《Python统计计算》

（2021年秋季学期）

翟祥
北京林业大学

E-mail: zhaixbh@126.com

统计计算

第14章 数值积分与MCMC

统计计算

- 在统计计算和其它科学计算中，经常需要计算各种函数的值，对函数进行逼近，用数值方法计算积分、微分。
- 函数逼近（非参回归）

  多项式逼近

  连分式逼近

  样条平滑
- 插值

  多项式插值

  样条插值
- 数值积分和数值微分

函数逼近

- 统计计算中经常要计算函数值，如计算基本初等函数及其他特殊函数；
- 当函数只在有限点集上给定函数值，要在包含该点集的区间上用公式给出函数的简单表达式。
- 这些都涉及到在区间$[a,b]$上用简单函数逼近已知复杂函数的问题，这就是**函数逼近问题。**

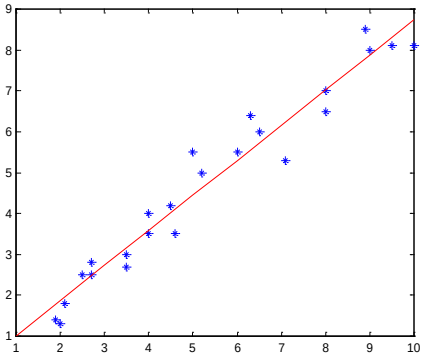
函数逼近

- 仍然是已知 $x_1 \cdots x_m$；$y_1 \cdots y_m$，求一个简单易算的近似函数 $P(x) \approx f(x)$。
- 困难：1、m 很大；2、② y_i 本身是测量值，不准确，即 $y_i \neq f(x_i)$

  - 使 $\max\limits_{1 \le i \le m} |P(x_i) - y_i|$ 最小 /* minimax problem */
  - 使 $\sum\limits_{i=1}^{m} |P(x_i) - y_i|$ 最小
  - 使 $\sum\limits_{i=1}^{m} |P(x_i) - y_i|^2$ 最小 /* Least-Squares method */

函数逼近

编 号	拉伸倍数 x_i	强 度 y_i	编 号	拉伸倍数 x_i	强 度 y_i
1	1.9	1.4	13	5	5.5
2	2	1.3	14	5.2	5
3	2.1	1.8	15	6	5.5
4	2.5	2.5	16	6.3	6.4
5	2.7	2.8	17	6.5	6
6	2.7	2.5	18	7.1	5.3
7	3.5	3	19	8	6.5
8	3.5	2.7	20	8	7
9	4	4	21	8.9	8.5
10	4	3.5	22	9	8
11	4.5	4.2	23	9.5	8.1
12	4.6	3.5	24	10	8.1

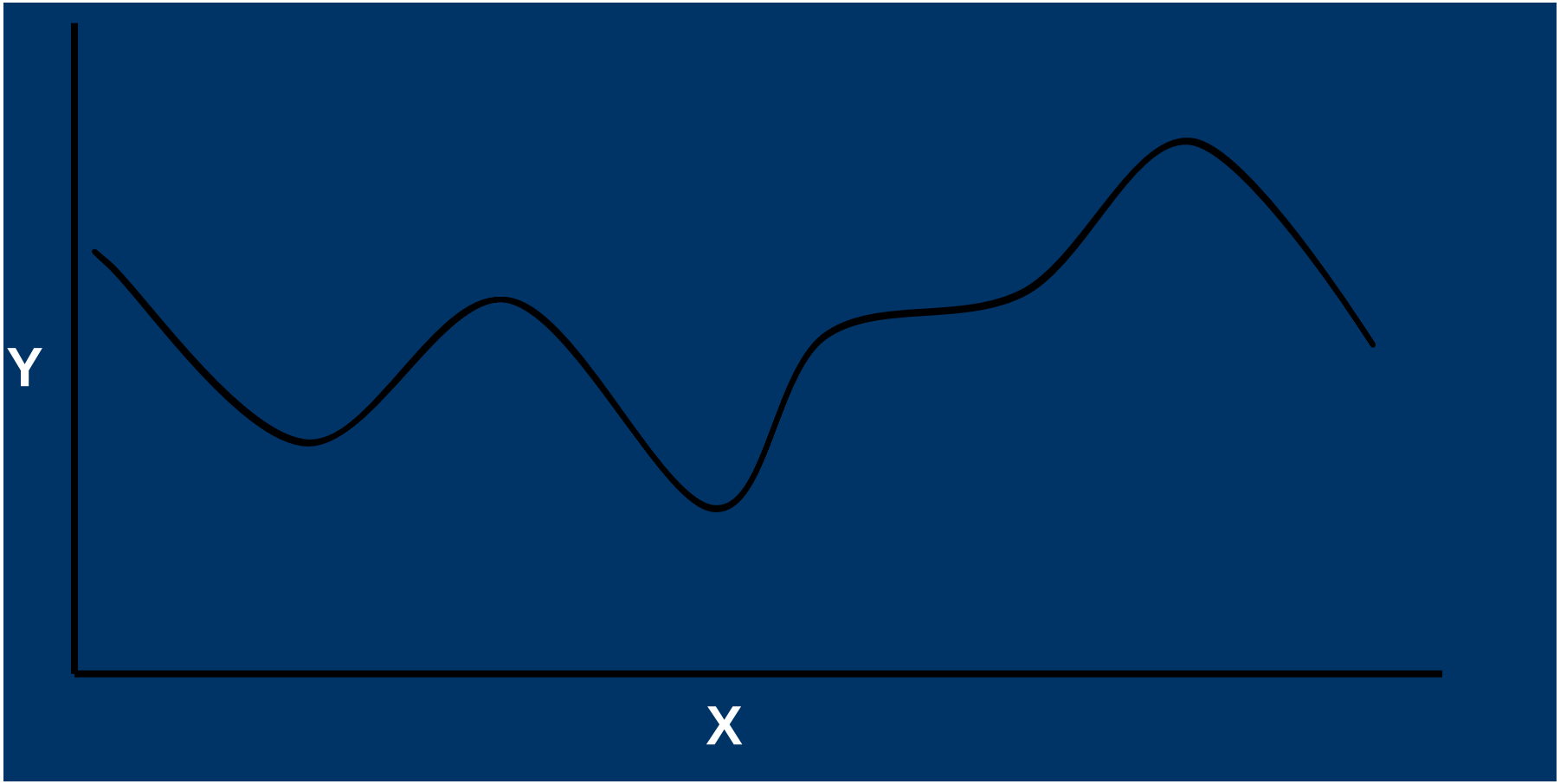


$$y(x) = a + bx$$

ε为噪声

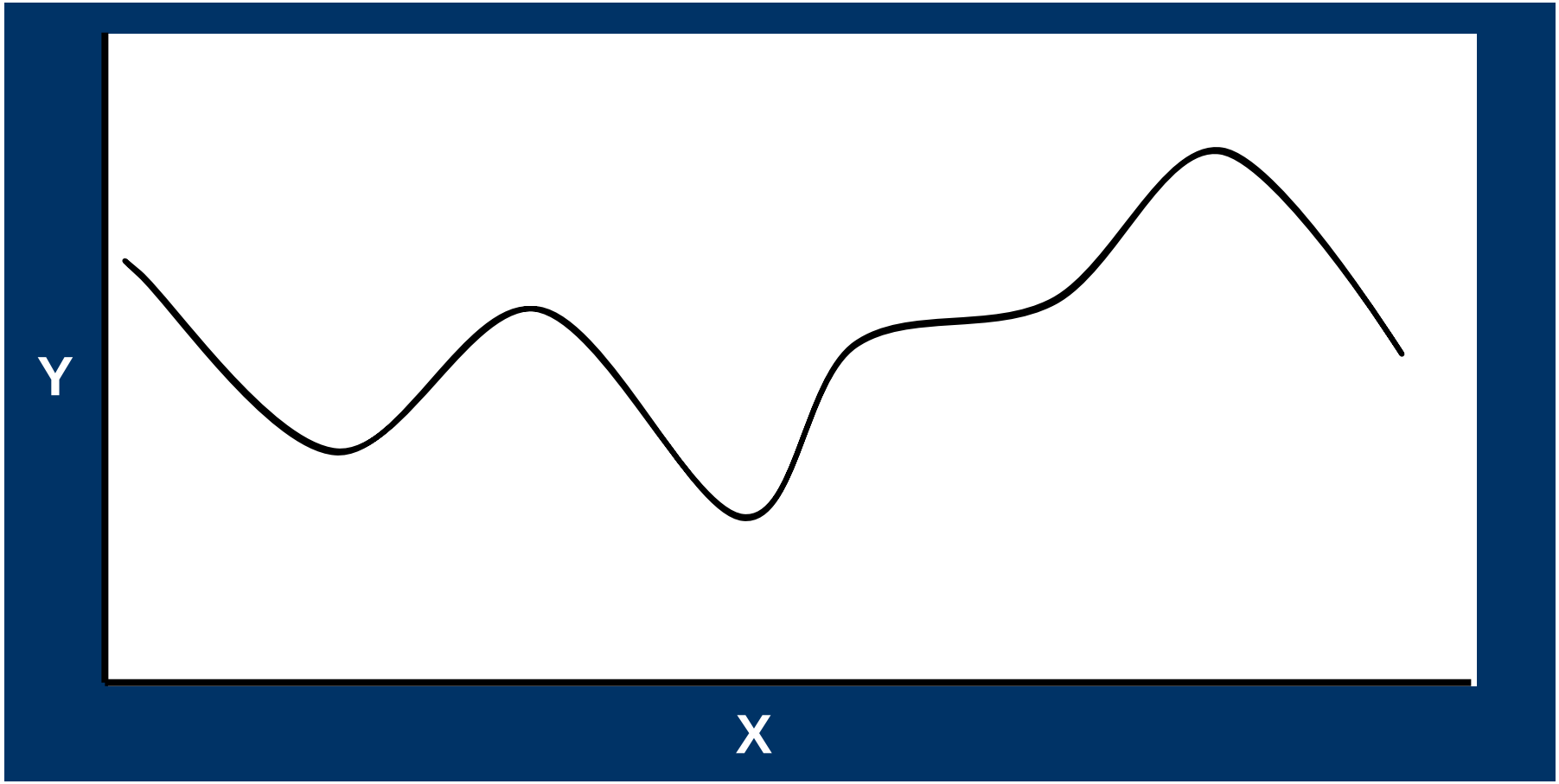
Idea of Local Regression

- It is difficult to find an appropriate parametric form for complicated curves.



Idea of Local Regression

Locally such curves can be well

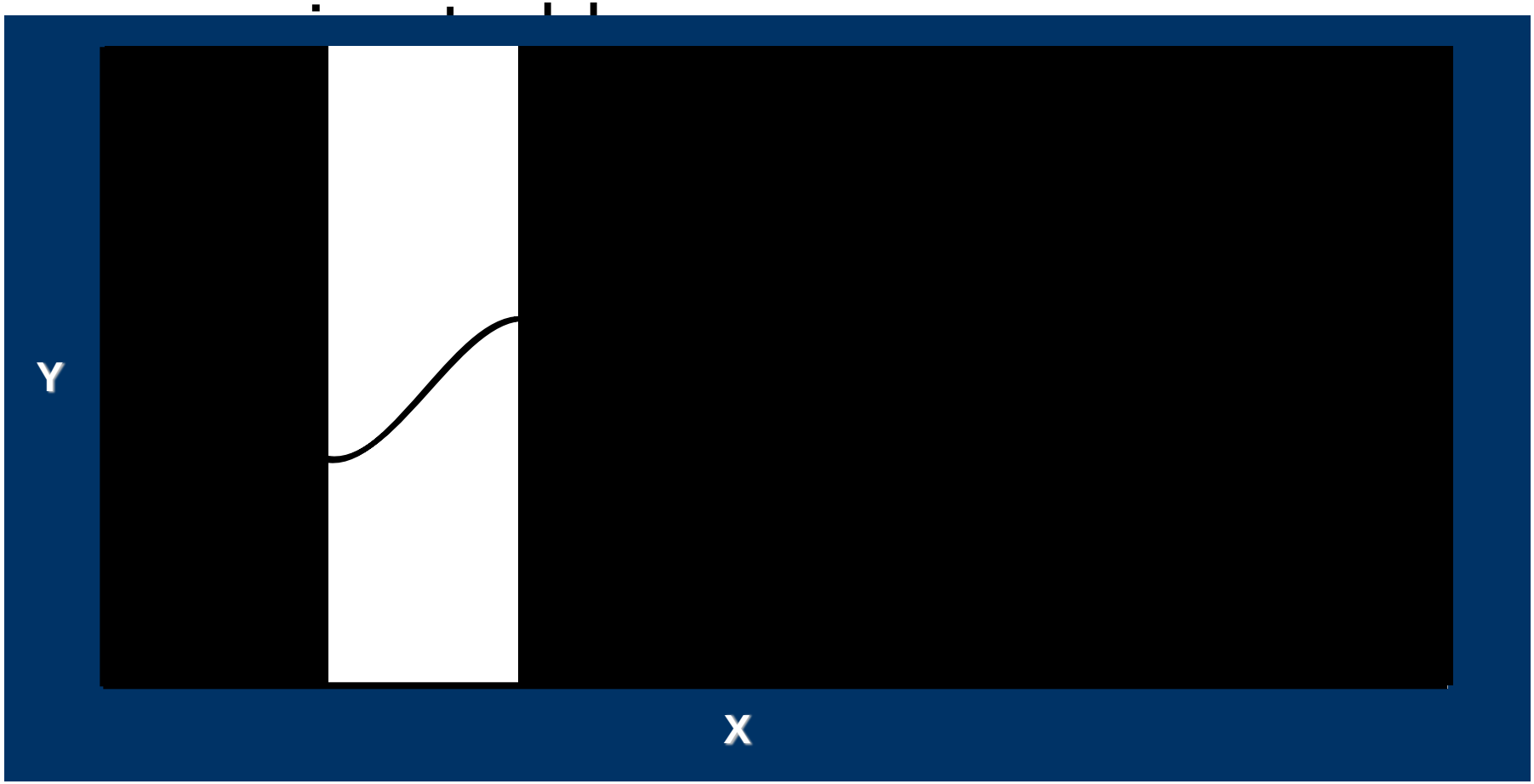


Idea of Local Regression

- Locally such curves can be well

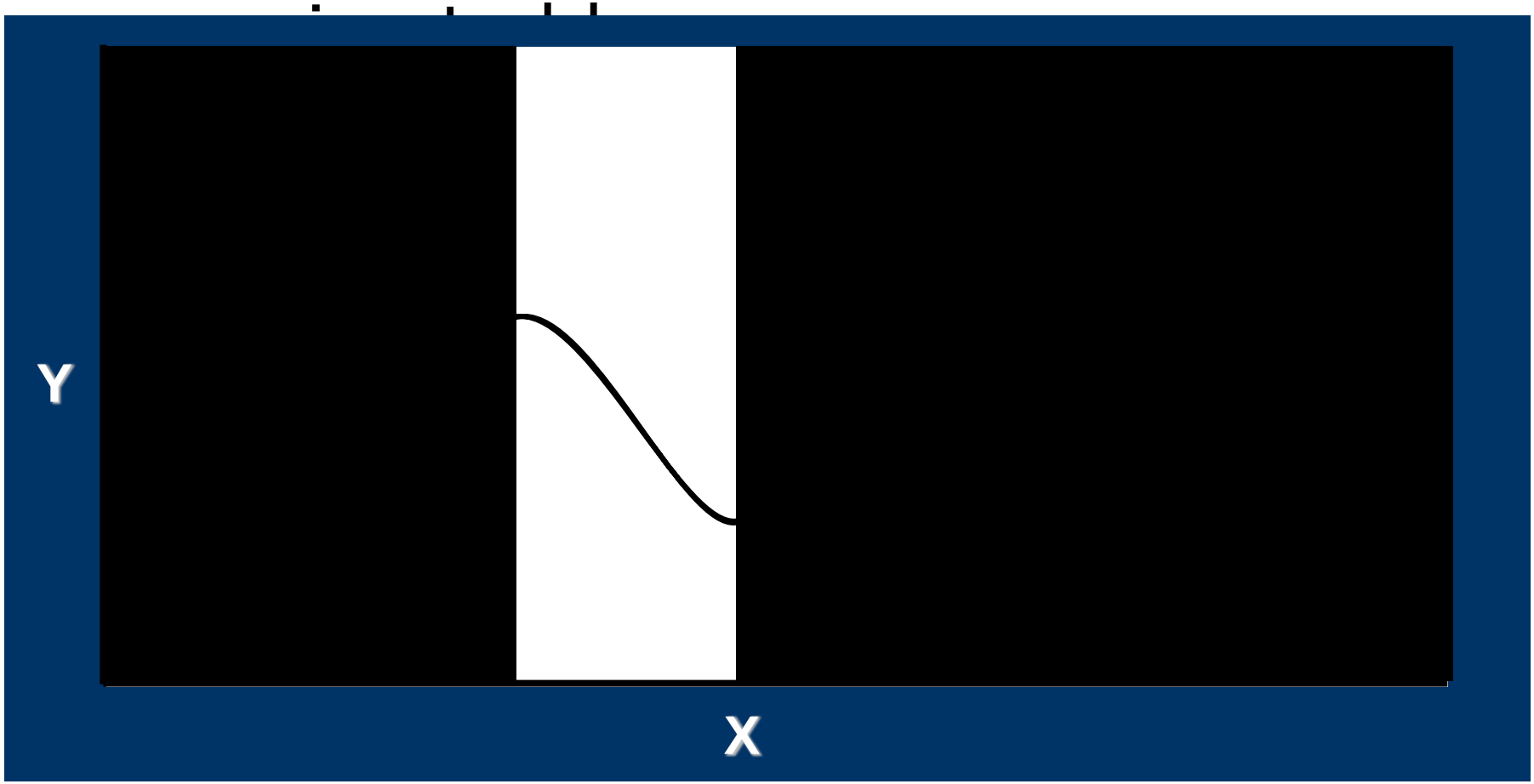
Idea of Local Regression

Locally such curves can be well



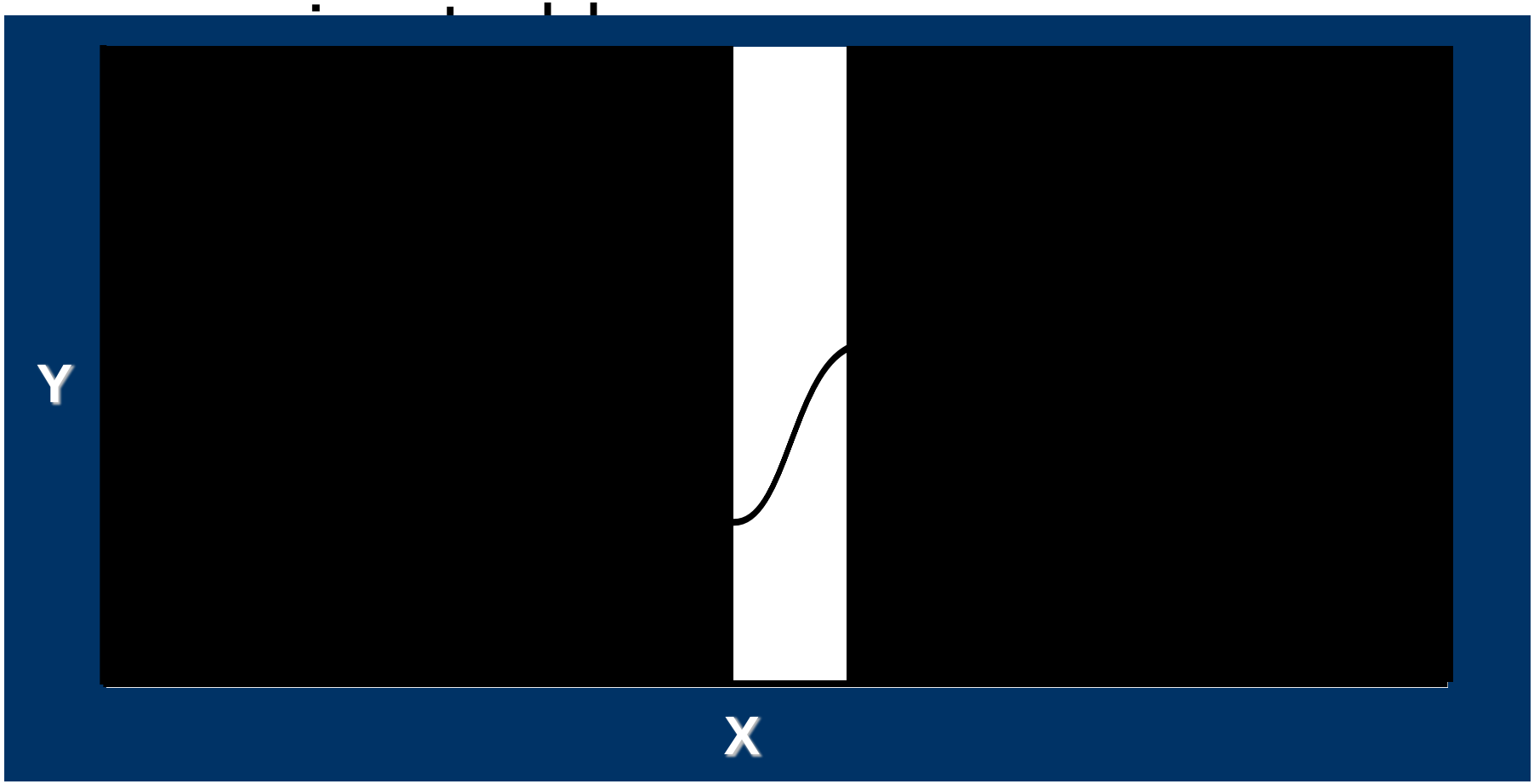
Idea of Local Regression

- Locally such curves can be well



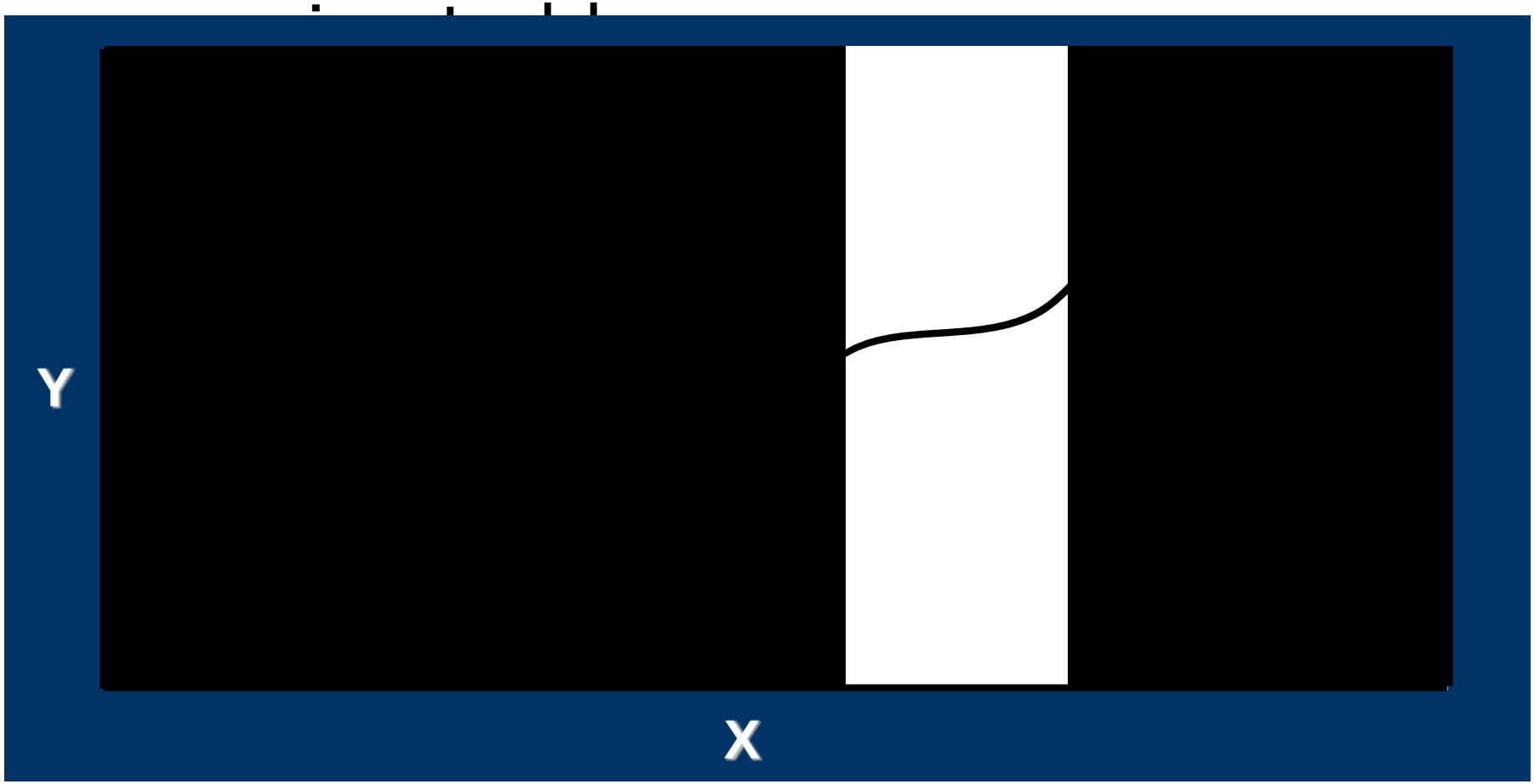
Idea of Local Regression

- Locally such curves can be well



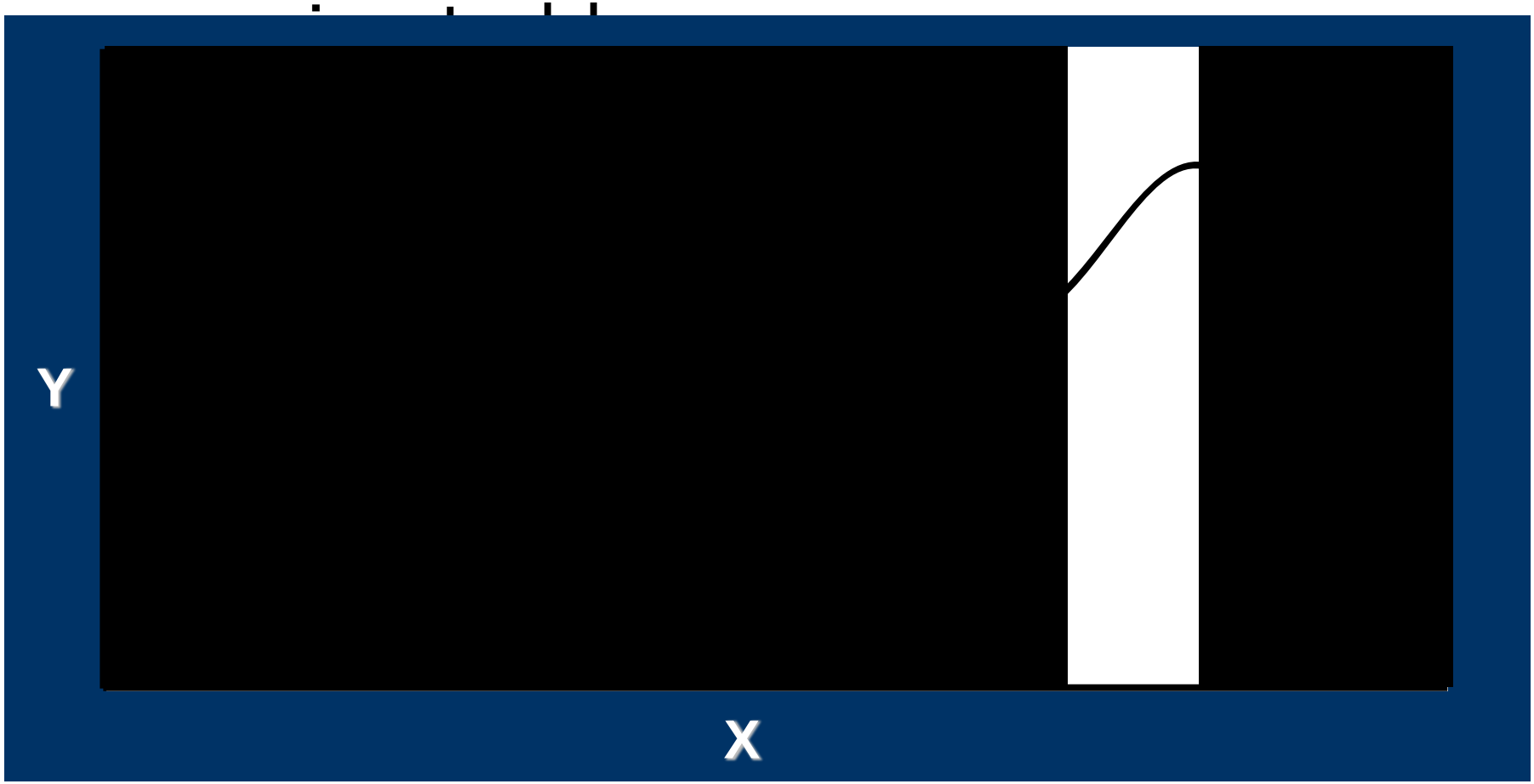
Idea of Local Regression

- Locally such curves can be well



Idea of Local Regression

- Locally such curves can be well

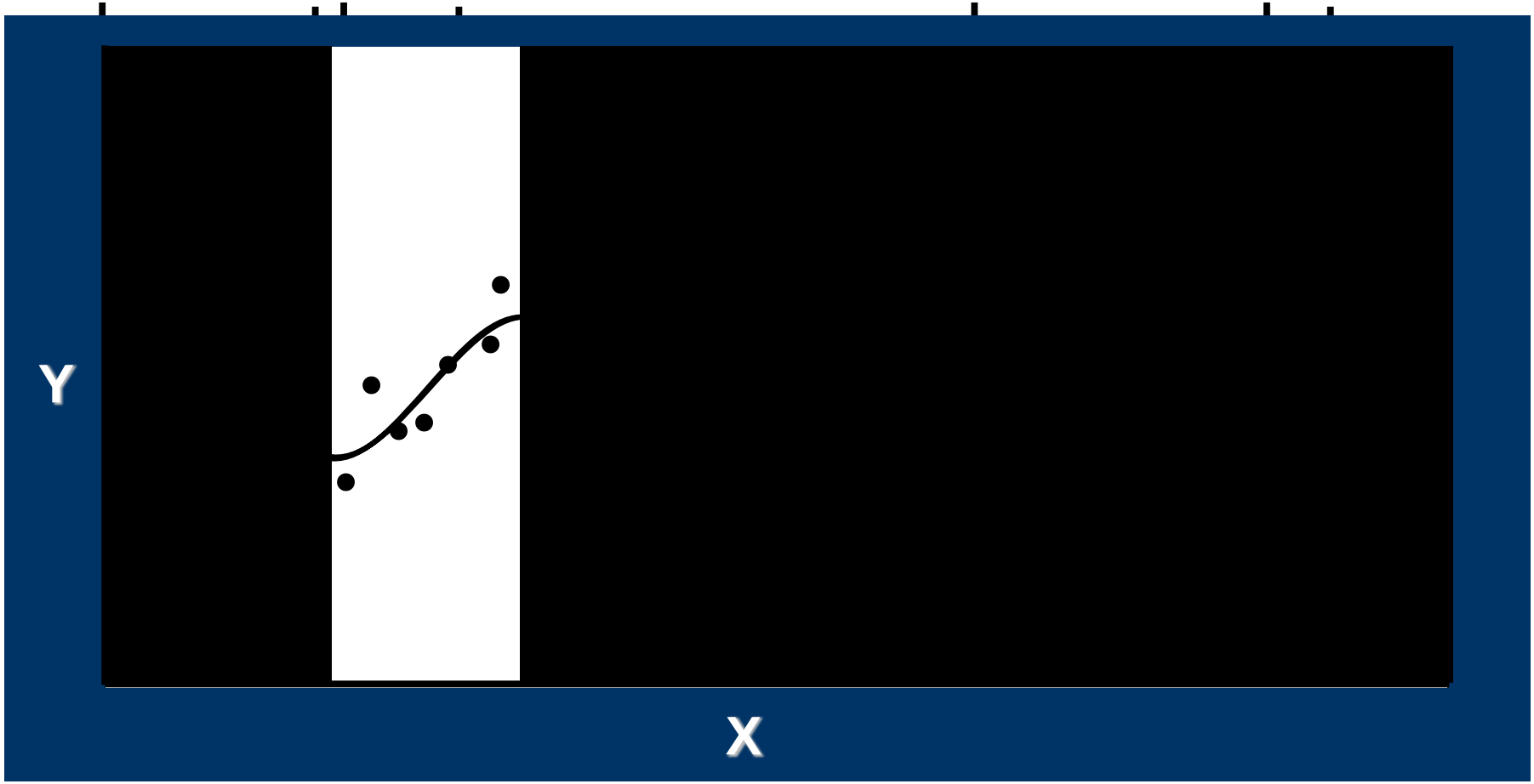


Idea of Local Regression

- Locally such curves can be well

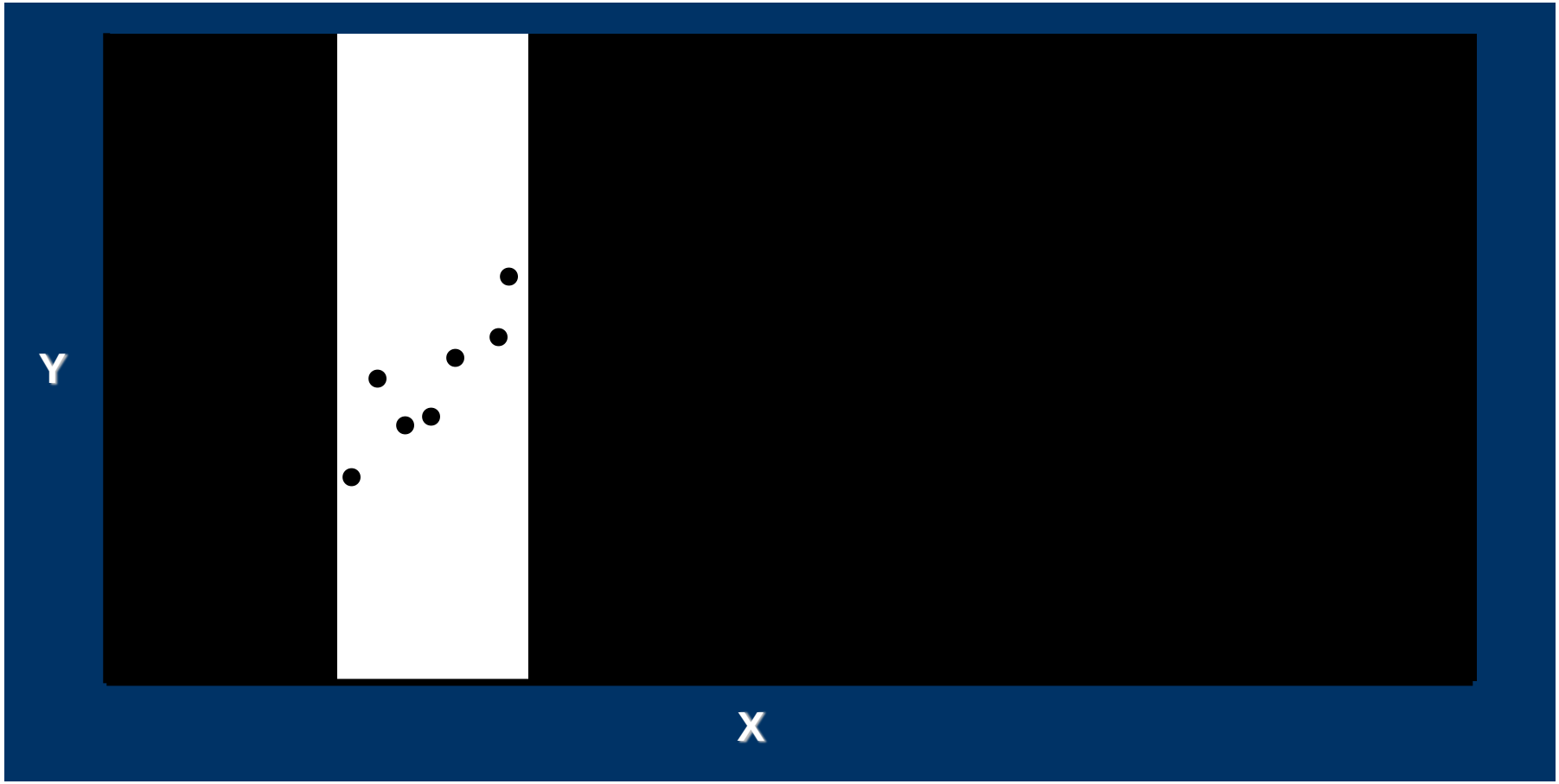
Idea of Local Regression

Consider the previous window, which



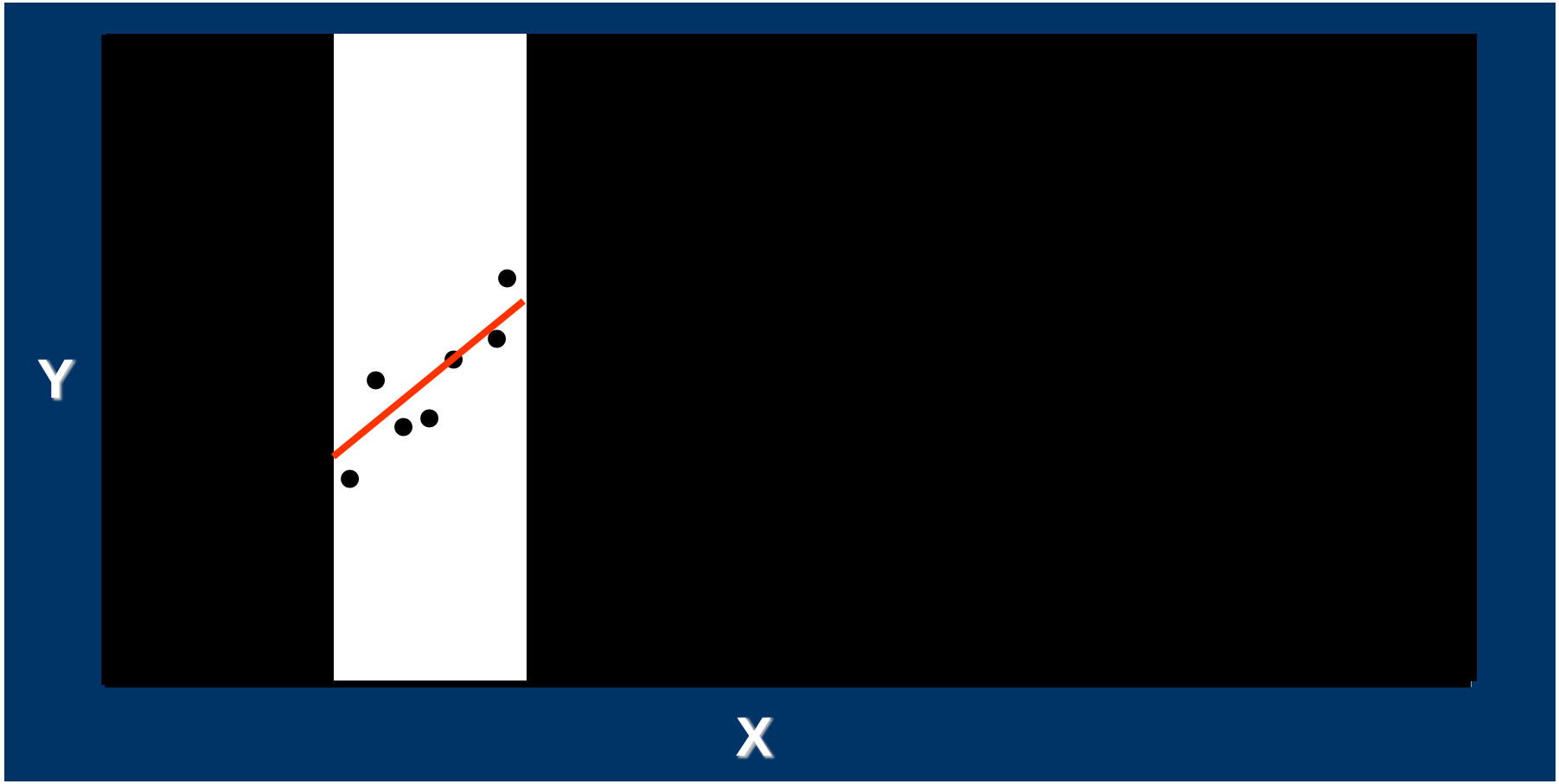
Idea of Local Regression

Just the data



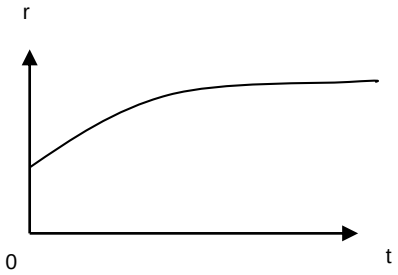
Idea of Local Regression

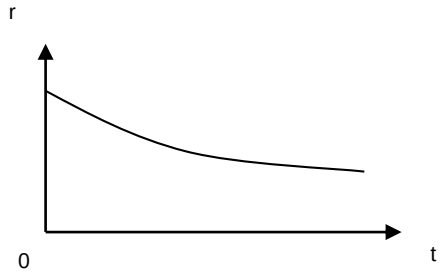
Find a local fit by linear regression.

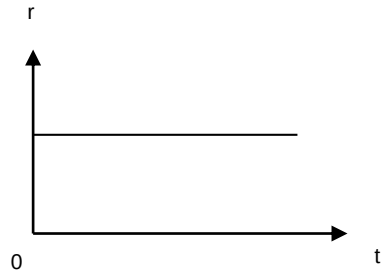


利率期限结构理论

- 一般而言收益率曲线形状主要有三种：收益率曲线是在以期限长短为横坐标，以收益率为纵坐标的直角坐标系上显示出来。主要有三种类型：第一类是正收益曲线（或称上升收益曲线），其显示的期限结构特征是短期国债收益率较低，而长期国债收益率较高。第二类是反收益曲线（或称下降收益曲线），其显示的期限结构特征是短期国债收益率较高，而长期国债收益率较低。这两种收益率曲线转换过程中会出现第三种形态的收益曲线，称水平收益曲线，其特征是长短期国债收益率基本相等。







收益率曲线的拟合方法

1.样条法

（1）多项式样条法

多项式样条法是由麦克库隆茨（Mc Culloch）于1971年提出的，它的主要思想是将贴现函数用分段的多项式函数来表示。在实际应用中，多项式样条函数的阶数一般取为三，从而保证贴现函数及其一阶和二阶导数都是连续的。于是我们用下式表示期限为t的贴现函数：

$$B(t)=\begin{cases} B_0(t)=d_0+c_0t+b_0t^2+a_0t^3, t\in[0,n] \\ B_n(t)=d_1+c_1t+b_1t^2+a_1t^3, t\in[n,m] \\ B_m(t)=d_2+c_2t+b_2t^2+a_2t^3, t\in[m,20] \end{cases}$$

收益率曲线的拟合方法

（2）指数样条法

- 指数样条法则是考虑到贴现函数基本上是一个随期限增加而指数下降的函数，它是瓦西塞克（Vasicek）和弗隆戈（Fong）在1982年提出的，该方法将贴现函数用分段的指数函数来表示。同样为了保证曲线的连续性和平滑性，通常采用三阶的指数样条函数，其形式如下：

$$B(t)=\begin{cases} B_0(t)=d_0+c_0e^{-ut}+b_0e^{-2ut}+a_0e^{-3ut}\ , & t\in[0,n] \\ B_n(t)=d_1+c_1e^{-ut}+b_1e^{-2ut}+a_1e^{-3ut}\ , & t\in[n,m] \\ B_m(t)=d_2+c_2e^{-ut}+b_2e^{-2ut}+a_2e^{-3ut}\ , & t\in[m,20] \end{cases}$$

收益率曲线的拟合方法

2.尼尔森-辛格尔（Nelson-Siegel） 模型

- 尼尔森和辛格尔在1987 年提出了一个用参数表示的瞬时 (即期限为零的) 远期利率函数。

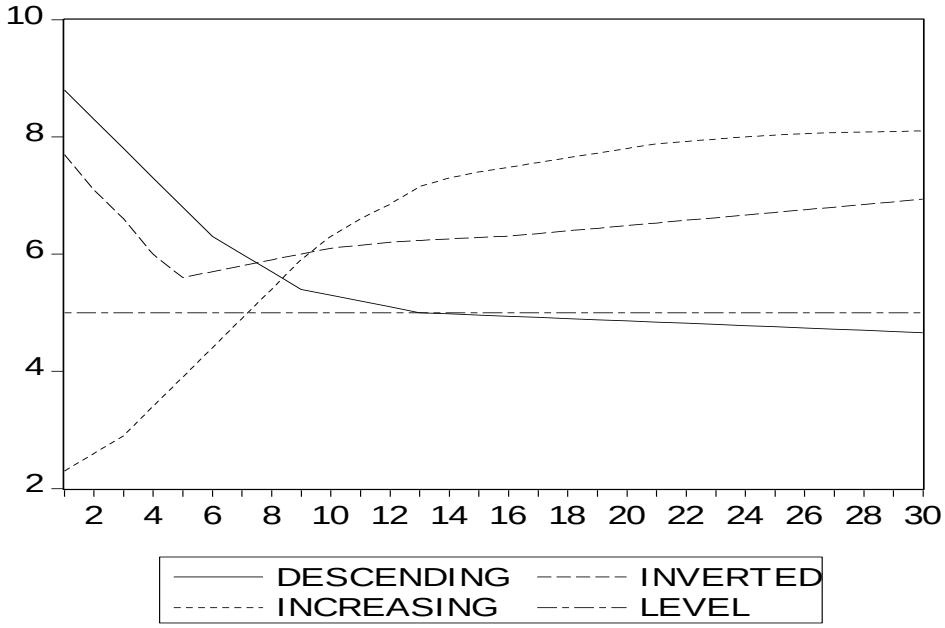
$$f(t) = \beta_0 + \beta_1 \exp\left(-\frac{t}{\tau_1}\right) + \beta_2 \left(\frac{t}{\tau_1}\right) \exp\left(-\frac{t}{\tau_1}\right)$$

- 由此我们可以求得即期利率的函数形式：

$$R(t) = \frac{\int_0^t f(s)ds}{t} = \beta_0 + \beta_1 \left[\frac{1-\exp\left(-\frac{t}{\tau_1}\right)}{\frac{t}{\tau_1}}\right] + \beta_2 \left[\frac{1-\exp\left(-\frac{t}{\tau_1}\right)}{\frac{t}{\tau_1}} - \exp\left(-\frac{t}{\tau_1}\right)\right]$$

收益率曲线的拟合方法

- 这个模型中只有四个参数，即 $\beta_0, \beta_1, \beta_2, \tau_1$ ，根据式中的即期利率，我们可以得到相应的贴现函数，从而计算债券的模型价值用以拟合市场数据。虽然参数的个数不多，但这样的函数形式已经有足够的灵活度来拟合收益率曲线的标准形状，递增的、递减的、水平和倒置的形状，如图所示。



收益率曲线的拟合方法

3. 斯文森（Svensson）模型

- 斯文森将Nelson-Siegel 模型作了推广，引进了另外两个参数 β_3, τ_2 ，而得到如下的即期利率函数：

$$R(t) = \beta_0 + \beta_1 \left[\frac{1 - \exp\left(-\frac{t}{\tau_1}\right)}{\frac{t}{\tau_1}} \right] + \beta_2 \left[\frac{1 - \exp\left(-\frac{t}{\tau_1}\right)}{\frac{t}{\tau_1}} - \exp\left(-\frac{t}{\tau_1}\right) \right]$$
$$+ \beta_3 \left[\frac{1 - \exp\left(-\frac{t}{\tau_2}\right)}{\frac{t}{\tau_2}} - \exp\left(-\frac{t}{\tau_2}\right) \right]$$

- 这个模型也被称为扩展的Nelson-Siegel 模型，这一模型在计算短期债券价格时的灵活性大大增强。

函数逼近

- 泰勒级数展开
- 平方可积函数空间
- 正交多项式

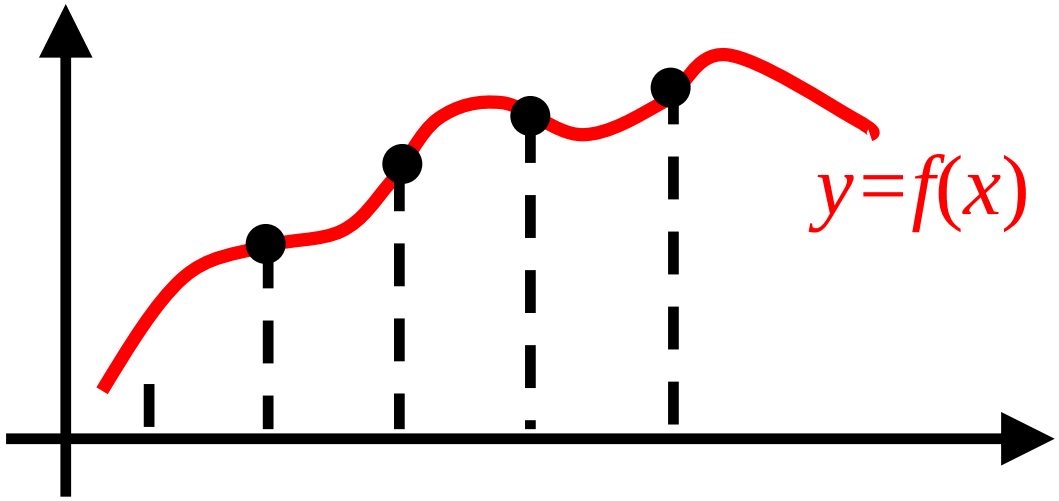
不同定义域和权重函数的正交多项式

函数空间	权函数	正交多项式	记号
$L^2[-1,1]$	1	Legendre	$P_n(x)$
$L^2[-1,1]$	$(1-x^2)^{-1/2}$(边界加重)	Chebyshev I 型	$T_n(x)$
$L^2[-1,1]$	$(1-x^2)^{1/2}$(中心加重)	Chebyshev II 型	$U_n(x)$
$L^2[0,\infty)$	$\exp(-x)$	Laguerre	$L_n(x)$
$L^2(-\infty,\infty)$	$\exp(-x^2)$	Hermite	$H_n(x)$
$L^2(-\infty,\infty)$	$\exp(-x^2/2)$	修正 Hermite	$He_n(x)$

插值（Interpolation）

- 函数解析式未知，通过实验观测得到的一组数据，即在某个区间[a, b]上给出一系列点的函数值 $y_i = f(x_i)$
- 或者给出函数表

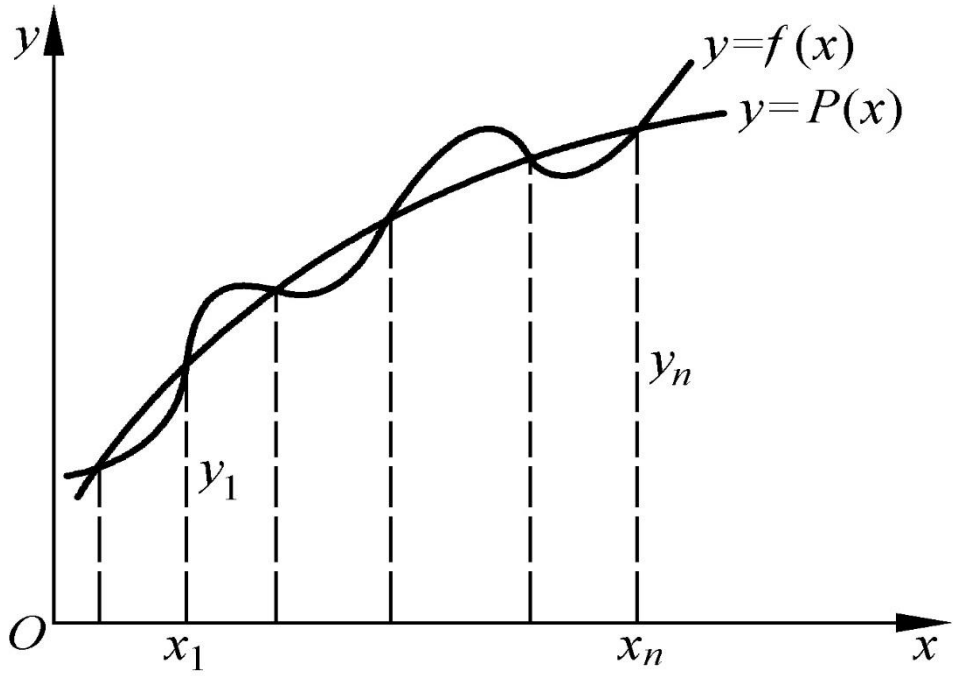
x	x_0	x_1	x_2	…	x_n
y	y_0	y_1	y_2	…	y_n



求解：$y = f(x)$ 在[a , b]上任一点处函数值的近似值？

插值

- 从几何上看，插值法就是确定曲$y = P(x)$，使其通过给定的$n+1$个点(x_i, y_i)，$i = 0,1,\cdots,n$，并用它近似已知曲线$y = f(x)$。



插值

- 插值是一种特殊的函数逼近，目的是为了补充函数缺失的部分，常用于补缺（imputation）。

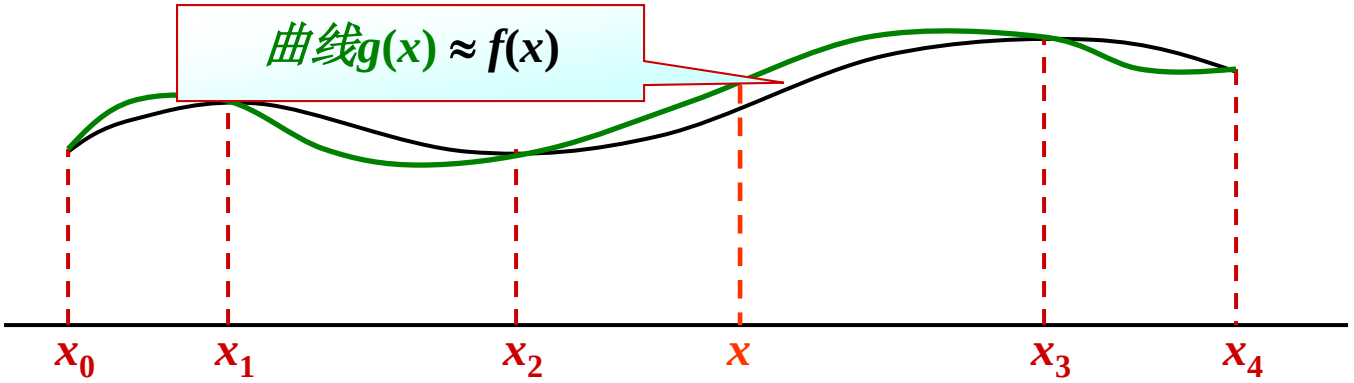
n	$\cdots$	20	$\cdots$	29	30	40	60	120	∞
$h(n)$	$\cdots$	4.35	$\cdots$	4.18	4.17	4.08	4.00	3.92	3.84

- 线性插值
- 抛物线插值
- 多项式插值
- 样条插值

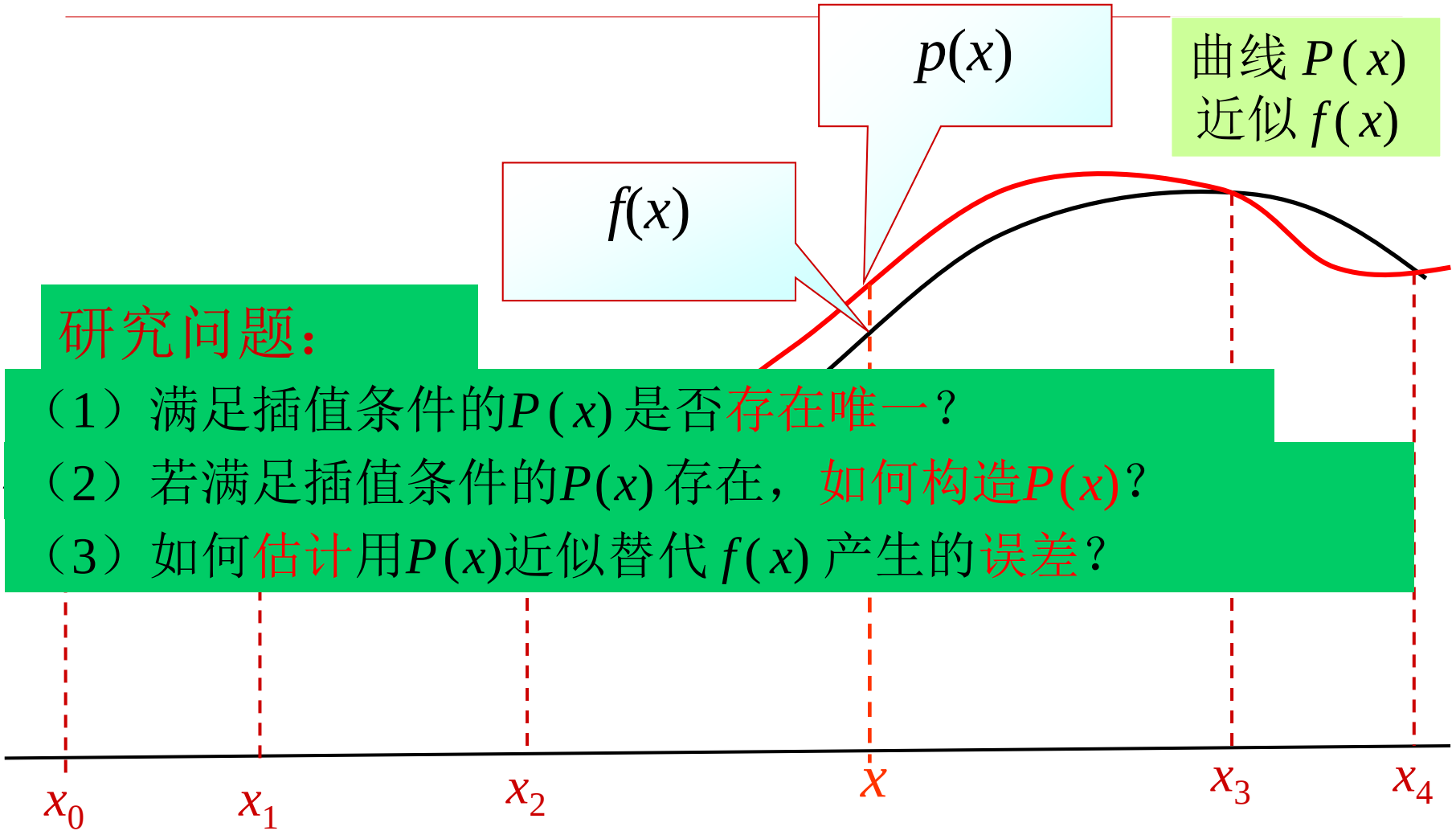
插值

从几何上看

当精确函数 $y = f(x)$ 非常复杂或未知时，在一系列节点 $x_0 \ldots x_n$ 处测得函数值 $y_0 = f(x_0), \ldots y_n = f(x_n)$，由此构造一个简单易算的近似函数 $g(x) \approx f(x)$，满足条件 $g(x_i) = f(x_i)$ $(i = 0, \ldots n)$。这里的 $g(x)$ 称为 $f(x)$ 的插值函数。最常用的插值函数是 …? 多项式



插值


p(x)
曲线 P(x)
近似 f(x)
f(x)
研究问题：
（1）满足插值条件的P(x)是否存在唯一？
（2）若满足插值条件的P(x)存在，如何构造P(x)？
（3）如何估计用P(x)近似替代 f(x) 产生的误差？
x0
x1
x2
x
x3
x4

数值积分

- 积分、最优化、矩阵计算都是在统计问题中最常见的计算问题，在统计计算中经常需要计算积分。

- 比如，从密度 $p(x)$ 计算分布函数 $F(x)$，如果没有解析表达式和精确的计算公式，需要用积分来计算：

$$F(x) = \int_{-\infty}^{x} p(u)du$$

用积分给出部分函数值后可以用插值和函数逼近得到 $F(x)$ 的近似公式。

- 已知联合密度 $p(\boldsymbol{x}_1, \boldsymbol{x}_2)$ 要求边缘密度 $p(\boldsymbol{x}_1)$，要用积分计算

$$p(\boldsymbol{x}_1) = \int p(\boldsymbol{x}_1, \boldsymbol{x}_2)d\boldsymbol{x}_2.$$

- 贝叶斯分析的主要问题是已知先验密度 $\pi(\theta)$ 和似然函数 $p(\boldsymbol{x}|\theta)$ 后求后验密度 $p(\theta|\boldsymbol{x})$:

$$p(\theta|\boldsymbol{x}) = \frac{p(\theta, \boldsymbol{x})}{p(\boldsymbol{x})} = \frac{\pi(\theta)p(\boldsymbol{x}|\theta)}{\int \pi(u)p(\boldsymbol{x}|u)du}$$

- 大多数情况下不能得到后验密度 $p(\theta|\boldsymbol{x})$ 的解析表达式，也可能需要计算积分，用后验密度求期望、平均损失函数也需要计算积分。

数值积分

积分$I=\int_a^b f(x)dx$只要找到被积函数$f(x)$原函数$F(x)$，便有牛顿—莱布尼兹(Newton—Leibniz)公式

$$\int_a^b f(x)dx = F(b)-F(a)$$

实际困难：大量的被积函数（$\frac{\sin x}{x}$, $\sin x^2$ 等），比如超越函数(Transcendental Functions), 找不到用初等函数表示的原函数；另外，$f(x)$是（测量或数值计算出的）一张数据表时，牛顿—莱布尼兹公式也不能直接运用。

数值积分

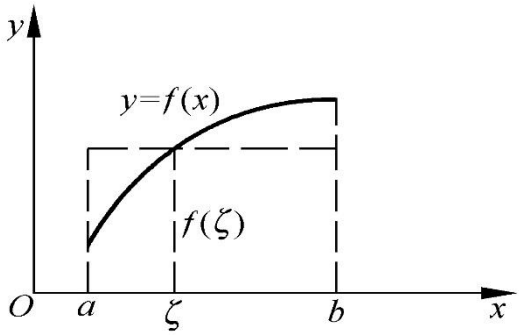
积分中值定理：在$[a, b]$内存在一点ξ，有

$$\int_a^b f(x)\,dx = (b-a)f(\xi)$$

成立。

就是说, 底为$b-a$ 而高为$f(\xi)$的矩形面积恰等于所求曲边梯形的面积 。

问题：主要在于点ξ的具体位置一般是不知道的，因而难以准确算出 $f(\xi)$的值。

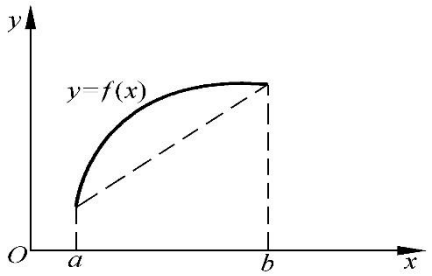


数值积分

将$f(\xi)$称为区间$[a,b]$上的平均高度。这样，只要对平均高度$f(\xi)$提供一种算法，相应地便获得一种数值求积方法。

如果用两端点的“高度”$f(a)$与$f(b)$的算术平均作为平均高度$f(\xi)$的近似值，这样导出的求积公式：

$$\int_a^b f(x)\,dx \approx \frac{b-a}{2}[f(a)+f(b)]$$



便是我们所熟悉的梯形公式 。

用区间中点$c=\frac{a+b}{2}$的“高度” $f(c)$近似地取代平均高度$f(\xi)$，则又可导出所谓中矩形公式(简称矩形公式)：

$$\int_a^b f(x)\,dx \approx (b-a)f\left(\frac{a+b}{2}\right)$$

数值积分

一般地，可以在区间$[a,b]$上适当选取某些节点x_k，然后用$f(x_k)$加权平均得到平均高度$f(\xi)$的近似值，这样构造出的求积公式具有下列形式：

$$\int_a^b f(x)dx \approx \sum_{k=0}^{n} A_k f(x_k)$$

式中x_k称为**求积节点**； A_k称为**求积系数**，亦称伴随节点 x_k的**权**。权A_k仅仅与节点x_k的选取有关，而不依赖于被积函数$f(x)$的具体形式。

这类数值积分方法通常称为机械求积，其特点是将积分求值问题归结为函数值的计算，这就避开了牛顿–莱布尼兹公式需要寻求原函数的困难。

数值积分

- 数值积分的最简单方法是直接用达布(Darboux)和计算。
- 更精确的积分方法是对被积函数进行多项式逼近然后对近似多项式用代数方法求积分。
- 这些近似多项式的形式可以不依赖于被积函数，只需要用被积函数的若干值。
- 多项式逼近可以是在全积分区间上进行，也可以把积分区间分为很多小区间在小区间上逼近。分为小区间的方法适用性更好。

数值积分

近似计算 $I=\int_a^b f(x)dx \approx \int_a^b P_n(x)dx$

思路 利用插值多项式 $P_n(x)\approx f(x)$ 则积分易算。

☞ 在$[a, b]$上取 $a\le x_0<x_1<\ldots<x_n\le b$，做 f 的 n 次插值多项式 $L_n(x)=\sum_{k=0}^{n} f(x_k)l_k(x)$ 即得到

$$\int_a^b f(x)dx \approx \sum_{k=0}^{n} f(x_k)\int_a^b l_k(x)dx$$

A_k

$$A_k=\int_a^b \prod_{j\neq k}\frac{(x-x_j)}{(x_k-x_j)}dx$$

由节点决定，与 $f(x)$ 无关。

计算达布和的积分方法

- 求定积分

$$I = \int_a^b f(x)dx, \tag{4.28}$$

- 把区间 $[a,b]$ 均匀分为 n 段，分点为 $x_0 = a, x_1, \ldots, x_{n-1}, x_n = b$，间隔为 $h = (b-a)/n$。
- 可以用

$$D_n = \sum_{i=1}^{n} f(x_i)h \tag{4.29}$$

近似计算 I。

- 把 D_n 改写成

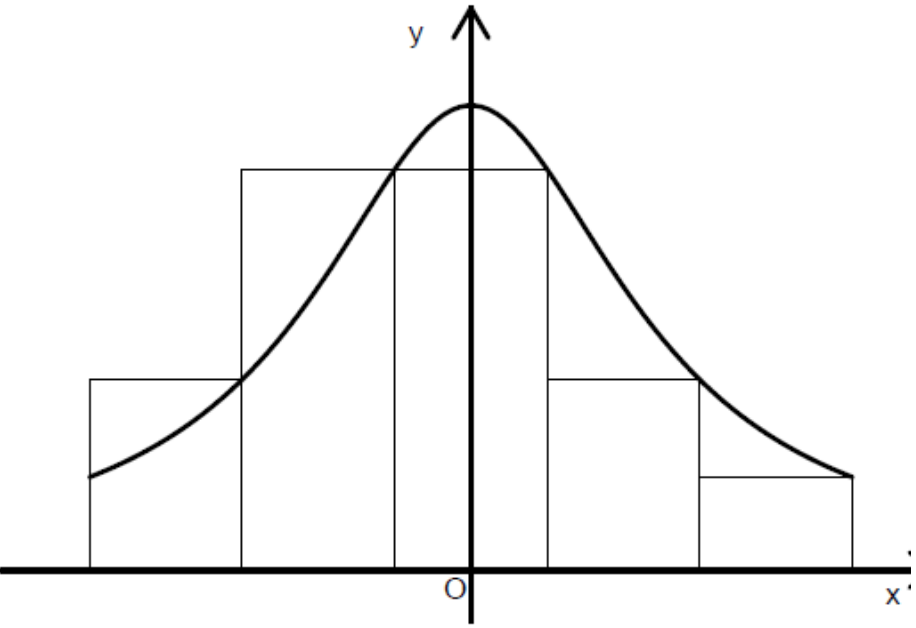
$$D_n = \sum_{i=0}^{n-1} f(x_{i+1})h$$

可以看出相当于把曲边梯形面积

$$\int_{x_i}^{x_{i+1}} f(x)dx \tag{4.30}$$

用以 $[x_i, x_{i+1}]$ 为底、右侧高度 $f(x_{i+1})$ 为高的矩形面积近似 (见下图)。

达布和数值积分图示



中点法则

- 上面的方法容易理解，但用区间端点近似整个区间的函数值误差较大，精度不好，基本不使用公式 (4.29) 计算一元函数积分。
- 如果使用区间中点作为代表，公式变成

$$M_n = h\sum_{i=0}^{n-1} f\left(a+\left(i+\frac{1}{2}\right)h\right) \quad (h=\frac{b-a}{n}) \tag{4.31}$$

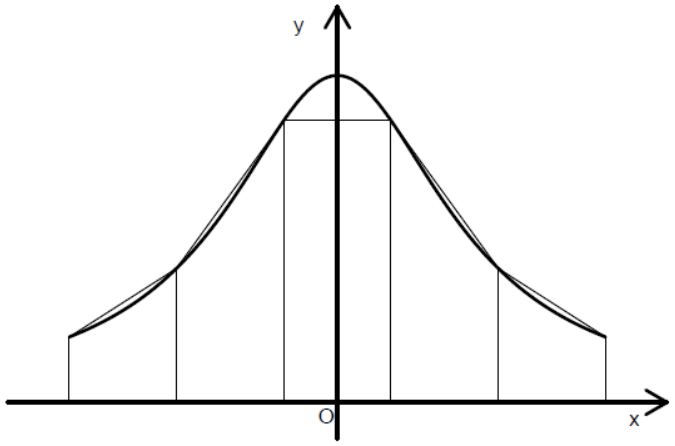
这个公式称为中点法则。

- 余项为

$$R_n = I - M_n = \frac{(b-a)^3}{24n^2} f''(\xi),\ \xi \in [a,b], \tag{4.32}$$

当 $f''(x)$ 有界时中点法则的精度为 $O(n^{-2})$，精度比 (4.29) 高得多，与下面的梯形法则精度相近。

梯形法则



- 在小区间 $[x_i, x_{i+1}]$ 内不是用常数而是用线性插值代替 $f(x)$，即用梯形代替曲边梯形 (见上图)。
- 积分公式

$$f(x) \approx f(x_i) + \frac{f(x_{i+1}) - f(x_i)}{h}(x - x_i),\ x \in [x_i, x_{i+1}]$$

$$\int_{x_i}^{x_{i+1}} f(x)dx \approx \frac{f(x_i) + f(x_{i+1})}{2} h \tag{4.33}$$

$$\int_a^b f(x)\,dx = \sum_{i=0}^{n-1} \int_{x_i}^{x_{i+1}} f(x)\,dx$$

$$\approx \frac{h}{2}\left\{ f(a) + 2\sum_{i=1}^{n-1} f(x_i) + f(b) \right\} \triangleq T_n \tag{4.34}$$

- 余项为

$$R_n = I - T_n = -\frac{(b-a)^3}{12n^2} f''(\xi), \xi \in (a, b). \tag{4.35}$$

- (4.34) 叫做复合梯形公式，在 $f''(x)$ 有界时算法精度为 $O(n^{-2})$。

辛普森 (Simpson) 法则

- 在等距的 (x_{-1}, x_0, x_1) 的区间中用抛物线插值公式近似 $f(x)$:

$$f(x) \approx \frac{1}{2}\left(f(x_{-1}) - 2f(x_0) + f(x_1)\right)\left(\frac{x - x_0}{h/2}\right)^2 + \frac{1}{2}\left(f(x_1) - f(x_{-1})\right)\left(\frac{x - x_0}{h/2}\right) + f(x_0) \tag{4.36}$$

- 其中 $h = x_1 - x_{-1} = 2(x_0 - x_{-1})$。
- 积分得

$$\int_{x_{-1}}^{x_1} f(x)\,dx = \frac{h}{6}\left\{f(x_{-1}) + 4f(x_0) + f(x_1)\right\}, h = x_1 - x_{-1}. \tag{4.37}$$

- 把区间 $[a, b]$ 等分为 n 份，记 $h = (b-a)/n$，记 $x_i = a + ih$，$i = 0, 1, 2, \ldots, n$，则 $x_0, x_1, \ldots, x_n$ 把 $[a, b]$ 等分为 n 份。
- 在每个小区间内取中点，记为 $x_{i+\frac{1}{2}} = a + (i + \frac{1}{2})h$，$i = 0, 1, \ldots, n-1$。
- 在 $[x_i, x_{i+1}]$ 内用公式 (4.37) 积分并把 n 个小区间的积分相加，得

$$I = \int_a^b f(x)\,dx \approx \frac{h}{6}\left(f(a) + 4\sum_{i=0}^{n-1} f(x_{i+\frac{1}{2}}) + 2\sum_{i=1}^{n-1} f(x_i) + f(b)\right) \triangleq S_n, \tag{4.38}$$

- 余项为

$$R_n = I - S_n = -\frac{(b-a)^5}{2880n^4} f^{(4)}(\xi), \xi \in (a, b). \tag{4.3}$$

- (4.38) 叫做复合辛普森公式，在 $f^{(4)}(x)$ 有界时复合辛普森公式误差为 $O(n^{-4})$。

牛顿-柯蒂斯(Ne

- 梯形公式和辛普森公式分别是在小区间用线性插值和抛物线插值近似被积函数 $f(x)$ 得到的积分公式。
- 一般地，对 $n+1$ 个等距节点 $(x_0 = a, x_1, \ldots, x_n = b)$ 可以进行 Lagrange 插值，得到 n 阶插值多项式 $P_n(x)$:

$$P_n(x) = \sum_{j=0}^{n} f(x_j) \frac{\prod_{k \neq j}(x - x_k)}{\prod_{k \neq j}(x_j - x_k)}$$

- 用 $P_n(x)$ 在 $[a, b]$ 上的积分近似 $\int_a^b f(x)dx$，有

$$\int_a^b f(x)\,dx \approx \int_a^b P_n(x)\,dx \tag{4.40}$$

$$= \sum_{j=0}^{n} \left\{ \int_a^b \frac{\prod_{k \neq j}(x - x_k)}{\prod_{k \neq j}(x_j - x_k)} dx \right\} f(x_j) \tag{4.41}$$

- 注意到 $x_j = a + jh$, $h = (b-a)/n$, 做变量替换 $x = a + th$, 上式变成

$$\int_a^b f(x)\,dx \approx \frac{b-a}{n} \sum_{j=0}^{n} \left\{ \int_0^n \frac{\prod_{k \neq j}(t - k)}{\prod_{k \neq j}(j - k)} dt \right\} f(x_j)$$

$$= (b-a) \sum_{j=0}^{n} C_j^{(n)} f(x_j) \tag{4.42}$$

- 这是被积函数在节点上函数 $f(x_i)$ 值的线性组合，其中各线性组合系数 $C_j^{(n)}$ 不依赖于 f 和积分区间:

$$C_j^{(n)} = \frac{1}{n} \int_0^n \frac{\prod_{k \neq j}(t - k)}{\prod_{k \neq j}(j - k)} dt = \frac{1}{n} \frac{(-1)^{n-j}}{j!(n-j)!} \int_0^n \prod_{k \neq j} (t - k) dt. \tag{4.43}$$

数值积分

- 高斯-勒让德**(Gauss-Legendre)** 积分
- Gauss-Hermite 积分
- 变步长积分

对一般的函数，提高插值多项式阶数并不能改善积分精度；

复合方法如复合梯形公式、复合辛普森公式则取点越多精度越高。

因为很难预估需要的点数，所以我们可以逐步增加取点个数直至达到需要的积分精度。

可以每次取点仅增加原来小区间的中点。

- 高维积分（运算量增长很快）

随机模拟计算的基本思路

1.针对实际问题建立一个简单且便于实现的概率统计模型，使所求的量（或解）恰好是该模型某个指标的概率分布或者数字特征。

2.对模型中的随机变量建立抽样方法，在计算机上进行模拟测试，抽取足够多的随机数，对有关事件进行统计

3.对模拟试验结果加以分析，给出所求解的估计及其精度(方差)的估计

4.必要时，还应改进模型以降低估计方差和减少试验费用，提高模拟计算的效率

数值积分问题

计算积分 $\alpha = \int_0^1 f(x)dx = Ef(X)$

我们可以将此积分看成 $f(x)$ 的数学期望。其中 $X \sim U[0,1]$ (均匀分布)。于是可以将上式积分看作是f(X)的数学期望. 若$\{U_k, 1 \le k \le n\}$为$i.i.d.U_k \sim U[0,1]$.

则可以取$\alpha_n = \frac{1}{n}\sum_{i=1}^{n} f(U_i)$作为$\alpha$的估计，

由大数定律，可以保证收敛性，即：

$$\alpha_n \to \alpha \quad with\ probability\ as\ n \to \infty$$

这表明可以用随机模拟的方法计算积分。

➢ 收敛性

- 由前面介绍可知，蒙特卡罗方法是由随机变量X的简单子样X_1，X_2，…，X_N的算术平均值：

$$\overline{X}_N = \frac{1}{N}\sum_{i=1}^{N} X_i$$

- 作为所求解的近似值。由大数定律可知，
- 如X_1，X_2，…，X_N独立同分布，且具有有限期望值（$E(X)<\infty$），则

$$P\left(\lim_{N\to\infty} \overline{X}_N = E(X)\right) = 1$$

- 即随机变量X的简单子样的算术平均值 $\overline{X}_N$ ，当子样数N充分大时，以概率1收敛于它的期望值E(X)。

➢ 误差

- 蒙特卡罗方法的近似值与真值的误差问题，概率论的中心极限定理给出了答案。该定理指出，如果随机变量序列X_1，X_2，…，X_N独立同分布，且具有有限非零的方差σ^2，即

$$0 \neq \sigma^2 = \int (x - E(X))^2 f(x) dx < \infty$$

- $f(X)$是X的分布密度函数。则

$$\lim_{N \to \infty} P\left(\frac{\sqrt{N}}{\sigma} \left| \overline{X}_N - E(X) \right| < x \right) = \frac{1}{\sqrt{2\pi}} \int_{-x}^{x} e^{-t^2/2} dt$$

- 当N充分大时，有如下的近似式

$$P\left(\left|\overline{X}_N - E(X)\right| < \frac{\lambda_\alpha \sigma}{\sqrt{N}}\right) \approx \frac{2}{\sqrt{2\pi}} \int_0^{\lambda_\alpha} e^{-t^2/2} dt = 1-\alpha$$

- 其中α称为置信度，1－α称为置信水平。

- 这表明，不等式 $\left|\overline{X}_N - E(X)\right| < \frac{\lambda_\alpha \sigma}{\sqrt{N}}$ 近似地以概率1－α成立，且误差收敛速度的阶为 $O(N^{-1/2})$。

- 通常，蒙特卡罗方法的误差ε定义为

$$\varepsilon = \frac{\lambda_\alpha \sigma}{\sqrt{N}}$$

- 上式中λ_α 与置信度α是一一对应的，根据问题的要求确定出置信水平后，就可以确定出 λ_α。

- 下面给出几个常用的α与的数值：

α	0.5	0.05	0.003
λ_α	0.6745	1.96	3

- 关于蒙特卡罗方法的误差需说明两点：第一，蒙特卡罗方法的误差为概率误差，这与其他数值计算方法是有区别的。第二，误差中的均方差σ是未知的，必须使用其估计值

$$\hat{\sigma} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} X_i^2 - (\frac{1}{N}\sum_{i=1}^{N} X_i)^2}$$

- 来代替，在计算所求量的同时，可计算出 $\hat{\sigma}$ 。

Monte Carlo积分的优点

与一般的数值积分方法比较，Monte Carlo方法具有以下优点：

1. 一般的数值方法很难推广到高维积分的情形，而Monte Carlo方法很容易推广到高维情形

2. 一般的数值积分方法的收敛阶为 $O(n^{-2/d})$，而由中心极限定理可以保证 Monte Carlo 方法的收敛阶为 $O(n^{-1/2})$ 。此收敛阶与维数无关，且在高维时明显优于一般的数值方法。

随机投点法

设函数$h(x)$ 在有限区间$[a,b]$ 上定义且有界，不妨设$0 \le h(x) \le M$。要计算$I = \int_a^b h(x)dx$，相当于计算曲线下的区域$D = \{(x,y) : 0 \le y \le h(x), x \in C = [a,b]\}$ 的面积。为此在$G = [a,b] \times (0,M)$ 上均匀抽样N次，得随机点$Z_1, Z_2 \ldots, Z_N$，$Z_i = (X_i, Y_i), i = 1,2,\ldots,N$。令

$$\xi_i = \begin{cases} 1, & Z_i \in D \\ 0, & \text{otherwise} \end{cases}, \quad i = 1,\ldots,N$$

则$\{\xi_i\}$ 是独立重复试验结果，$\{\xi_i\}$ 独立同 b(1,p) 分布，

$$p = P(Z_i \in D) = V(D)/V(G) = I/[M(b-a)] \tag{11.1}$$

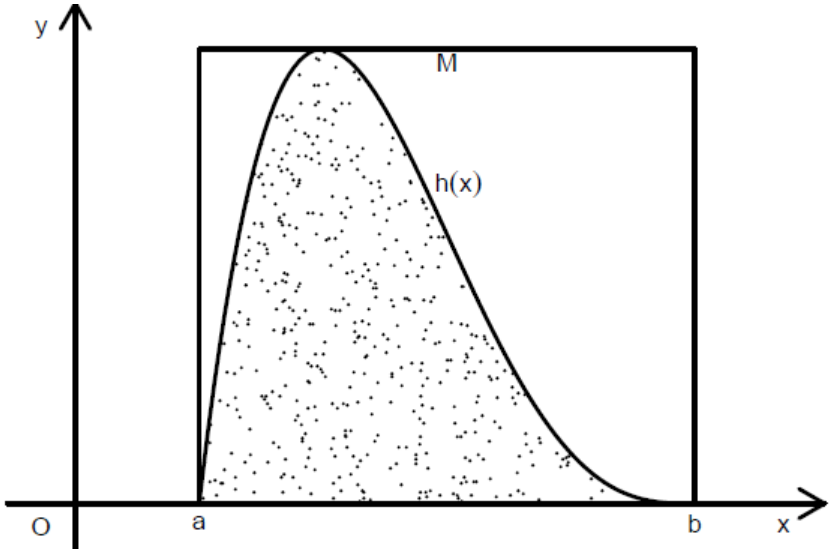
其中$V(\cdot)$ 表示区域面积。

从模拟产生的随机样本$Z_1, Z_2 \ldots, Z_N$，可以用这N 个点中落入曲线下方区域D 的百分比$\hat{p}$ 来估计(11.1)中的概率p，然后由$I = pM(b-a)$ 得到定积分I 的近似值

$$\hat{I} = \hat{p}M(b-a) \tag{11.2}$$

这种方法叫做**随机投点法**。这样计算的定积分有随机性，误差中包含了随机模拟误差。

随机投点法



由强大数律可知

$$\hat{p} = \frac{\sum \xi_i}{N} \to p, \text{ a.s. } (N \to \infty)$$

$$\hat{I} = \hat{p} M(b-a) \to pM(b-a) = I, \text{ a.s. } (N \to \infty)$$

即$N \to \infty$ 时精度可以无限地提高（当然，在计算机中要受到数值精度的限制）。

那么，提高精度需要多大的代价呢？由中心极限定理可知

$$\sqrt{N}(\hat{p}-p)/\sqrt{p(1-p)} \xrightarrow{\mathrm{d}} \mathrm{N}(0,1), \ (N \to \infty),$$

从而

$$\sqrt{N}(\hat{I}-I) = M(b-a)(\hat{p}-p) \xrightarrow{\mathrm{d}} \mathrm{N}\big(0, [M(b-a)]^2 p(1-p)\big) \tag{11.3}$$

当N 很大时$\hat{I}$ 近似服从 $\mathrm{N}(I, [M(b-a)]^2 p(1-p)/N)$ 分布，称此近似分布的方差$[M(b-a)]^2 p(1-p)/N$ 为$\hat{I}$ 的**渐近方差**。计算渐近方差可以用$\hat{p}$ 代替p 估计为 $[M(b-a)]^2 \hat{p}(1-\hat{p})/N$。式(11.3)说明 $\hat{I}$ 的误差为$O_p(\frac{1}{\sqrt{N}})$, 这样，计算$\hat{I}$ 的精度每增加一位小数，计算量需要增加 100 倍。随机模拟积分一般都服从这样的规律。

均值法

- 随机投点法容易理解，但是效率较低。
- 另一种效率更高的方法是利用期望值的估计。
- 为了计算有限区间 $[a,b]$ 上的定积分 $I=\int_a^b h(x)\,dx$, 取 $U\sim \mathrm{U}(a,b)$。
- 则

$$E[h(U)]=\int_a^b h(u)\frac{1}{b-a}du=\frac{I}{b-a}$$

$$I=(b-a)\cdot Eh(U)$$

- 若取 $\{U_i,i=1,\ldots,N\}$ 独立同 $\mathrm{U}(a,b)$ 分布, 则 $Y_i=h(U_i),i=1,2,\ldots,N$ 是 iid 随机变量列, 由强大数律,

$$\bar{Y}=\frac{1}{N}\sum_{i=1}^{N}h(U_i)\to Eh(U)=\frac{I}{b-a},\quad \text{a.s. } (N\to\infty)$$

- 于是

$$\tilde{I}=\frac{b-a}{N}\sum_{i=1}^{N}h(U_i) \tag{3.5}$$

是 I 的强相合估计。

- 称这样计算定积分 I 的方法为**平均值法**。

误差和

定义估计的根均方误差为

$$\mathrm{RMSE} = \sqrt{E(\hat{\theta} - \theta)^2} \tag{11.8}$$

在以上用独立同分布样本均值估计期望值$\theta = EY$ 的做法中，显然

$$\mathrm{RMSE} = \sqrt{\mathrm{Var}(\hat{\theta})} = \sqrt{\mathrm{Var}(\bar{Y}_n)} = \frac{\sigma}{\sqrt{N}}, \tag{11.9}$$

所以根均方误差可以用$\frac{S_N}{\sqrt{N}}$ 估计。

定义估计的平均绝对误差为

$$\mathrm{MAE} = E|\hat{\theta} - \theta| \tag{11.10}$$

当N 充分大时由$\hat{\theta}$ 的近似正态分布式(11.7)可知

$$\begin{aligned} \mathrm{MAE} =& E|\hat{\theta} - \theta| = \frac{\sigma}{\sqrt{N}} E\left|\frac{\hat{\theta} - \theta}{\sigma/\sqrt{N}}\right| \\ \approx& \sqrt{\frac{2}{\pi}} \frac{\sigma}{\sqrt{N}} \approx 0.7979 \frac{\sigma}{\sqrt{N}} \end{aligned} \tag{11.11}$$

所以平均绝对误差可以用$0.80\frac{S_N}{\sqrt{N}} = 0.80\mathrm{RMSE}$ 估计。

定义估计的平均相对误差为

$$\mathrm{MRE} = E\left|\frac{\hat{\theta} - \theta}{\theta}\right| = \mathrm{MAE}/\theta \tag{11.12}$$

所以平均相对误差可以用$0.80\frac{S_N}{\sqrt{N}\theta} = 0.80\frac{\mathrm{RMSE}}{\theta}$ 估计。平均相对误差为 0.005 相当于估计结果有两位有效数字，平均相对误差为 0.0005 相当于估计结果有三位有效数字。

误差和样本量分析

python™

样本量计算

- 为了计算样本量 N 需要取多大才能控制估计的根均方误差小于 σ_0，可以先取较小的 N_0，抽取 N_0 个样本值计算出 $S_{N_0}^2$，用 $S_{N_0}^2$ 估计总体方差 σ^2，然后求需要的 N 的大小：

$$\frac{S_{N_0}}{\sqrt{N}} < \sigma_0, \quad N > \frac{S_{N_0}^2}{\sigma_0^2}.$$

- 这是适用于平均值法的公式。

参数的近似 95% 置信区间

- 用 $\hat{\theta} = \bar{Y}_N$ 估计 $\theta = EY$ 时，可以利用近似正态分布式(3.3)计算 θ 的近似 95% 置信区间：

$$\hat{\theta} \pm 2\frac{S_N}{\sqrt{N}}. \tag{3.9}$$

- 随机投点法和平均值法都有局限性。
- $I = \int_C h(\boldsymbol{x})\, d\boldsymbol{x}$ 中积分区域 C 可能是任意形状的，也可能无界；
- $h(\boldsymbol{x})$ 在 C 内各处的取值大小差异可能很大，使得直接用平均值法估计 I 时，很多样本点处于 $|h(\boldsymbol{x})|$ 接近于零的地方，造成浪费，另外使得 $\hat{I}_2$ 的渐近方差 (见是（3.13）) 中的 $\mathrm{Var}(h(\boldsymbol{U}))$ 很大 ($\boldsymbol{U} \sim \mathrm{U}(C)$)。
- 为此，考虑用非均匀抽样: $|h(\boldsymbol{x})|$ 大的地方密集投点，$|h(\boldsymbol{x})|$ 小的地方稀疏投点。这样可以有效利用样本。

取$N = 10000$ 个样本点的一次模拟的结果得到$\hat{I} = 0.8754$, $S_N = 0.9409$, 根均方误差的估计为RMSE $= \frac{S_N}{\sqrt{N}} = 0.0094$, 平均绝对误差的估计为MAE $= 0.80$RMSE $= 0.0075$, 平均相对误差的估计为MRE $=$ MAE$/\hat{I} = 0.0087$, 说明结果只有约两位有效数字精度。为了达到四位小数的精度，需要平均相对误差控制在0.00005 左右, 需要解$0.8\frac{S_N}{\sqrt{N}\hat{I}} = 0.00005$, 代入得$N$ 需要约 3 亿次, 可见随机模拟方法提高精度的困难程度。

重要抽样法

- 设 $g(\boldsymbol{x}), \boldsymbol{x} \in C$ 是一个密度，其形状与 $|h(\boldsymbol{x})|$ 相近，且当 $g(\boldsymbol{x}) = 0$ 时 $h(\boldsymbol{x}) = 0$，当 $\|\boldsymbol{x}\| \to \infty$ 时 $h(\boldsymbol{x}) = o(g(\boldsymbol{x}))$。称 $g(\boldsymbol{x})$ 为**试投密度**或重要抽样密度。
- 设 $\boldsymbol{X}_i$ iid $\sim g(\boldsymbol{x})$, $i = 1, 2, \ldots, N$。
- 令

$$\eta_i = \frac{h(\boldsymbol{X}_i)}{g(\boldsymbol{X}_i)}, \ i = 1, 2, \ldots, N$$

- 则

$$E\eta_1 = \int_C \frac{h(\boldsymbol{x})}{g(\boldsymbol{x})} g(\boldsymbol{x})\, d\boldsymbol{x} = \int_C h(\boldsymbol{x})\, d\boldsymbol{x} = I$$

- 因此可以用 $\{\eta_i, i = 1, 2, \ldots, N\}$ 的样本平均值来估计 I，即

$$\hat{I}_3 = \bar{\eta} = \frac{1}{N} \sum_{i=1}^{N} \frac{h(\boldsymbol{X}_i)}{g(\boldsymbol{X}_i)}. \tag{3.14}$$

- 用式 (3.14) 估计 I 与 §2.2.4 的舍选法 II 有类似的想法，这种方法叫做**重要抽样法** (importance sampling)，是随机模拟的重要方法。

分层抽样法

用平均值法计算$\int_C h(\boldsymbol{x})\,d\boldsymbol{x}$，若$h(\boldsymbol{x})$ 在C 内取值变化范围大则估计方差较大。重要抽样法选取了与$f(x)$形状相似但是容易抽样的密度$g(\boldsymbol{x})$ 作为试投密度，大大提高了精度，但是这样的$g(\boldsymbol{x})$ 有时难以找到。

如果把C 上的积分分解为若干个子集上的积分，使得$h(\boldsymbol{x})$ 在每个子集上变化不大，分别计算各个子集上的积分再求和，可以提高估计精度。这种方法叫做**分层抽样法**。这也是抽样调查中的重要技术。

例 13.1. 对函数

$$h(x)=\begin{cases}1+\frac{x}{10}, & 0\le x\le 0.5\\ -1+\frac{x}{10}, & 0.5<x\le 1\end{cases}$$

求定积分

$$I=\int_0^1 h(x)\,dx,$$

可以得I 的精确值为$I=0.05$。我们用平均值法和分层抽样法来估计I 并比较精度。

在$(0,1)$ 区间随机抽取N 点用平均值法得$\hat{I}_2$，其渐近方差为

$$\mathrm{Var}(\hat{I}_2)=\frac{\mathrm{Var}(h(U))}{N}=\frac{143}{150N}\approx\frac{0.9533}{N}.$$

分层抽样法

理论值为 0.9533。

把I 拆分为$[0, 0.5]$ 和$[0.5, 1]$ 上的积分，即

$$I = a + b = \int_0^{0.5} h(x)\,dx + \int_{0.5}^{1} h(x)\,dx,$$

对a 和b 分别用平均值法，得

$$\begin{aligned}
\hat{a} =& \frac{0.5}{N/2}\sum_{i=1}^{N/2} h(0.5U_i) = \frac{0.5}{N/2}\sum_{i=1}^{N/2}(1 + 0.05U_i),\\
\hat{b} =& \frac{0.5}{N/2}\sum_{i=(N/2)+1}^{N} h(0.5 + 0.5U_i) = \frac{0.5}{N/2}\sum_{i=(N/2)+1}^{N}(-1 + 0.05 + 0.05U_i),\\
\hat{I}_5 =& \hat{a} + \hat{b},
\end{aligned}$$

则分层抽样法结果$\hat{I}_5$ 的渐近方差为

$$\begin{aligned}
\mathrm{Var}(\hat{I}_5) =& \mathrm{Var}(\hat{a} + \hat{b}) = \mathrm{Var}(\hat{a}) + \mathrm{Var}(\hat{b})\\
=& 0.25\frac{\mathrm{Var}(1 + 0.05U)}{N/2} + 0.25\frac{\mathrm{Var}(-0.95 + 0.05U)}{N/2} = \frac{1/4800}{N},
\end{aligned}$$

分层后的估计方差远小于不分层的结果, 可以节省样本量约 4500 倍。

分层抽样法

例 13.2. 设$U \sim \mathrm{U}(0,1)$, 用分层抽样法估计$\theta = Eh(U) = \int_0^1 h(x)\,dx$。

令$Y = \mathrm{ceiling}(mU)$, 即当且仅当$\frac{j-1}{m} < U \le \frac{j}{m}$ 时$Y = j$，$j = 1, 2, \ldots, m$, 可以按照Y 分层抽样估计θ:

$$\begin{aligned}\theta =& E[h(U)] = \sum_{j=1}^{m} E[h(U)|Y = j]\,P(Y = j)\\ =& \frac{1}{m}\sum_{j=1}^{m} E[h(U)|Y = j],\end{aligned}$$

易见$Y = j$ 条件下U 服从$(\frac{j-1}{m}, \frac{j}{m})$ 上的均匀分布，设$U_1, U_2, \ldots, U_n$ 是 U(0,1) 的独立抽样，则用分层抽样法取每层$N_j = 1$ 估计$\theta = Eh(U)$ 为

$$\hat{\theta} = \frac{1}{m}\sum_{j=1}^{m} h\left(\frac{j-1+U_j}{m}\right).$$

随机模拟方法虽然有着适用性广、方法简单的优点，但是又有精度低、计算量大的缺点，一整套模拟算几天几夜也是常有的事情。如果能成倍地减小随机模拟误差方差，就可以有效地节省随机模拟时间，有些情况下可以把耗时长到不具有可行性的模拟计算（比如几个月）缩短到可行（比如几天）。
这里关于定积分计算的重要抽样法、分层抽样法都是降低随机模拟误差方差的重要方法，也可以用在一般的模拟问题中。

MCMC

- 实际工作中经常遇到分布复杂的高维随机向量抽样问题。重要抽样法可以应付维数不太高的情况，但是对于维数很高而且分布很复杂（比如，分布密度多峰而且位置不易确定的情况）则难以处理。
- MCMC(马氏链蒙特卡洛) 是一种对高维随机向量抽样的方法，此方法模拟一个马氏链，使马氏链的平稳分布为目标分布，由此产生大量的近似服从目标分布的样本，但样本不是相互独立的。MCMC 的目标分布密度函数或概率函数可以只计算到差一个常数倍的值。MCMC 方法适用范围广，近年来获得了广泛的应用。

马尔科夫链

设 $\{X_t, t=0,1,\dots\}$ 为随机变量序列，称为一个**随机过程**。称 X_t 为“系统在时刻 t 的**状态**”。为讨论简单起见，设所有 X_t 均取值于有限集合 $S=\{1,2,\dots,m\}$, 称 S 为**状态空间**。如果 $\{X_t\}$ 满足

$$\begin{aligned} &P(X_{t+1}=j|X_0=k_0,\dots X_{t-1}=k_{t-1},X_t=i) \\ =&P(X_{t+1}=j|X_t=i)=p_{ij},\ t=0,1,\dots,\ k_0,\dots,k_{t-1},i,j\in S, \end{aligned} \tag{3.85}$$

则称 $\{X_t\}$ 为**马氏链**, p_{ij} 为**转移概率**，矩阵 $P=(p_{ij})_{m\times m}$ 为**转移概率矩阵**。显然 $\sum_{j=1}^m p_{ij}=1,\ i=1,2,\dots,m$。对马氏链，$P(X_{t+k}=j|X_t=i)\triangleq p_{ij}^{(k)}$ 也不依赖于 t，称为k **步转移概率**。如果对任意 $i,j\in S, i\neq j$ 都存在 $k\geq 1$ 使得 $p_{ij}^{(k)}>0$ 则称 $\{X_t\}$ 为**不可约马氏链**。不可约马氏链的所有状态是互相连通的，即总能经过若干步后互相转移。对马氏链 $\{X_t\}$ 的某个状态 i, 如果存在 $k\geq 0$ 使得 $p_{ii}^{(k)}>0$ 并且 $p_{ii}^{(k+1)}>0$，则称 i 是**非周期**的。如果一个马氏链所有状态都是非周期的，则该马氏链称为非周期的。不可约马氏链只要有一个状态是非周期的则所有状态是非周期的。对只有有限个状态的非周期不可约马氏链有

$$\lim_{n\to\infty} P(X_n=j)=\pi_j,\ j=1,2,\dots,m, \tag{3.86}$$

其中 $\{\pi_j, j=1,2,\dots,m\}$ 为常数, 称为 $\{X_t\}$ 的**极限分布**。$\{\pi_j\}$ 满足方程组

$$\begin{cases} \sum_{i=1}^m \pi_i p_{ij}=\pi_j,\ j=1,2,\dots,m \\ \sum_{j=1}^m \pi_j=1, \end{cases} \tag{3.87}$$

称满足(3.87)的分布 $\{\pi_j\}$ 为**平稳分布**。对只有有限个状态的非周期不可约马氏链，极限分布和平稳分布存在且为同一分布。

Metropolis-Hasting 抽样

- 设随机变量 X 分布为 $\pi(x), x \in \mathcal{X}$。为论述简单起见仍假设 $\mathcal{X}$ 是离散集合。
- 算法需要一个试转移概率函数 $T(y|x), x, y \in \mathcal{X}$，满足 $0 \leq T(y|x) \leq 1$, $\sum_y T(y|x) = 1$，并且

$$T(y|x) > 0 \Leftrightarrow T(x|y) > 0. \tag{3.20}$$

- 算法首先从 $\mathcal{X}$ 中任意取初值 $X^{(0)}$。设经过 t 步后算法的当前状态为 $X^{(t)}$, 则下一步由试转移分布 $T(y|X^{(t)})$ 抽取 Y，并生成 U ~U(0,1), 然后按如下规则转移:

$$X^{(t+1)} = \begin{cases} Y & \text{若} U \leq r(X^{(t)}, Y) \\ X^{(t)} & \text{否则} \end{cases} \tag{3.90}$$

- 其中

$$r(x, y) = \min\left\{1, \frac{\pi(y)T(x|y)}{\pi(x)T(y|x)}\right\}. \tag{3.91}$$

- 在 MH 算法中如果取 $T(y|x) = T(x|y)$，则 $r(x, y) = \min\left(1, \frac{\pi(y)}{\pi(x)}\right)$, 相应的算法称为 **Metropolis 抽样法**。
- 如果取 $T(y|x) = g(y)$(不依赖于 x)，则 $r(x, y) = \min\left(1, \frac{\pi(y)/g(y)}{\pi(x)/g(x)}\right)$, 相应的算法称为 **Metropolis 独立抽样法**, 和重要抽样有相似之处, 试抽样分布 $g(\cdot)$ 经常取为相对重尾的分布。

概率转移核的构造（一）

MCMC的方法有很多，在此我们只介绍Metropolis-Hastings方法

基本思路：

任意选择一个不可约的转移概率 $q(\bullet,\bullet)$ 以及一个函数 $\alpha(\bullet,\bullet)\ (0 \le \alpha(\bullet,\bullet) \le 1)$。对任一组合 (x, x')，定义：

$$p(x,x') = q(x,x')\alpha(x,x') \qquad x \neq x'$$

$$p(x,x) = 1 - \int_{x \neq x'} q(x,x')\alpha(x,x')dx' \qquad x = x$$

则易见 $p(x,x')$ 构成一个概率转移核。

概率转移核的构造（二）

此方法的实施比较直观：如果链在时刻 t 处于状态 x，即 $X^{(t)}=x$，则首先由 $q(\bullet|x)$ 产生一个潜在的转移 $x\to x'$，然后根据概率 $\alpha(x,x')$ 接受 x' 作为链下一时刻的状态值，而以概率 $1\text{-}\alpha(x,x')$ 拒绝转移到 x'，从而链在下一时刻仍然处于状态 x。

我们的目标是使 $\pi(x)$ 成为马氏链的平稳分布，下面我们就介绍在给定 $q(\bullet,\bullet)$ 后，如何选择 $\alpha(\bullet,\bullet)$

概率转移核的构造（三）

Metropolis-Hastings 算法：

一个常用的选择：

$$\alpha(x, x') = \min\{1, \frac{\pi(x')q(x', x)}{\pi(x)q(x, x')}\}$$

此时有：

$$p(x, x') = \begin{cases} q(x, x') & \pi(x')q(x', x) \geq \pi(x)q(x, x') \\ q(x', x)\frac{\pi(x')}{\pi(x)} & \pi(x')q(x', x) < \pi(x)q(x, x') \end{cases}$$

定理：由上述过程产生的 Markov 链是可逆的，即：

$$\pi(x)\,p(x,x') = \pi(x')\,p(x',x) \qquad (*)$$

且 $\pi(x)$ 是 Markov 链的平稳分布。

proof：若$x = x'$,则（*）式显然成立。下面设$x \neq x'$，则：

$$\begin{aligned}
\pi(x)\,p(x,x') &= \pi(x)q(x,x')\min\left\{1, \frac{\pi(x')q(x',x)}{\pi(x)q(x,x')}\right\} \\
&= \min\{\pi(x)q(x,x'), \pi(x')q(x',x)\} \\
&= \pi(x')q(x',x)\min\left\{\frac{\pi(x)q(x,x')}{\pi(x')q(x',x)}, 1\right\} \\
&= \pi(x')\,p(x',x)
\end{aligned}$$

所以（*）成立

（续上页证明）

因为（*）式成立，所以有：

$$\int \pi(x)\,p(x,x')dx = \int \pi(x')\,p(x',x)dx$$

$$= \pi(x')\int p(x',x)dx$$

$$= \pi(x')$$

（最后一个等号成立是因为 $p(x',x)$ 是一个概率核）

所以，$\pi(x)$ 是 Markov 链的平稳分布。

Metropolis-Hastings 算法的具体步骤

1.任意选取马氏链的初始状态 $X_0 = x$

2.由转移核 $q(\bullet | x)$ 产生一个尝试移动 x'

3.生成 $U(0,1)$ 随机数 u，如果 $u \le \alpha(x, x')$，则令 $X_1 = x'$，否则保持当前状态不变，即 $X_1 = X_0 = x$

4.重复上述步骤，依次生成 $X_2, X_3 \ldots\ldots X_n$

几种常用的 q(x,x′)（一）

python™

1.Metropolis选择：

Metropolis 曾经考虑对称分布，即：

$$q(x,x') = q(x',x), \qquad \forall x, x'$$

此时 $\alpha(x,x') = \min\left\{1, \frac{\pi(x')}{\pi(x)}\right\}$

对称的分布是很常用的，比如当 x 给定时，$q(x,x')$ 可以取成正态分布，它以 x 为均值，方差为常数。

几种常用的 q(x,x′)（二）

2.独立抽样：

如果 $q(x,x')$ 与当前状态x无关，即 $q(x,x')=q(x')$ 则由此分布导出的 Metropolis-Hastings 算法称为独立抽样。$\alpha(x,x')=\min\{1,\frac{\omega(x')}{\omega(x)}\}$，其中$\omega(x)=\frac{\pi(x)}{q(x)}$

一般，独立抽样的效果可能很好，也可能很不好。通常，要使独立抽样由好的效果，$q(x)$ 应该接近 $\pi(x)$。

Metropolis-Hasting 抽样-贝叶斯推断

- 在金融投资中，投资者经常把若干种证券组合在一起来减少风险。假设有 5 支股票的 $n = 250$ 个交易日的收益率记录，每个交易日都找出这 5 支股票收益率最高的一个，设 X_i 表示第 i 支股票在 n 个交易日中收益率为最高的次数 $(i = 1, 2, \ldots, 5)$。
- 设 $(X_1, \ldots, X_5)$ 服从多项分布，相应的概率假设为

$$\boldsymbol{p} = \left(\frac{1}{3}, \frac{1-\beta}{3}, \frac{1-2\beta}{3}, \frac{2\beta}{3}, \frac{\beta}{3}\right),$$

  其中 $\beta \in (0, 0.5)$ 为未知参数。
- 假设 β 有先验分布 $p_0(\beta) \sim \mathrm{U}(0, 0.5)$。

Metropolis-Hasting 抽样-贝叶斯推断

- 设 $(x_1,\ldots,x_5)$ 为 $(X_1,\ldots,X_5)$ 的观测值，则 β 的后验分布为

$$
\begin{aligned}
&f(\beta|x_1,\ldots,x_5) \propto p(x_1,\ldots,x_5|\beta)p_0(\beta)\\
=&\binom{n}{x_1,\ldots,x_5}\left(\frac{1}{3}\right)^{x_1}\left(\frac{1-\beta}{3}\right)^{x_2}\left(\frac{1-2\beta}{3}\right)^{x_3}\left(\frac{2\beta}{3}\right)^{x_4}\left(\frac{\beta}{3}\right)^{x_5}\\
&\frac{1}{0.5}I_{(0,0.5)}(\beta)\\
\propto&(1-\beta)^{x_2}(1-2\beta)^{x_3}\beta^{x_4+x_5}I_{(0,0.5)}(\beta) \triangleq \tilde{\pi}(\beta).
\end{aligned}
$$

- 为了求 β 后验均值，需要产生服从 $f(\beta|x_1,\ldots,x_5)$ 的抽样。

- 从 β 的后验分布很难直接抽样，采用 Metropolis 抽样法。
- 设当前 β 的状态为 $\beta^{(t)}$，取试抽样分布 $T(y|\beta^{(t)})$ 为 U(0,0.5)，则 $T(y|x)=T(x|y)$，

$$
\begin{aligned}
r(\beta^{(t)},y)&=\min\left(1,\frac{\tilde{\pi}(y)}{\tilde{\pi}(\beta^{(t)})}\right)\\
&=\min\left(1,\left(\frac{1-y}{1-\beta^{(t)}}\right)^{x_2}\left(\frac{1-2y}{1-2\beta^{(t)}}\right)^{x_3}\left(\frac{y}{\beta^{(t)}}\right)^{x_4+x_5}\right),
\end{aligned}
$$

- 从 U(0, 0.5) 试抽取 y，以概率 $r(\beta^{(t)},y)$ 接受 $\beta^{(t+1)}=y$ 即可。

随机游动MH 算法

MH 抽样中试转移概率函数$T(y|x)$ 较难找到，容易想到的是从$x^{(t)}$ 作随机游动的试转移方法，叫做随机游动 Metropolis 抽样。

设X 的目标分布$\pi(\boldsymbol{x})$ 取值于欧式空间$\mathcal{X} = \mathbb{R}^d$。从$\boldsymbol{x}^{(t)}$ 出发试转移，令

$$\boldsymbol{y} = \boldsymbol{x}^{(t)} + \varepsilon_t,$$

其中$\varepsilon_t \sim g(\boldsymbol{x}; \sigma)$ 对不同t 是独立同分布的, $T(\boldsymbol{y}|\boldsymbol{x}) = g(\boldsymbol{y} - \boldsymbol{x})$。设$g$ 是关于$\boldsymbol{x} = \boldsymbol{0}$ 对称的分布，则$T(\boldsymbol{y}|\boldsymbol{x}) = T(\boldsymbol{x}|\boldsymbol{y})$。

常取g 为 $\mathrm{N}(\boldsymbol{0}, \sigma^2 I)$ 和半径为σ 的中心为$\boldsymbol{0}$ 的球内的均匀分布。

转移法则为: 从$\boldsymbol{x}^{(t)}$ 出发试转移到$\boldsymbol{y}$ 后，若$\pi(\boldsymbol{y}) > \pi(\boldsymbol{x}^{(t)})$ 则令$\boldsymbol{x}^{(t+1)} = \boldsymbol{y}$; 否则, 独立地抽取$U \sim \mathrm{U}(0, 1)$, 取

$$\boldsymbol{x}^{(t+1)} = \begin{cases} \boldsymbol{y}, & \text{as } U \le \pi(\boldsymbol{y})/\pi(\boldsymbol{x}^{(t)}), \\ \boldsymbol{x}^{(t)}, & \text{otherwise.} \end{cases}$$

随机游动 MH 算法是一种 Metropolis 抽样方法。随机游动的步幅σ 是重要参数，步幅过大导致拒绝率大，步幅过小使得序列的相关性太强，收敛到平衡态速度太慢。一个建议选法是试验各种选法，使得试抽样被接受的概率在 0.25 到 0.35 之间。见 Liu (2001) § 5.4 P. 115.

Gibbs 抽样

一般的 MH 抽样每一步首先进行尝试运动，然后根据新的状态是否靠近目标分布来接受或拒绝试抽样点，所以可能会存在多次的无效尝试，效率较低。

Gibbs 抽样是另外一种 MCMC 方法，它仅在坐标轴方向尝试转移，用当前点的条件分布决定下一步的试抽样分布，所有试抽样都被接受，不需要拒绝，所以效率可以更高。

设状态用$\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ 表示, 设目标分布为$\pi(\boldsymbol{x})$, 用$\boldsymbol{x}_{(-i)}$ 表示$(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$，假设$\pi(\cdot)$ 的条件分布 $p(x_i|\boldsymbol{x}_{(-i)})$ 都能够比较容易地抽样。

Gibbs 抽样每一步从条件分布中抽样，可以轮流从每一分量抽样，这样的算法称为系统扫描 Gibbs 抽样算法:

从$\pi(\boldsymbol{x})$的取值区域任意取一个初值$\boldsymbol{X}^{(0)}$

for(t in $0 : (N-1)$){

  for(i in $1 : n$){

    从条件分布$p(x_i|X_1^*, \ldots, X_{i-1}^*, X_{i+1}^{(t)}, \ldots, X_n^{(t)})$抽取$X_i^*$

  }

  $\boldsymbol{X}^{(t+1)} \leftarrow (X_1^*, \ldots, X_n^*)$

}

Gibbs 抽样

从条件分布抽样的次序也可以是随机选取各个分量，这样的算法称为随机扫描 Gibbs 抽样算法:

从$\pi(\boldsymbol{x})$的取值区域任意取一个初值$\boldsymbol{X}^{(0)}$

for(t in $0:(N-1)$){

　　按概率$\boldsymbol{\alpha}=(\alpha_1,\ldots,\alpha_n)$随机抽取下标$i$

　　从条件分布$p(x_i|\boldsymbol{X}^{(t)}_{(-i)})$抽取$X_i^*$

　　$\boldsymbol{X}^{(t+1)}\leftarrow(X_1^{(t)},\ldots,X_{i-1}^{(t)},X_i^*,X_{i+1}^{(t)},\ldots,X_n^{(t)})$

}

其中下标的抽样概率$\boldsymbol{\alpha}$ 为事先给定。

容易看出，无论采用系统扫描还是随机扫描的 Gibbs 抽样，如果$\boldsymbol{X}^{(t)}$ 服从目标分布，则$\boldsymbol{X}^{(t+1)}$ 也服从目标分布。以系统扫描方法为例，设在第$t+1$ 步已经抽取了$X_1^*,\ldots,X_{i-1}^*$，令$\boldsymbol{Y}=(X_1^*,\ldots,X_{i-1}^*,X_i^{(t)},\ldots,X_n^{(t)})$，设$\boldsymbol{Y}\sim\pi(\cdot)$。下一步从$\pi(\cdot)$ 的边缘密度 $p(x_i|X_1^*,\ldots,X_{i-1}^*,X_{i+1}^{(t)},\ldots,X_n^{(t)})$ 抽取X_i^*，则$\boldsymbol{Y}^*=(X_1^*,\ldots,X_{i-1}^*,X_i^*,X_{i+1}^{(t)},\ldots,X_n^{(t)})$ 的分布密度在$\boldsymbol{Y}^*$ 处的值为

$$
\begin{aligned}
&p(X_1^*,\ldots,X_{i-1}^*,X_i^*,X_{i+1}^{(t)},\ldots,X_n^{(t)})\\
=&p(X_i^*|X_1^*,\ldots,X_{i-1}^*,X_{i+1}^{(t)},\ldots,X_n^{(t)})\,p(X_1^*,\ldots,X_{i-1}^*,X_{i+1}^{(t)},\ldots,X_n^{(t)})\\
=&\pi(X_1^*,\ldots,X_{i-1}^*,X_i^*,X_{i+1}^{(t)},\ldots,X_n^{(t)})
\end{aligned}
$$

即$\boldsymbol{Y}\sim\pi(\cdot)$ 则$\boldsymbol{Y}^*\sim\pi(\cdot)$。

生成二维正态分布

- 设目标分布为二元正态分布，设 $\boldsymbol{X} \sim \pi(\boldsymbol{x})$ 为

$$\mathrm{N}\left\{\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}\right\}$$

- 采用系统扫描 Gibbs 抽样方案，每一步的迭代为：

$$\text{抽取} X_1^{(t+1)} | X_2^{(t)} \sim \mathrm{N}(\rho X_2^{(t)}, 1-\rho^2)$$
$$\text{抽取} X_2^{(t+1)} | X_1^{(t+1)} \sim \mathrm{N}(\rho X_1^{(t+1)}, 1-\rho^2)$$

- 递推可得

$$\begin{pmatrix} X_1^{(t)} \\ X_2^{(t)} \end{pmatrix} \sim \mathrm{N}\left\{\begin{pmatrix} \rho^{2t-1} X_2^{(0)} \\ \rho^{2t} X_2^{(0)} \end{pmatrix}, \begin{pmatrix} 1-\rho^{4t-2} & \rho-\rho^{4t-1} \\ \rho-\rho^{4t-1} & 1-\rho^{4t} \end{pmatrix}\right\} \quad (3.22)$$

- 当 $t \to \infty$ 时，$(X_1^{(t)}, X_2^{(t)})$ 的期望与目标分布期望之差为 $O(|\rho|^{2t})$，方差与目标分布方差之差为 $O(|\rho|^{4t})$。

Beta分布

- 设目标分布为

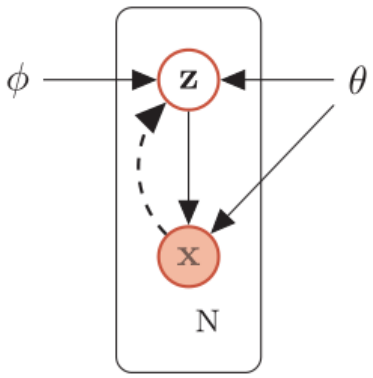
$$\pi(x,y) \propto \binom{n}{x} y^{x+\alpha-1}(1-y)^{n-x+\beta-1},\ x=0,1,\ldots,n,\ 0\le y\le 1, \tag{3.23}$$

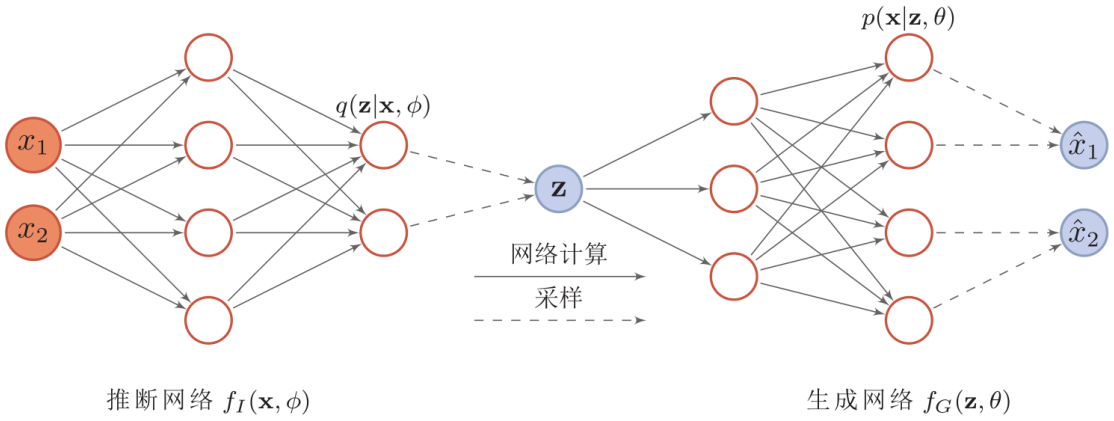
- 则 $X|Y \sim \mathrm{B}(n,y)$, $Y|X \sim \mathrm{Beta}(x+\alpha, n-x+\beta)$。
- 易见 Y 的边缘分布为 $\mathrm{Beta}(\alpha,\beta)$。可以用 Gibbs 抽样方法模拟生成 (X,Y) 的样本链。

变分自编码器(VAE)

用神经网络来替代

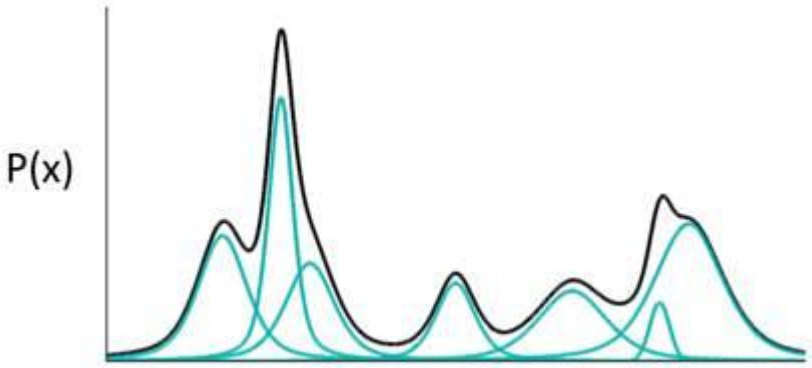
- 变分自编码器的模型结构可以分为两个部分：
  - 寻找后验分布p(z|x;θ)的变分近似q(z|x;φ*)；
    - 变分推断：用简单的分布q去近似复杂的分p(z|x;θ)
  - 在已知q(z|x;φ*)的情况下，估计更好的分布p(x|z;θ)。





变分自编码器（VAE）

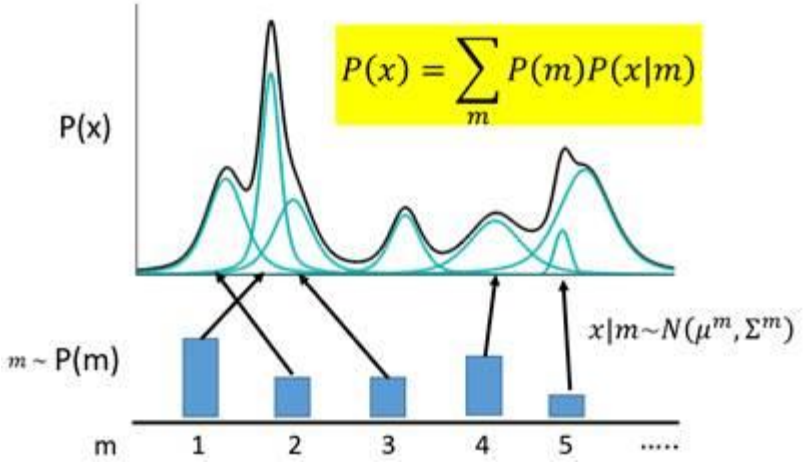
一个任意复杂数据的分布



任何一个数据的分布，都可以看作是若干正态（高斯）分布的叠加。就是高斯混合模型GMM。这种拆分方法已经证明出，当拆分的数量达到512时，其叠加的分布相对于原始分布而言，误差是非常非常小的了。可以利用这一理论模型去考虑如何给数据进行编码。一种最直接的思路是，直接用每一组高斯分布的参数作为一个编码值实现编码。

Auto-Encoding Variational Bayes

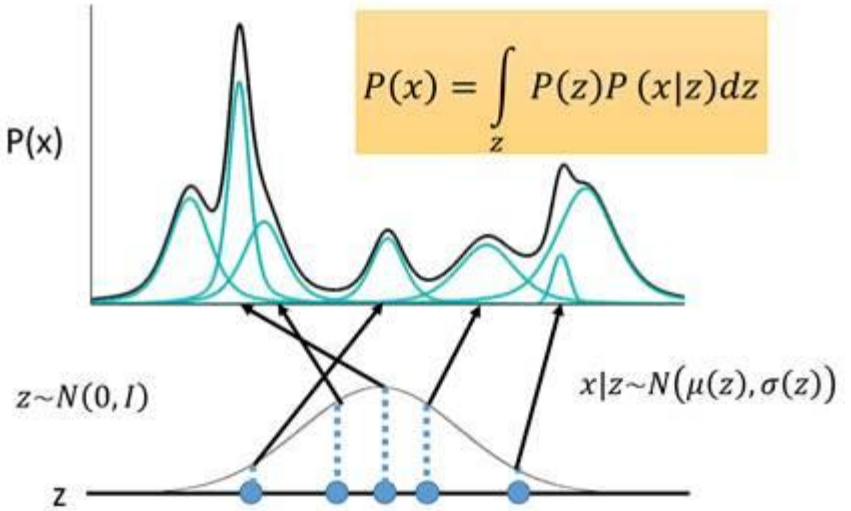
变分自编码器（VAE）



m代表着编码维度上的编号，譬如实现一个512维的编码，m的取值范围就是1,2,3.....512。m会服从于一个概率分布P(m)（多项式分布）。现在编码的对应关系是，每采样一个m，其对应到一个小的高斯分布$x|m \sim N(\mu^m, \Sigma^m)$，P(X)就可以等价为所有的这些高斯分布的叠加，即：

$$P(x) = \sum_m P(m)P(x|m)，其中 m \sim P(m)$$

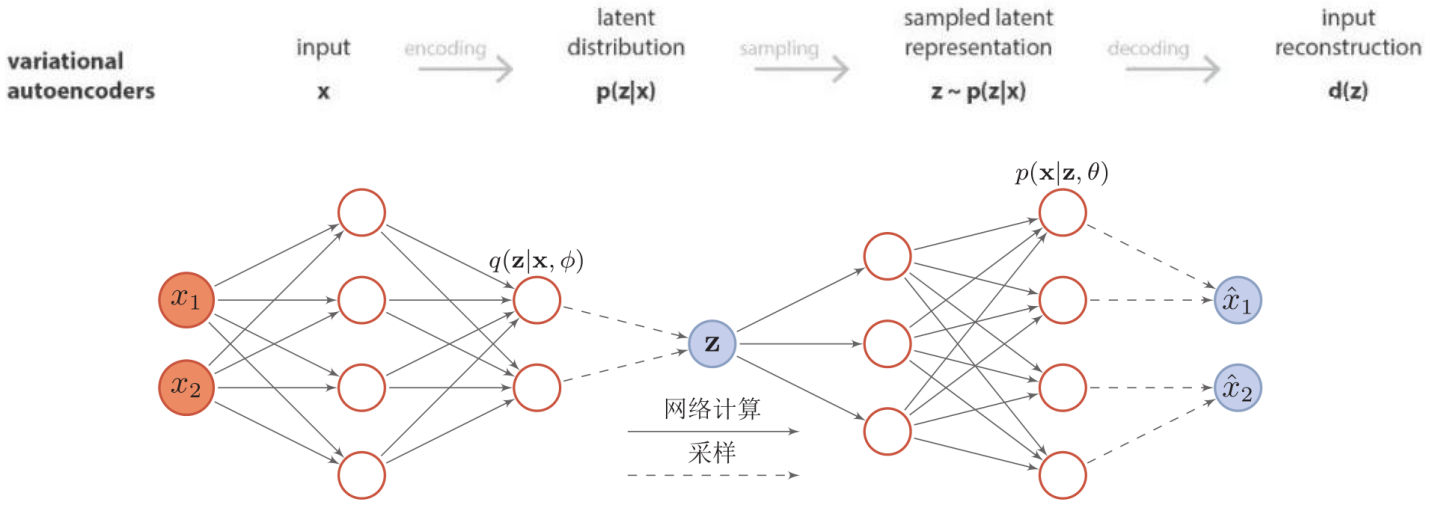
这种编码方式是离散的、有大量的丢失区域。

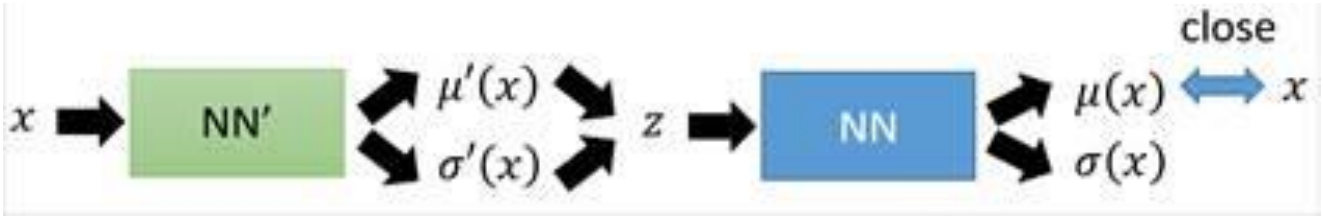


现在编码换成一个连续变量z，我们规定z服从正态分布 （实际上并不一定要选用$N(0, I)$，其他的连续分布都是可行的）。每对于一个采样z，会有两个函数$\mu(z)$和$\sigma(z)$，分别决定z对应到的高斯分布的均值和方差，然后在积分域上所有的高斯分布的累加就成为了原始分布P(X),即：

$$P(x) = \int_z P(z)P(x|z)dz，其中 z \sim N(0, I)，x|z \sim N(\mu(z), \sigma(z))$$

变分自编码器（VAE）


variational autoencoders
input
x
encoding
latent distribution
p(z|x)
sampling
sampled latent representation
z ~ p(z|x)
decoding
input reconstruction
d(z)
$p(\mathbf{x}|\mathbf{z},\theta)$
$q(\mathbf{z}|\mathbf{x},\phi)$
x_1
x_2
$\mathbf{z}$
$\hat{x}_1$
$\hat{x}_2$
网络计算
采样
推断网络 $f_I(\mathbf{x},\phi)$
生成网络 $f_G(\mathbf{z},\theta)$

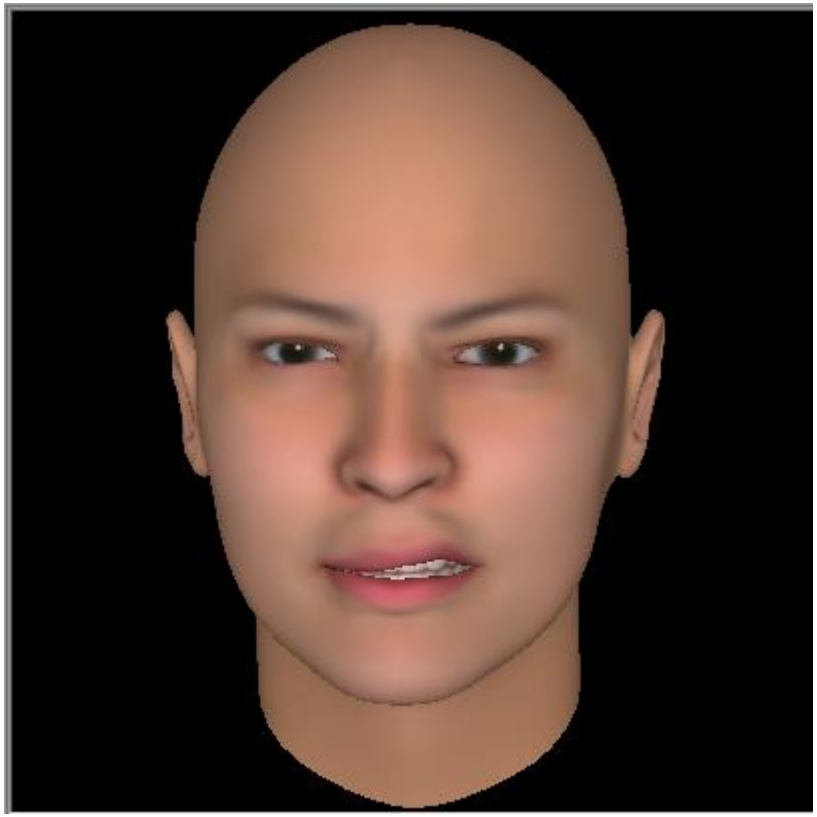

x
NN'
$\mu'(x)$
$\sigma'(x)$
z
NN
$\mu(x)$
$\sigma(x)$
close
x

六种基本表情

- 愤怒(anger)

六种基本表情

- 厌恶(disgust)

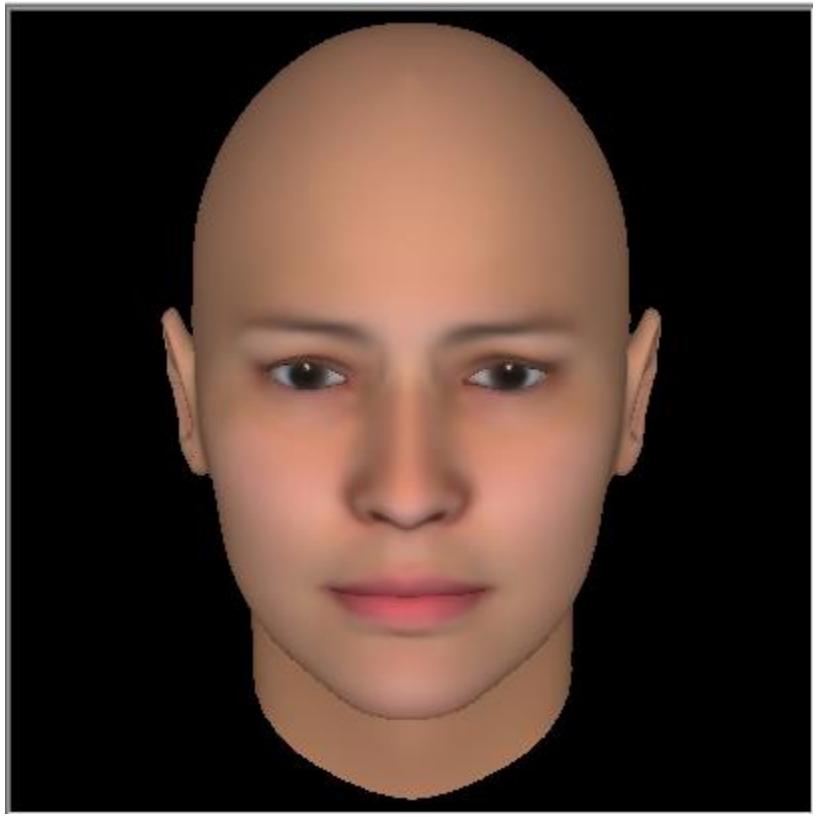


六种基本表情

- 恐惧(fear)

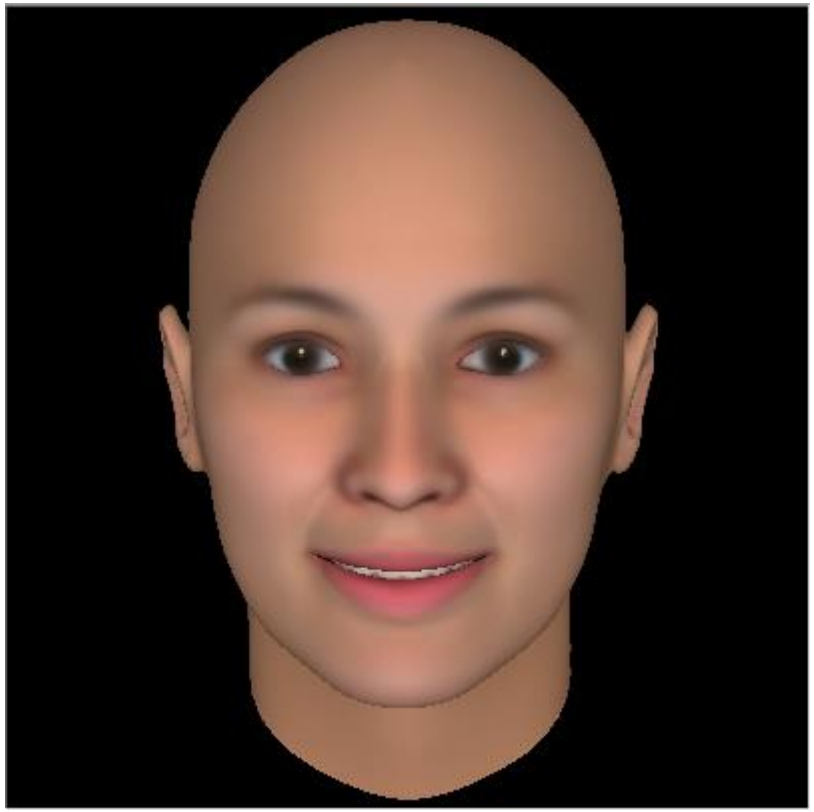
六种基本表情

- 悲伤(sad)



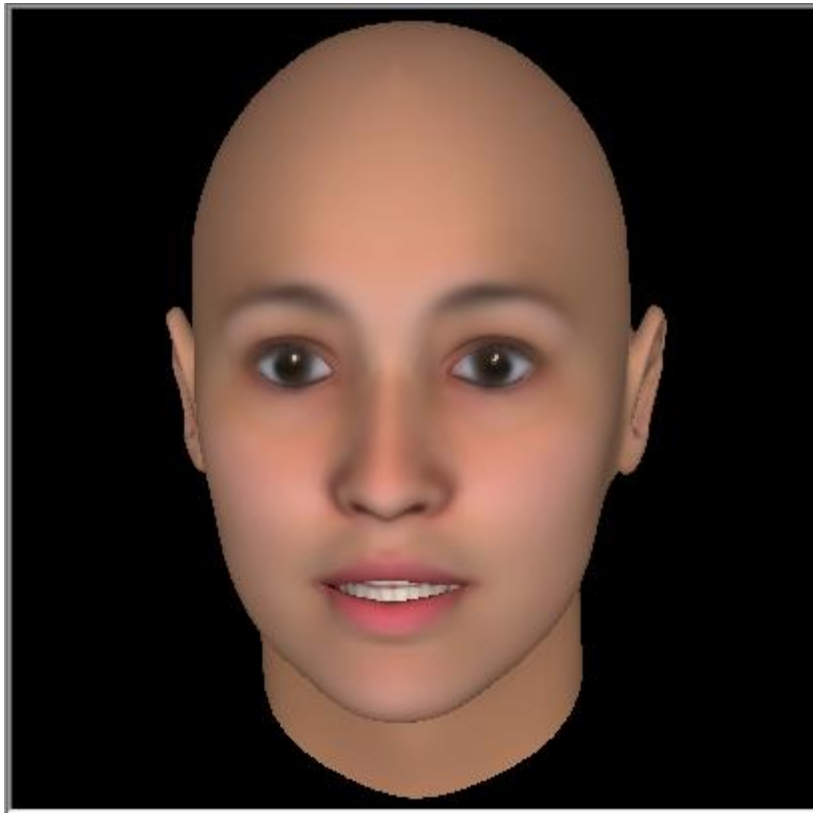
六种基本表情

- 高兴(smile)

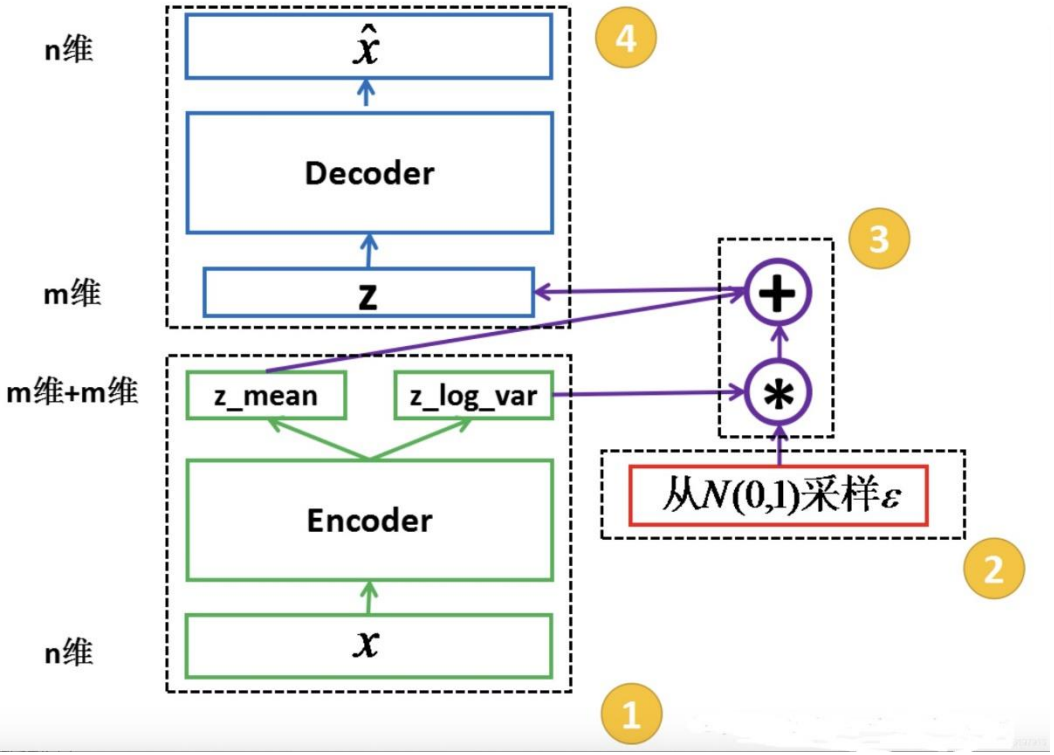


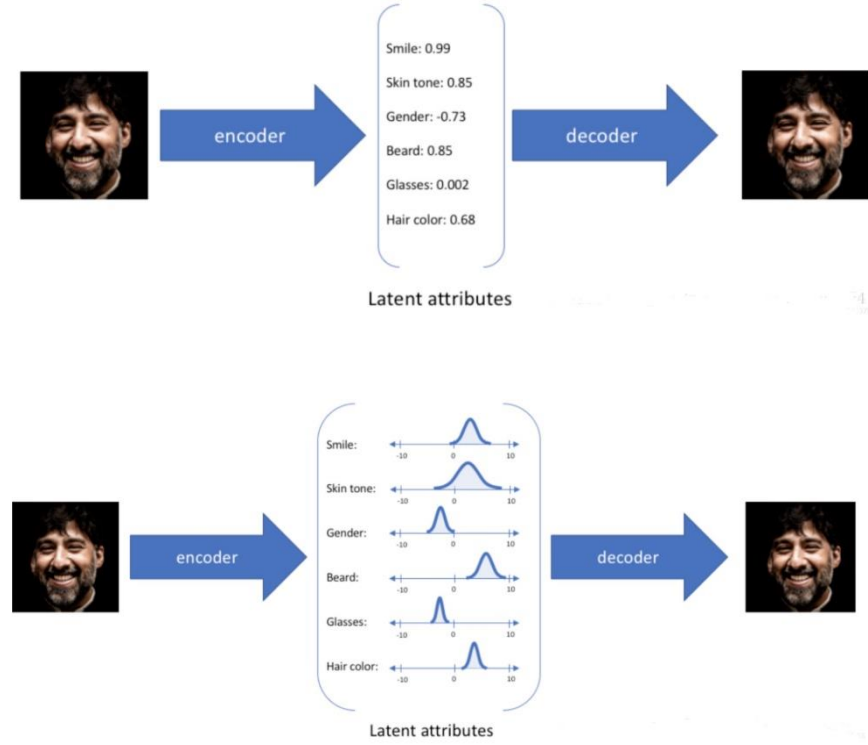
六种基本表情

- 惊奇(surprise)



VAE


n维
$\hat{x}$
4
Decoder
m维
z
3
m维+m维
z_mean
z_log_var
Encoder
从N(0,1)采样ε
2
n维
x
1


Smile: 0.99
Skin tone: 0.85
Gender: -0.73
Beard: 0.85
Glasses: 0.002
Hair color: 0.68
encoder
decoder
Latent attributes
Smile:
Skin tone:
Gender:
Beard:
Glasses:
Hair color:
encoder
decoder
Latent attributes

VAE

https://github.com/pytorch/examples/tree/master/vae

https://github.com/taichuai/d2l_zh_tensorflow2.0/blob/master/VAE%E5%AE%9E%E6%88%98.ipynb

python™

谢谢Q/A