00   1st Qu.: 5.19   1: 35
 Median :21.20   Median : 0.25651   Median :  0.00   Median : 9.69
 Mean   :22.53   Mean   : 3.61352   Mean   : 11.36   Mean   :11.14
 3rd Qu.:25.00   3rd Qu.: 3.67708   3rd Qu.: 12.50   3rd Qu.:18.10
 Max.   :50.00   Max.   :88.97620   Max.   :100.00   Max.   :27.74
      nox               rm             age              dis              rad
 Min.   :0.3850   Min.   :3.561   Min.   :  2.90   Min.   : 1.130   Min.   : 1.000
 1st Qu.:0.4490   1st Qu.:5.886   1st Qu.: 45.02   1st Qu.: 2.100   1st Qu.: 4.000
 Median :0.5380   Median :6.208   Median : 77.50   Median : 3.207   Median : 5.000
 Mean   :0.5547   Mean   :6.285   Mean   : 68.57   Mean   : 3.795   Mean   : 9.549
 3rd Qu.:0.6240   3rd Qu.:6.623   3rd Qu.: 94.08   3rd Qu.: 5.188   3rd Qu.:24.000
 Max.   :0.8710   Max.   :8.780   Max.   :100.00   Max.   :12.127   Max.   :24.000
      tax           ptratio            b              lstat
 Min.   :187.0   Min.   :12.60   Min.   :  0.32   Min.   : 1.73
 1st Qu.:279.0   1st Qu.:17.40   1st Qu.:375.38   1st Qu.: 6.95
 Median :330.0   Median :19.05   Median :391.44   Median :11.36
 Mean   :408.2   Mean   :18.46   Mean   :356.67   Mean   :12.65
 3rd Qu.:666.0   3rd Qu.:20.20   3rd Qu.:396.23   3rd Qu.:16.95
 Max.   :711.0   Max.   :22.00   Max.   :396.90   Max.   :37.97
```

线性回归模型

如果自变量有 p 个 (对于例1波士顿住房数据, $p = 13$), 线性回归模型的形式为:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p + \epsilon, \tag{0.1}$$

这里, ϵ 为误差项, 凡是因变量和自变量之间的线性关系解释不了的部分都属于误差. 如果有 n 个样本 (对于例1, $n = 506$), 则线性模型的样本形式为:

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip} + \epsilon_i, \quad i = 1, 2, \ldots, n. \tag{0.2}$$

线性回归模型: 矩阵形式

如果记

$$\boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}; \quad \boldsymbol{X} = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1p} \\ 1 & x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & x_{n2} & \cdots & x_{np} \end{bmatrix}; \quad \boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{bmatrix}; \quad \boldsymbol{\epsilon} = \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{bmatrix},$$

则矩阵形式的式 (0.2) 为:

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}. \tag{0.3}$$

$$\hat{y} = X\hat{\beta}$$

回归模型的一般形式

一般的回归模型可以写成

$$\boldsymbol{y} = \mu(\boldsymbol{X}, \boldsymbol{\beta}, \boldsymbol{\epsilon});$$

如果误差项假定是可加的，则上述模型可写为:

$$\boldsymbol{y} = \mu(\boldsymbol{X}, \boldsymbol{\beta}) + \boldsymbol{\epsilon}.$$

再假定线性模型，有 $\mu(\boldsymbol{X}, \boldsymbol{\beta}) = \boldsymbol{X}\boldsymbol{\beta}$，就是式 (0.3) 的形式. 人们选取线性模型的原因主要是最简单直观，而且容易计算. 其实，现实问题很少是线性的.

损失函数

定义回归中的残差 (residual) 为因变量值及其拟合值之间的差: $\boldsymbol{e} = \boldsymbol{y} - \hat{\boldsymbol{y}}$ 或 $e_i = y_i - \hat{y}_i \ (i = 1, 2, \ldots, n)$. 在第 i 个观测值处的损失函数可以定义为某个残差的函数 $\rho(e_i) = \rho(y_i - \hat{y}_i)$. 当然, 在独立样本的假定下, 我们可以主观选择总的损失为单独观测值损失之和:

$$\sum_{i=1}^{n} \rho(y_i - \hat{y}_i) = \sum_{i=1}^{n} \rho(e_i).$$

不同损失函数导致最小二乘回归、最小一乘回归、分位数回归

下面的问题就是选取损失函数 $\rho()$ 的形式，这不是数学问题，也不是科学问题，而是工程问题．你可以选择绝对值，也可以选择绝对值的某次方，比如对于线性回归问题式 (0.2) 来说：

- 当 ρ 为绝对值函数 $\rho(u) = |u|$ 时，回归称最小一乘回归，损失为：

$$\sum_{i=1}^{n} e_i = \sum_{i=1}^{n} |y_i - \hat{y}_i|.$$

- 当 ρ 为平方函数 $\rho(u) = u^2$ 时，回归称最小二乘回归，损失为：

$$\sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2;$$

- 当 ρ 为函数 $\rho(u) = u(\tau - I(u < 0))$ 时 (这里 $I(u < 0)$ 当 $u < 0$ 时等于 1，否则为 0，而 τ 为在 $(0, 1)$ 区间的实数)，回归称为分位数回归，损失为：

$$\sum_{i=1}^{n} e_i(\tau - I(e_i < 0)) = \sum_{i=1}^{n} (y_i - \hat{y}_i) \left[\tau - I\left(y_i - \hat{y}_i < 0\right)\right].$$

图11为三种损失函数图: 绝对值 (左)、二次函数 (中)、$\tau = 0.3$ 的分位数损失函数 (右).

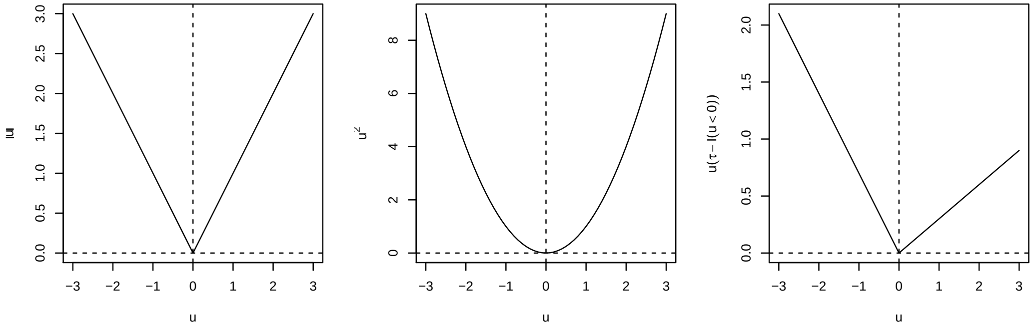


图11中前两种回归是对称的损失函数, 而最后一种是非对称的.

注意

实际生活中的损失有很多是非对称的，比如一个木制房梁，做完后长了一厘米和短了一厘米的损失很不一样，买东西称重时多称了和少称了的损失也不是对称的，把癌症病人误诊成没病和把健康的人误诊为有病的损失也不一样．但人们为什么主要讨论对称的，而且是最小二乘回归呢？这主要是因为二次函数的推导 (比如求导数等) 及计算相对简单，而且有一些 “漂亮” 的数学结论．在一百年前数据量不大时，用手工计算也可以得出最小二乘回归的结果，而其他两种回归只有在计算机时代才能广泛应用．损失函数不是推导出来的，而是根据目标及回归的可行性主观选择的．

- 最小二乘回归的参数估计满足 (没有惩罚项)

$$\hat{\boldsymbol{\beta}} = \mathop{\textbf{arg min}}_{\boldsymbol{\beta}} \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2,$$

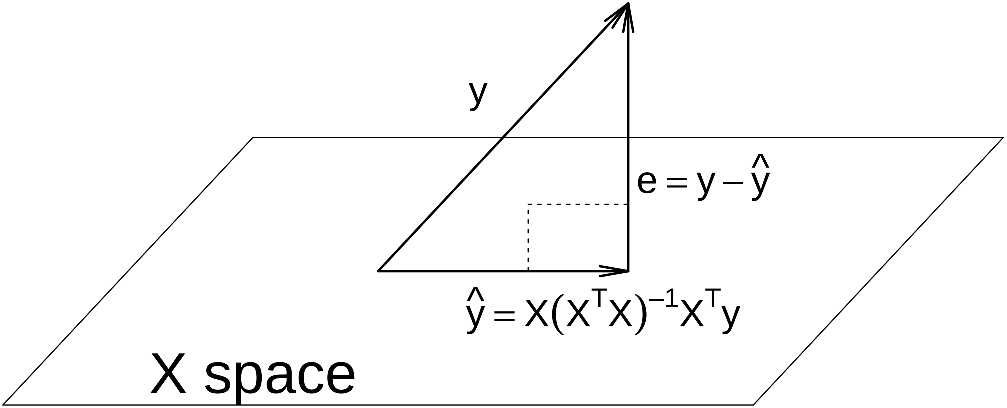
- 岭回归的参数估计满足 (包含惩罚项 $\lambda \sum_{j=1}^{p} \beta_j^2$)

$$\hat{\beta}^{(\textbf{ridge})} = \mathop{\textbf{arg min}}_{\boldsymbol{\beta}} \sum_{i=1}^{n} \left[\left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \right].$$

- lasso 回归的参数估计满足 (包含惩罚项 $\lambda \sum_{j=1}^{p} |\beta_j|$)

$$\hat{\beta}^{(\textbf{lasso})} = \mathop{\textbf{arg min}}_{\boldsymbol{\beta}} \sum_{i=1}^{n} \left[\left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right].$$

最小二乘线性回归的参数估计 最小二乘回归往往用图14来描述，那里的 $\boldsymbol{e} = \boldsymbol{y} - \hat{\boldsymbol{y}}$ 就是原始的 $\boldsymbol{y}$ 投影到 $\boldsymbol{X}$ 张成的空间 (图中画成 (超) 平面的样子) 上的 $\hat{\boldsymbol{y}}$ 的残差，该图能够使得下面的过程更容易理解.



最小二乘估计就是找到使得残差平方和最小的参数

最小二乘线性回归的参数估计

最小二乘估计就是找到使得残差平方和最小的参数, 即

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 = \arg\min_{\boldsymbol{\beta}} \left[(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})^\top (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}) \right].$$

关于 $\boldsymbol{\beta}$ 求偏导数, 解方程

$$\begin{aligned} \frac{\partial \left[(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})^\top (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}) \right]}{\partial \boldsymbol{\beta}} &= \frac{\partial (\boldsymbol{y}^\top \boldsymbol{y} - 2\boldsymbol{y}^\top \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\beta}^\top \boldsymbol{X}^\top \boldsymbol{X}\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} \\ &= -2\boldsymbol{X}^\top \boldsymbol{y} + 2\boldsymbol{X}^\top \boldsymbol{X}\boldsymbol{\beta} = 0 \end{aligned}$$

可以得到

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{X}^\top \boldsymbol{y}.$$

最小二乘线性回归的参数估计

因变量的拟合值为:

$$\hat{\boldsymbol{y}} = \boldsymbol{X}\hat{\boldsymbol{\beta}} = \boldsymbol{X}(\boldsymbol{X}^\top \boldsymbol{X})^{-1}\boldsymbol{X}^\top \boldsymbol{y}, \tag{0.4}$$

这里, 矩阵 $\boldsymbol{P} \equiv \boldsymbol{X}(\boldsymbol{X}^\top \boldsymbol{X})^{-1}\boldsymbol{X}^\top$ 称为预测矩阵 (prediction matrix) 或者帽子矩阵 (hat matrix), 称为后者是因为它作用到 $\boldsymbol{y}$, 使其 "戴了帽子"(变成 $\hat{\boldsymbol{y}}$).[1] 预测矩阵 $\boldsymbol{P}$ 把任何向量, 诸如 $\boldsymbol{y}$, 投影到由 $\boldsymbol{X}$ 张成的空间之中, 因此有 $\boldsymbol{P}\boldsymbol{y} = \hat{\boldsymbol{y}}$, 显然,

$$\boldsymbol{P}\boldsymbol{e} = \boldsymbol{P}(\boldsymbol{y} - \hat{\boldsymbol{y}}) = \boldsymbol{P}(\boldsymbol{y} - \boldsymbol{X}(\boldsymbol{X}^\top \boldsymbol{X})^{-1}\boldsymbol{X}^\top \boldsymbol{y}) = \boldsymbol{P}\boldsymbol{y} - \boldsymbol{X}(\boldsymbol{X}^\top \boldsymbol{X})^{-1}\boldsymbol{X}^\top \boldsymbol{X}(\boldsymbol{X}^\top \boldsymbol{X})^{-1}\boldsymbol{X}^\top$$

也就是说, $\boldsymbol{e}$ 到 $\boldsymbol{X}$ 张成的平面的投影为 $\boldsymbol{0}$, 即残差 $\boldsymbol{e}$ 和 $\boldsymbol{X}$ 张成的平面垂直, 当然也有 $\boldsymbol{e} \perp \hat{\boldsymbol{y}}$.

[1] $\boldsymbol{P}$ 也称为投影矩阵, 除了记号 "$\boldsymbol{P}$", 它也被各种文献记为 "$\boldsymbol{H}$" 或 "$\boldsymbol{M}$" 等.

得到例1的线性最小二乘回归系数估计的代码及结果如下:

```
> a=lm(cmedv~.,w);a$coefficients
(Intercept)      crim          zn       indus       chas1         nox          rm
     0.3637   -0.1062     0.04772     0.02325       2.692     -0.1774       3.789
        age       dis         rad         tax     ptratio           b       lstat
  0.0005749    -1.502      0.3038    -0.01270     -0.9239    0.009228     -0.5307
```

根据上面计算的输出, 该线性模型

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \cdots + \hat{\beta}_{13} x_{13}$$

的线性最小二乘回归系数估计为:

$$\hat{\beta}_0 = 0.363;\ \hat{\beta}_1 = -0.106; \hat{\beta}_2 = 0.048;\ \hat{\beta}_3 = 0.023;\ \hat{\beta}_4 = 2.692;$$
$$\hat{\beta}_5 = -0.1774;\ \hat{\beta}_6 = 3.789;\ \hat{\beta}_7 = 0.0006;\ \hat{\beta}_8 = -1.502;$$
$$\hat{\beta}_9 = 0.304;\ \hat{\beta}_{10} = -0.013;\ \hat{\beta}_{11} = -0.924;\ \hat{\beta}_{12} = 0.009;\ \hat{\beta}_{13} = -0.531.$$

到此为止，我们已经完整地介绍了最小二乘回归的全过程：从假定的线性模型和选择的二次损失函数到具体的参数估计．但是，似乎还少了些什么，这就是如何评价最小二乘回归结果的好坏．不同的视角产生不同的方法及结论．下面首先介绍在计算机时代才能实现的基于预测精度的交叉验证方法．

交叉验证

所谓交叉验证就是用一部分数据建立模型, 这部分数据集合称为训练集 (training set); 用另一部分数据检验该模型对这部分数据的预测精度, 这部分数据集合称为测试集 (testing set). 交叉验证中预测精度不好的模型肯定不是好模型.

训练集和测试集的确定方法有很多. 常用的一种就是$\boldsymbol{m}$ 折交叉验证 (m-fold cross validation), 它把数据随机分成 m 份 (折), 各折的观测值数目尽量相等. 然后做 m 次验证, 每次轮流用 1 份数据作为测试集, 其余 $m-1$ 份作为训练集, 用训练集建模, 再用根据训练集所建立的模型拟合测试集. 这样, 在 m 次验证之后得到 m 次预测误差, 最终得到交叉验证的平均误差.

交叉验证的误差度量

回归交叉验证的误差通常基于残差平方和 (SSE) 或均方误差 (MSE). 记训练集的自变量矩阵为 $\boldsymbol{X}_{train}$, 训练集的因变量为 $\boldsymbol{y}_{train}$, 记测试集的自变量矩阵为 $\boldsymbol{X}_{test}$, 测试集的因变量为 $\boldsymbol{y}_{test}$.

交叉验证的残差平方和为 (下面假定进入测试集的观测值总数为 n[2]):

$$SSE_{cv} = \sum_{i=1}^{n} \left(y_{test\,i} - \hat{y}_{test\,i}\right)^2 ,$$

而交叉验证的均方误差为:

$$MSE_{cv} = \frac{1}{n} SSE_{cv} = \frac{1}{n} \sum_{i} \left(y_{test\,i} - \hat{y}_{test\,i}\right)^2 .$$

[2]如果原数据样本量为 n, 进行 K 折交叉验证, 那么每个观测值都会轮流作为测试集成员出现一次, 这时测试集的总样本量也是 n. 如果把数据仅仅分成一个训练集和一个测试集, 那么原数据样本量和测试集的样本量就不一样了.

NMSE 和 R^2 如果没有模型，最朴素的预测就是均值，那时，可用均值 $\bar{\boldsymbol{y}}_{test}$ 来代替上面 SSE_{cv} 或 MSE_{cv} 的 $\hat{\boldsymbol{y}}_{test}$(分别记为 SST_{cv} 和 MST_{cv})。我们可以用 SST_{cv} 或 MST_{cv} 作为除数来将 SSE_{cv} 或 MSE_{cv} 标准化，得到标准化均方误差 (normalized mean square error, NMSE)

$$NMSE_{cv}=\frac{SSE_{cv}}{SST_{cv}}=\frac{MSE_{cv}}{\mathrm{MST}_{cv}}=\frac{\sum_{i=1}^{n}\left(y_{test\,i}-\hat{y}_{test\,i}\right)^2}{\sum_{i=1}^{n}\left(y_{test\,i}-\bar{y}_{test}\right)^2},$$

同样，也可以得到另一个等价指标 R^2_{cv}:

$$R^2_{cv}=1-NMSE_{cv}=1-\frac{\sum_{i=1}^{n}\left(y_{test\,i}-\hat{y}_{test\,i}\right)^2}{\sum_{i=1}^{n}\left(y_{test\,i}-\bar{y}_{test}\right)^2},$$

该指标在经典回归中 (即没有测试集或训练集就是测试集时) 称为可决系数 (coefficient of determination)(按照机器学习术语也称为记分或得分 (score)).

显然，如果标准化均方误差 $NMSE_{cv} > 1$ 或者等价地 $R^2_{cv} < 0$，说明有模型还不如没有模型 (用朴素的均值作为预测值)．显然，交叉验证的这两个等价度量不仅仅适用于最小二乘回归，而且适用于任何模型，可以用于不同模型预测精度的比较．

对于例1的数据求标准化均方误差及记分，我们做 $Z = 10$ 折交叉验证（对于例1的数据，每份（折）数据约 50 个观测值）。用函数CV 把数据行下标随机打乱，输出 Z 个下标集。然后用这个函数得到的 10 个下标集组成 10 个数据集，轮流每次用 9 个数据集合并作为训练集，再用一个作为测试集，做十次拟合，得到 $NMSE_{cv} = 0.2739$ 及 $R^2_{cv} = 0.7261$。如果不用交叉验证，即全部数据既是训练集又是测试集，则得到 $NMSE = 0.256$ 及 $R^2 = 0.744$。

经典线性回归的数学假定:

1. 模型对于参数的线性形式为:

$$\boldsymbol{y} = \boldsymbol{X\beta} + \boldsymbol{\epsilon} \text{ 或 } y_i = \beta_0 + \beta_1 X_{i1} + \cdots + \beta_p X_{ip} + \epsilon_i.\ i = 1, 2, \ldots, n;$$

2. $\boldsymbol{y}$ 的所有值都是互相独立的, 如果把 $\boldsymbol{X}$ 看成是固定的, 则等价于 $\boldsymbol{\epsilon}$ 的所有值都是互相独立的, 这意味着没有自相关;
3. 对于每个 $\boldsymbol{X}$ 的值, $\boldsymbol{y}$ 的分布是正态分布的, 如果把 $\boldsymbol{X}$ 看成是固定的, 则等价于 $\boldsymbol{\epsilon}$ 的分布是正态分布;
4. 条件期望 $E(\boldsymbol{y}|\boldsymbol{X}) = \boldsymbol{X\beta}$ 或 $E(\boldsymbol{\epsilon}|\boldsymbol{X}) = \boldsymbol{0}$, 协方差阵 $\text{Cov}(\boldsymbol{y}|\boldsymbol{X}) = \text{Cov}(\boldsymbol{\epsilon}|\boldsymbol{X}) = \sigma^2\boldsymbol{I}$, 这意味着所有观测值独立同方差 (homoscedasticity), 对于正态分布意味着所有观测值都独立同分布 (当然误差项也独立同分布);
5. 自变量和 $\boldsymbol{\epsilon}$ 独立;

注意: 如果假定自变量是弱外生的 (weak exogeneity), 可以近似地把 $\boldsymbol{X}$ 看成是固定的, 上面的概率就不是条件概率了, 自变量和 $\boldsymbol{\epsilon}$ 独立的条件就用不着了.

> 当自变量被看成是固定的, 人们所说的最小二乘线性回归的基本假定就是一句话: 观测值独立同正态分布. 或者用公式表示为: $\boldsymbol{y} \sim N(\boldsymbol{X\beta}, \sigma^2\boldsymbol{I})$, $\boldsymbol{\epsilon} \sim N(\boldsymbol{0}, \sigma^2\boldsymbol{I})$.

注意

1. 所有上述假定都无法核对.
2. 把因变量和自变量的关系假定为线性关系的主要原因是当年人们的能力只能勉强应对线性模型所带来的数学及计算等一些问题.
3. 把样本假定为独立同正态分布, 这大多不合乎实际, 但基于与上一款同样的理由, 这种假定在数学上是比较方便的. 这种分布假定和最小二乘回归方法本身无关, 仅仅和与回归相关的各种推断有关, 比如 t 检验及 F 检验统计量的分布、各种估计量的分布及性质等等.

没有假定则无法做任何经典统计推断 (诸如假设检验等), 只有基于计算机算法的机器学习出现之后, 才有可能摆脱主观假定的束缚.