非参数回归

慧航

2024年6月

加权最小二乘

在线性回归中的加权问题：可以使用加权最小二乘法，即最小化经过权重调整的误差平方和：

$$\min_{\beta} \sum_{i=1}^{N} \left[w_i \left(y_i - x_i' \beta \right)^2 \right]$$

从而得到：

$$\hat{\beta}^w = \left(\sum_{i=1}^{N} w_i x_i x_i' \right)^{-1} \left(\sum_{i=1}^{N} w_i x_i y_i \right)$$

其中w_i为权重。

局部估计

考虑一个一维的x

- 如果如果我们希望获得

$$\mathbb{E}\left(y|x=x_0\right)$$

的估计，我们仅仅关心在$x=x_0$点处的估计

- 一个最简单的方法是在最小二乘法的目标函数中，给与$x=x_0$点附近的样本残差平方以更大的权重，而远离$x=x_0$点处的样本残差平方以更小的权重
- 那么只要最小化加权的最小二乘目标函数：

$$\min_{\beta}\sum_{i=1}^{N}\left[w_i\left(y_i-\alpha-\beta x_i\right)^2\right]$$

- 问题：权重如何选取？

核函数

权重一般可以如下选取：

- 令$K(x)$为一个以纵轴对称的函数，即$K(x)=K(-x)$，从而$\int_{\mathbb{R}} xK(x)\,dx=0$，且在$x\in[0,\infty)$是单调递减的，$K(x)\geq 0$
- 令权重：

$$w_i = K\left(\frac{x_i - x_0}{h}\right)$$

其中$h>0$为窗宽（bandwidth），而$K(x)$被称为“核函数”（kernel function）

- 一般核函数可以选取为对称分布的密度函数。

Rectangle核函数

如果令：

$$K_0(x) = \begin{cases} 1 & |x| \leq 1 \\ 0 & |x| > 1 \end{cases}$$

那么权重为：

$$w_i = K_0\left(\frac{x_i - x_0}{h}\right) = \begin{cases} 1 & x_0 - h \leq x_i \leq x_0 + h \\ 0 & otherwise \end{cases}$$

带入到目标函数中去，就得到了：

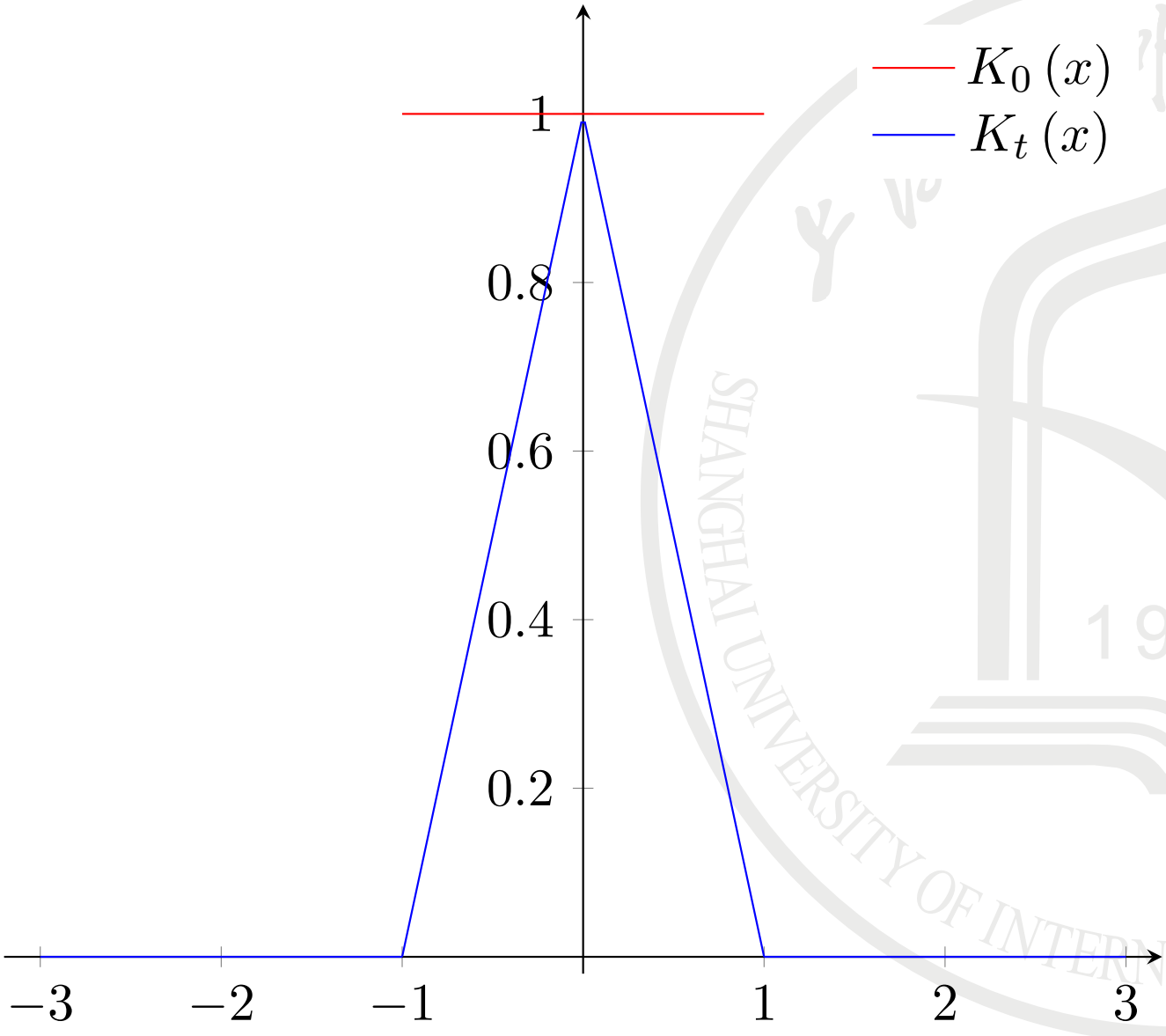
$$\min_{\beta} \sum_{i=1}^{N} \left[1\left\{|x_i - x_0| \leq h\right\}(y_i - \alpha - \beta x_i)^2\right]$$

以上目标函数等价于在区间$[x_0 - h, x_0 + h]$内做简单的最小二乘法。

Rectangle核函数

- 这也就是h取名“窗宽”的由来：
  - h决定了x_0附近区间的大小，h越小，则在x_0附近取的区间越小，使用的数据量也就越少。
- 注意到$K_0(x)$在临近的区间内权重都相等，所以我们也称其为矩形（rectangle）核函数

核函数



Triangle核函数

- 我们还可以令权重在临近区间内不相等：当x_{i}距离x_{0}越近时权重越高
- 此时可以在$K_0(x)$的基础上进行修改，使用：

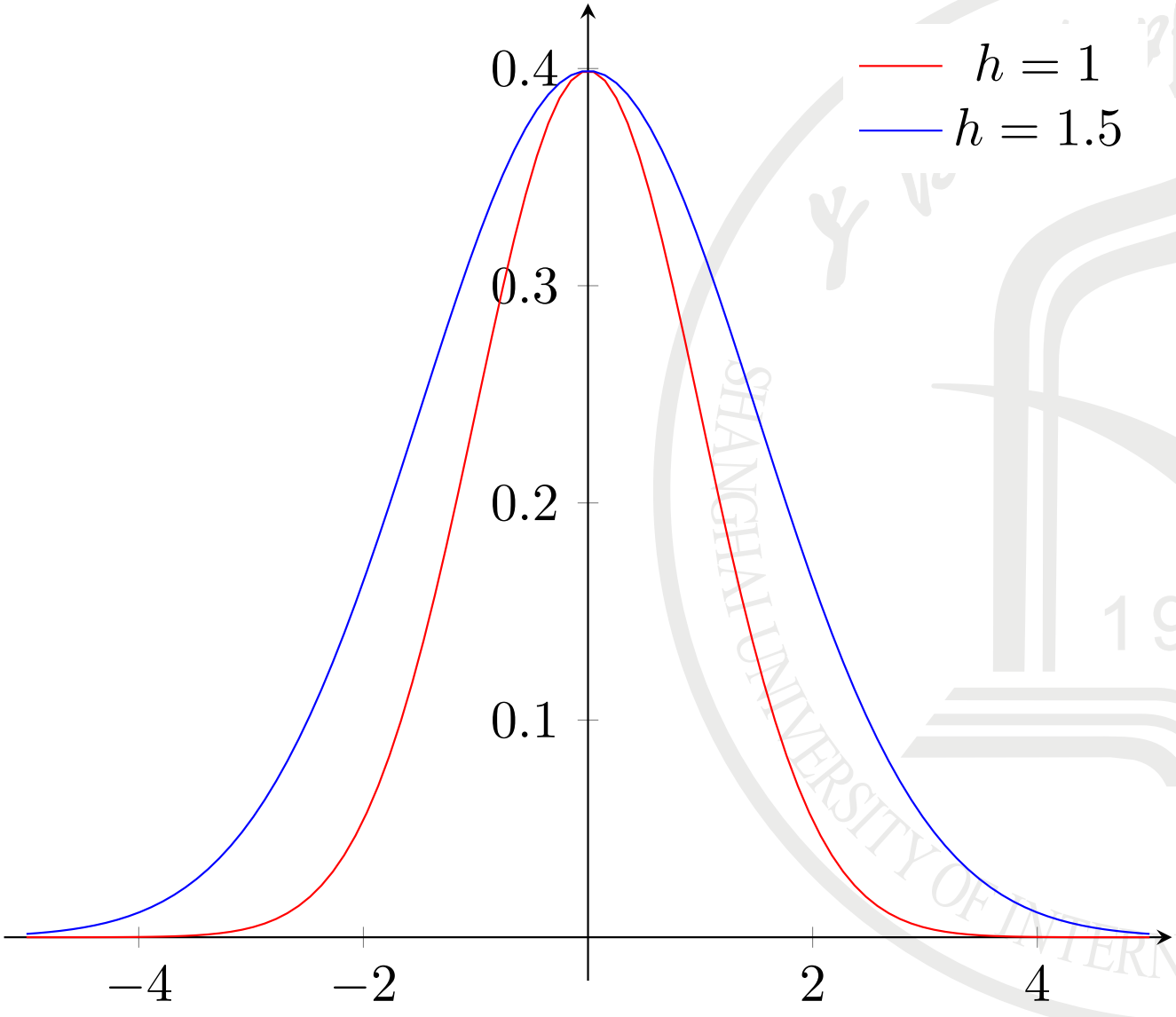
$$K_t(x) = \begin{cases} 1-|x| & |x| \leq 1 \\ 0 & |x| > 1 \end{cases}$$

- 该函数同样在$|x|>1$时取值为0，但是在$|x| \leq 1$这个区间内，随着x远离0，有个递减的过程。
- 我们称该核函数为三角（triangle）核函数。

高斯核函数

- 或者，也可以取$K(x)=\phi(x)$，其中$\phi(\cdot)$为标准正态分布的密度函数，该核函数被称为高斯核函数。
- 权重：$w_i=\phi\left(\frac{x_i-x_0}{h}\right)$
  - 同样的，当x_i越靠近x_0时，权重w_i越大
  - 不过不会像$K_0(x)$那样变为0，而是会收敛到0。
  - 而在这种情况下，h同样也被称为窗宽，因为h扮演的作用是一样的：
    - h越大，则随着x_i远离x_0，权重收敛到0的速度越慢，相当于用了一个更“大”的区间；
    - 反之h越小，则权重收敛到0的速度越快，相当于用了一个更“小”的区间。

核函数



局部常数估计

- 如果只在局部使用常数项对$\mathbb{E}(y|x=x_0)$进行估计：

$$\min_{\beta} \sum_{i=1}^{N} \left[w_i \left(y_i - \alpha \right)^2 \right]$$

- 此时可以计算得到：

$$\hat{y}_0 = \hat{\alpha} = \frac{\sum_{i=1}^{N} w_i y_i}{\sum_{i=1}^{N} w_i} = \sum_{i=1}^{N} \frac{w_i}{\sum_{i=1}^{N} w_i} y_i = \sum_{i=1}^{N} w_i^* y_i$$

其中w_i^*为规范化的权重，使得$\sum_i w_i^* = 1$。

局部常数估计

- 如果将核函数带入，就得到了：

$$\hat{y}_0 = \frac{\sum_{i=1}^{N} K\left(\frac{x_i - x_0}{h}\right) y_i}{\sum_{i=1}^{N} K\left(\frac{x_i - x_0}{h}\right)}$$

  - 如果我们取核函数$K(\cdot) = K_0(\cdot)$，以上估计量无非就是在$(x_0 - h, x_0 + h)$邻域内对y_i进行一个简单的加权平均
  - 如果取$K(\cdot) = \phi(\cdot)$，也是同样的道理
- 以上估计量也被称为“局部常数项估计”（local constant estimator）。

局部常数估计

一个模拟

假设数据生成过程：

$$y = \exp\left(\sin x^3\right) + u$$

其中$x \sim U(0,2), u \sim N(0,1)$，假设样本量$N = 300$。如果我们关心的是当$x = 2$时$y$的预测值（真实值为2.6895），我们选取标准正态分布的密度函数作为核函数，此外选取$h = 0.1$，计算得到$\hat{y}_{x=2} = 2.07$（local_constant.do）。

局部线性估计

- 注意到，在上例中，局部常数项回归计算的实际上是在$x = 2$的一个小的邻域中的均值
- 然而在上例中，由于x=2恰好是x的取值范围的上界，所以我们实际上只使用了x=2左边的一个小的邻域$(2 - h, 2)$。
- 可以想象，由于在$x = 2$左边，真实的数据生成过程是单调递增的，所以在这个小邻域中计算均值会低估$\mathbb{E}(y|x = 2)$。
  - 上例的结果也可以看出，局部常数项的估计的确低估了$\mathbb{E}(y|x = 2)$。

局部线性估计

- 为此，我们可以在这个小的邻域中做一个线性回归。
- 一个常用的处理方法是首先计算$x^{\#}=x-x_0$，带入到目标函数中就是：

$$\min_{\alpha,\beta}\sum_{i=1}^{N}\left[K\left(\frac{x_i-x_0}{h}\right)\left(y_i-\alpha-\beta x_i^{\#}\right)^2\right]$$

$$\Leftrightarrow \min_{\alpha,\beta}\sum_{i=1}^{N}\left[K\left(\frac{x_i-x_0}{h}\right)\left[y_i-\alpha-\beta\left(x_i-x_0\right)\right]^2\right]$$

- 当$x=x_0$时，$x^{\#}=0$，从而对于$\mathbb{E}\left(y|x=x_0\right)$的预测$\hat{y}_{x=x_0}=\hat{\alpha}$。
- 由于该方法可以看作是在一个小的邻域中使用线性回归对$x=x_0$处的y进行预测，所以也叫做**局部线性**（**local linear**）回归。

局部线性回归

局部线性回归模拟

接上例，我们可以使用如下代码进行局部线性回归：

```
gen w=normalden((x-2)/0.1)
gen x_2=x-2
reg y x_2 [iw=w]
di _b[_cons]
```

结果为$\hat{y}_{x=2} = 2.776$，与真实值更为接近，并且没有低估真实值了。

局部多项式回归

更一般的，我们可以使用局部多项式（local polynomial）回归：

$$\min_{\beta} \sum_{i=1}^{N} \left[K\left(\frac{x_i - x_0}{h}\right) \left(y_i - \alpha - \sum_{k=1}^{K} \beta_k \left(x_i^{\#}\right)^k \right)^2 \right]$$

在局部进行更精细的逼近。

局部多项式回归

局部多项式回归模拟

接上例，我们可以使用如下代码进行局部三阶多项式回归：

```
gen w=normalden((x-2)/0.1)
gen x_2=x-2
gen x_22=x_2^2
gen x_23=x_2^3
reg y x_2* [iw=w]
di _b[_cons]
```

结果为$\hat{y}_{x=2} = 2.495$。

窗宽和多项式阶数选取

- 多项式阶数并不是越多越好，过高的多项式阶数会导致预测结果，特别是在端点的预测效果不稳定。
- 而关于窗宽h，考虑局部常数项回归以及$K(\cdot)=K_0(\cdot)$作为例子，此时的估计量无非是(x_0-h,x_0+h)区间的所有y_i的平均数
  - 可以想象一个过小的窗宽意味着能够使用的样本量更少，所以估计量的方差会很大；但是由于窗口比较小，我们上面所讨论的“低估”就会更不明显，也就是说估计量的偏差（bias）会更小。
  - 而反过来，一个大的窗宽会降低估计量的方差，但是偏差则会提高。
- 回忆均方误差可以写为偏差的平方和方差之和，所以理论上应该会有一个最优的h，使得均方误差达到最小。

选取方法

一般多项式阶数和窗宽的选取方法：

- 理论推导计算
  - 一些特殊情况有理论计算结果，如非参数回归、RD设计等
- 交叉验证
  - 我们仅仅关注$x = x_0$处的预测，所以在做交叉验证时，并不需要所有的样本点都作为测试集，而是仅仅把离x_0最近的一些点作为测试集就好了。

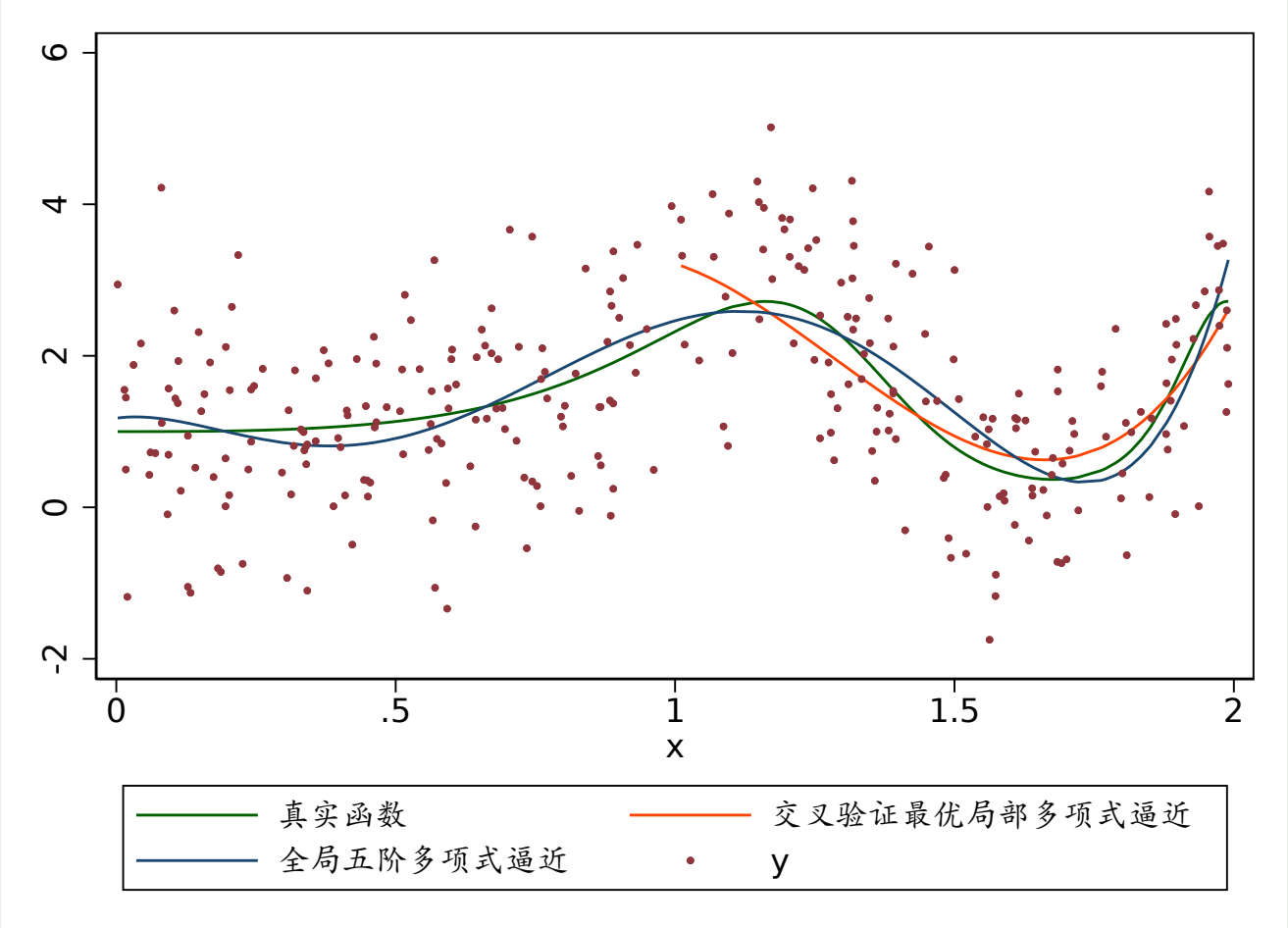
窗宽和多项式阶数选取

交叉验证选取多项式阶数和窗宽

接上例，我们使用local_poly_cv.do代码，在$p = 1, 2, ..., 5$阶多项式、$h = 0.01, 0.02, ..., 0.5$的范围内搜索最优的p和h的组合。在以上代码中，我们针对每一个(p, h)的组合，都使用与$x = 2$最近的10个点作为测试集，用留一验证的方法在测试集上计算交叉验证的均方误差，最后选取交叉验证均方误差最小的(p, h)的组合，并进行了局部多项式的回归。选取的结果是
当$h = 0.48, p = 3$时，均方误差最小，此时预测为$\hat{y}_{x=2} = 2.743$，与真实值2.6895非常接近。

窗宽和多项式阶数选取

交叉验证选取多项式阶数和窗宽



多个解释变量

- 现在如果我们有两个解释变量，即$x=(x_1,x_2)'$，如果需要预测$\mathbb{E}\left(y|x_1=x_1^0,x_2=x_2^0\right)$，那么可以在$x$的两个维度上分别取一个核函数和一个窗宽，并使用两个变量核函数的乘积作为权重：

$$w_i=K_1\left(\frac{x_{1i}-x_1^0}{h_1}\right)K_2\left(\frac{x_{2i}-x_2^0}{h_2}\right)$$

  然后做加权最小二乘就可以了。
- 由于必须在多个维度都很接近x_0才会对$\mathbb{E}\left(y|x=x_0\right)$的估计有贡献，然而在有限样本的情况下，随着维度的增加，x_0附近的点会越来越少，为了保证相对较小的方差，就必须扩大范围，而扩大范围会造成比较大的偏差，所以在多维解释变量的情况下，以上的局部多项式估计可能并不理想，我们将这种现象称为“维数的诅咒”（the curse of dimensionality）。