机器学习

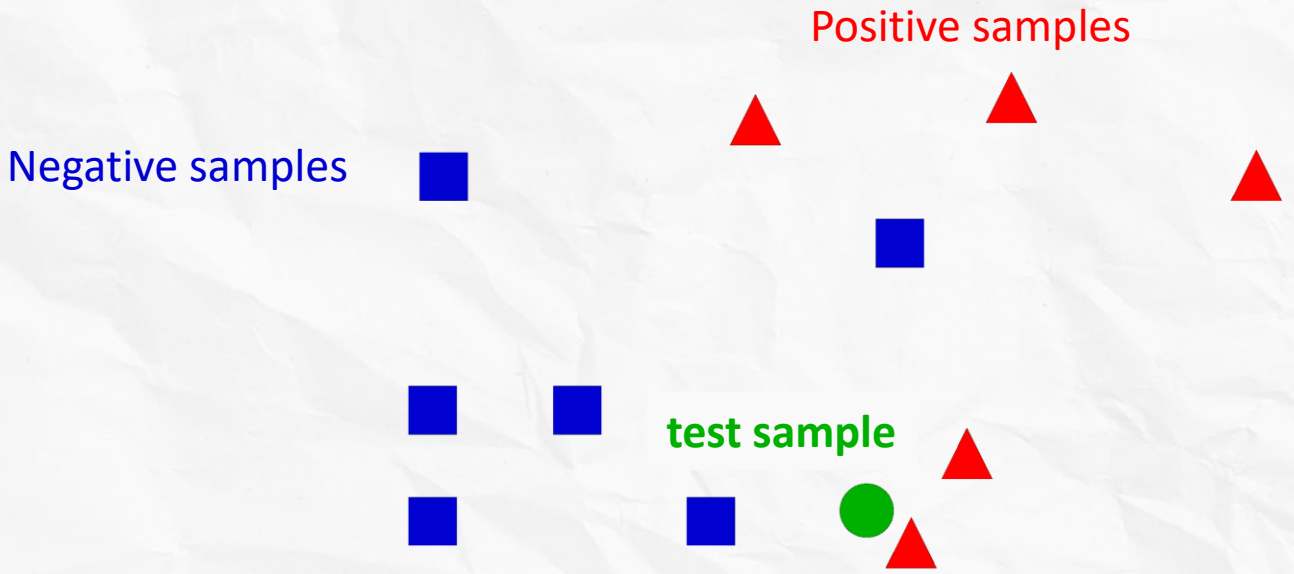
K近邻算法

涂文婷
tu.wenting@mail.shufe.edu.cn

什么是K近邻算法

- K近邻分类与回归

给定测试样本，基于某种距离度量找出训练集中与其最靠近的k个训练样本，然后基于这k个“邻居”的信息来进行预测.



y of neighbors → y of test sample

投票法，平均
加权投票法，加权平均法

机器学习

密度估计

涂文婷
tu.wenting@mail.shufe.edu.cn

密度估计

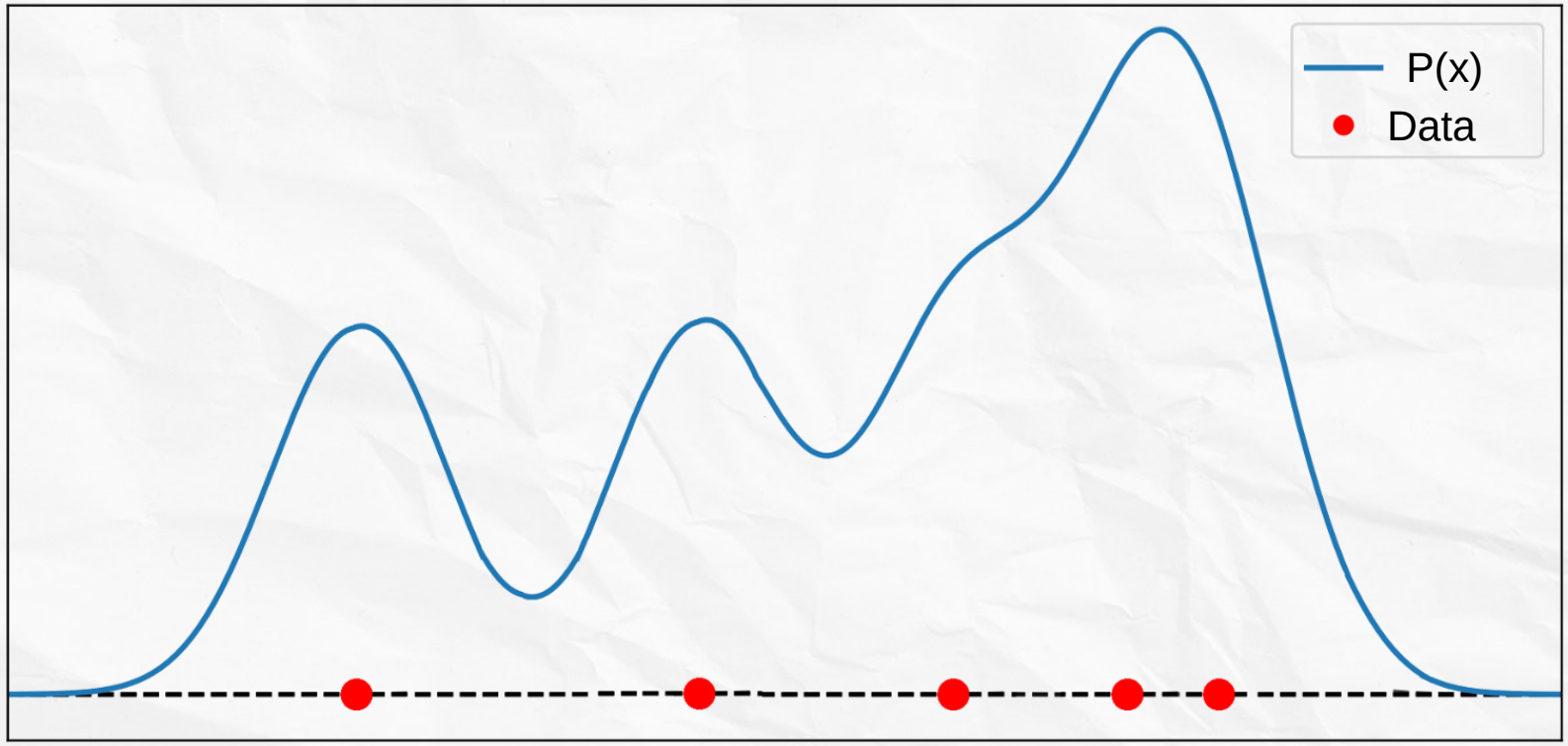
- 任务背景

利用独立采样点估计$p(\boldsymbol{x})$的概率密度函数$f(\boldsymbol{x})$

密度估计

- 非参数化方法

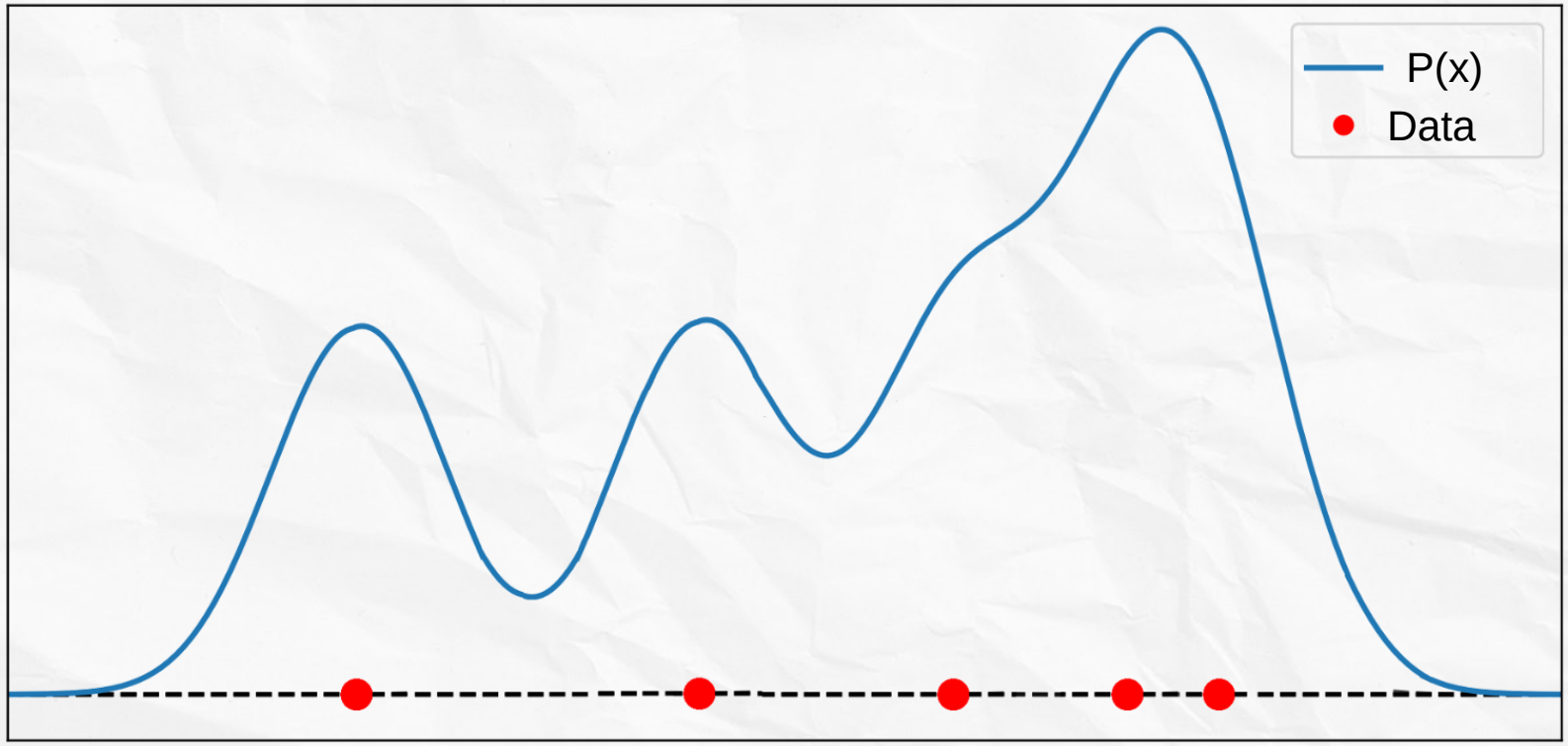
参数化 *parametric* 方法假设概率分布都有具体的函数形式，并且由少量的参数控制。之后通过采样集估计这些参数的值。



密度估计

- 非参数化方法

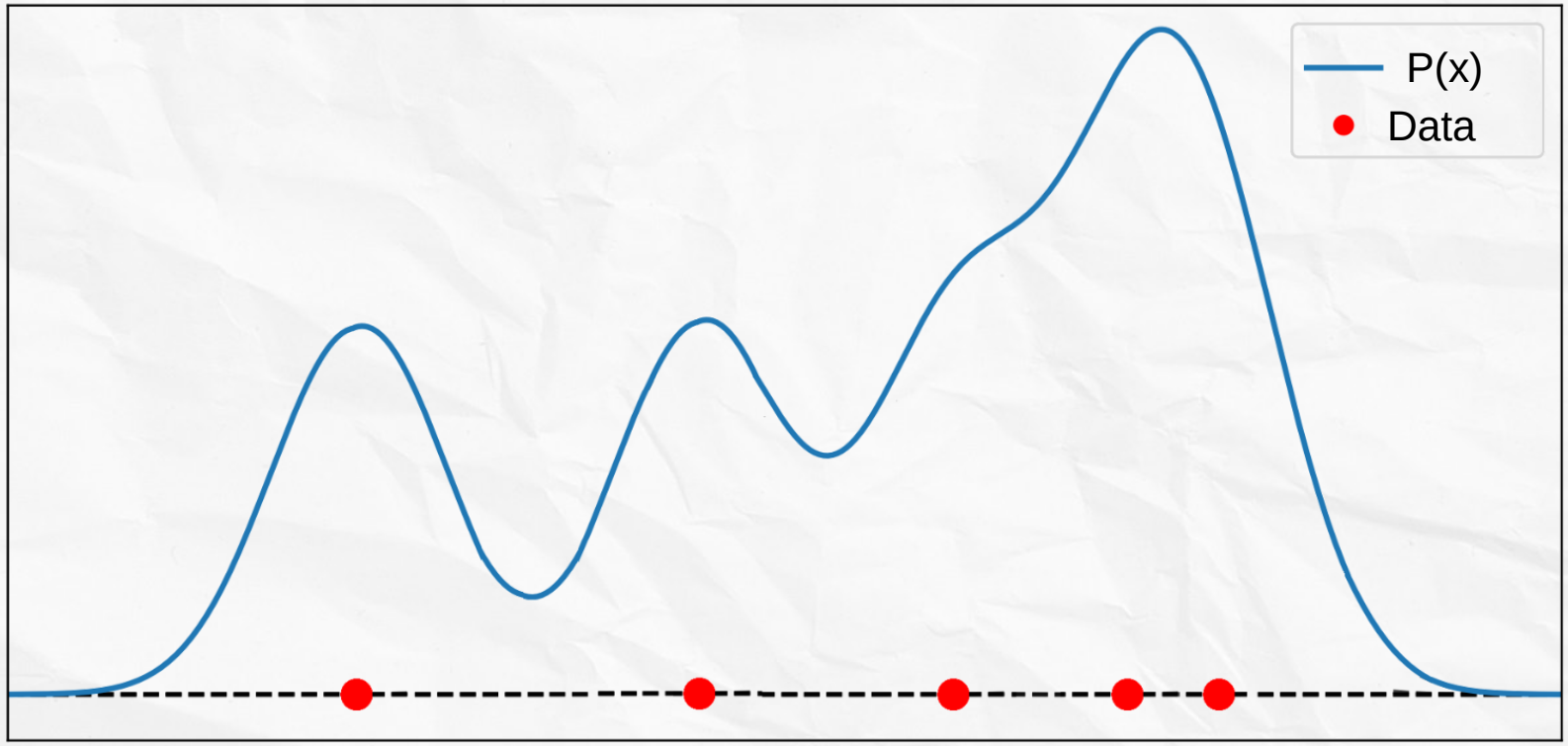
参数化 *parametric* 方法的一个重要局限性是选择的概率密度可能对于生成数据来说，是一个很差的模型，从而会导致相当差的预测表现。例如，如果生成数据的过程是多峰的，那么这种分布不可能被高斯分布描述，因为它是单峰的。



密度估计

- 非参数化方法

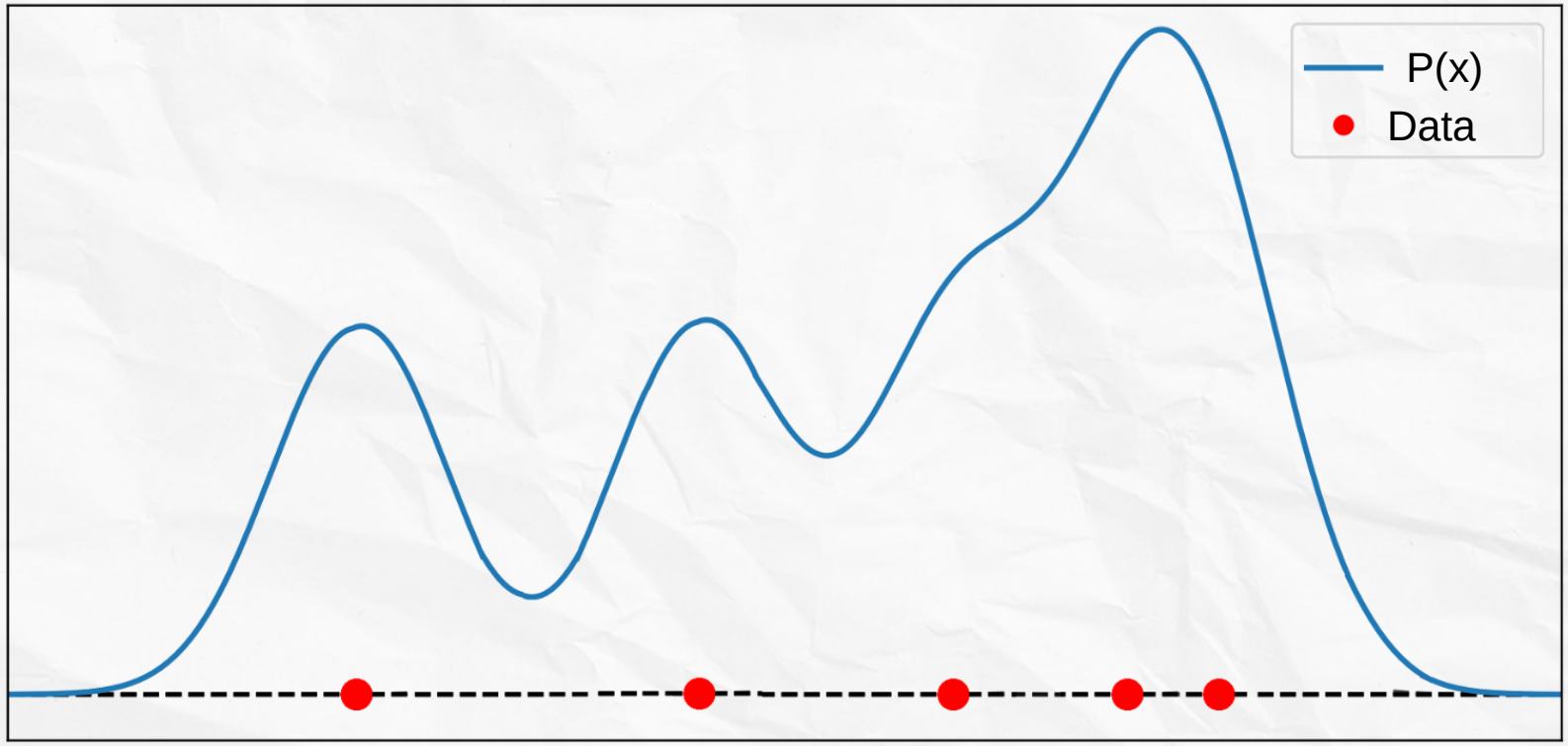
非参数化 *nomparametric* 则不对概率分布的函数形式作出参数化的假设。而是在很少的其他假设下进行估计。



密度估计

- 任务背景

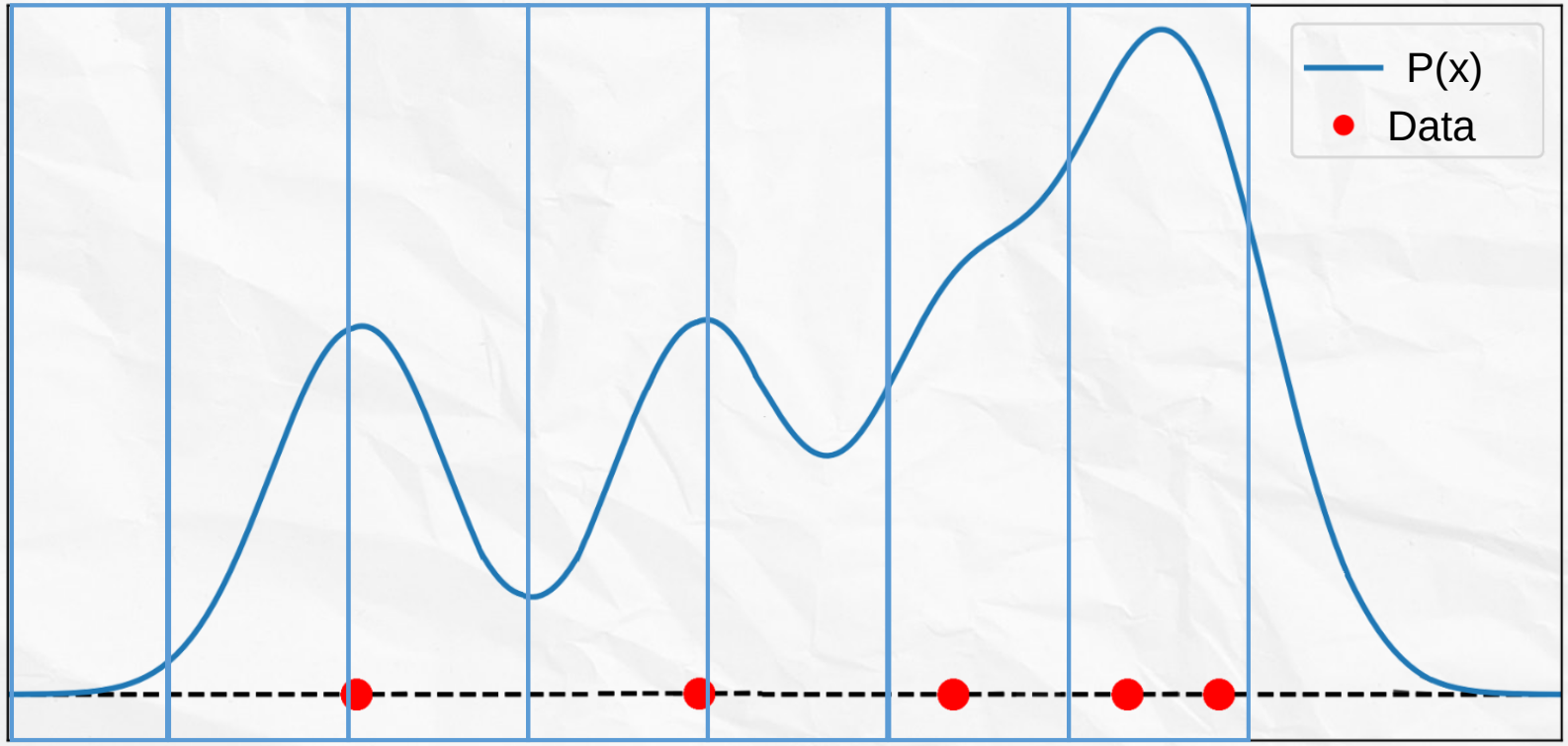
利用独立采样点估计$p(\boldsymbol{x})$的概率密度函数$f(\boldsymbol{x})$



密度估计

直方图法

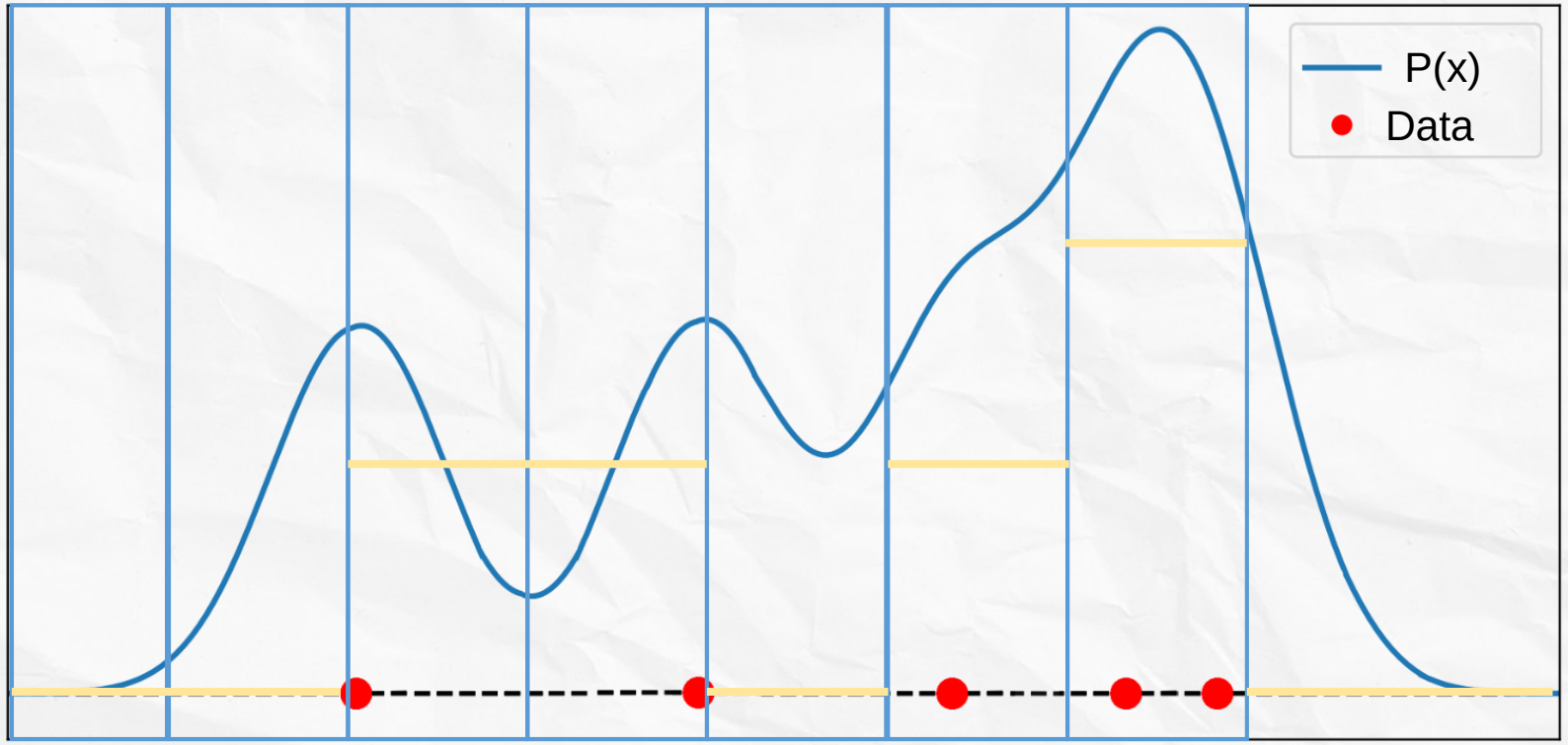
直方图法简单地把x的值域划分成不同的宽度为固定长度的箱子，然后对落在第每个箱子中的x的观测点数进行计数，然后把这种计数转换成归一化的概率密度。



密度估计

直方图法

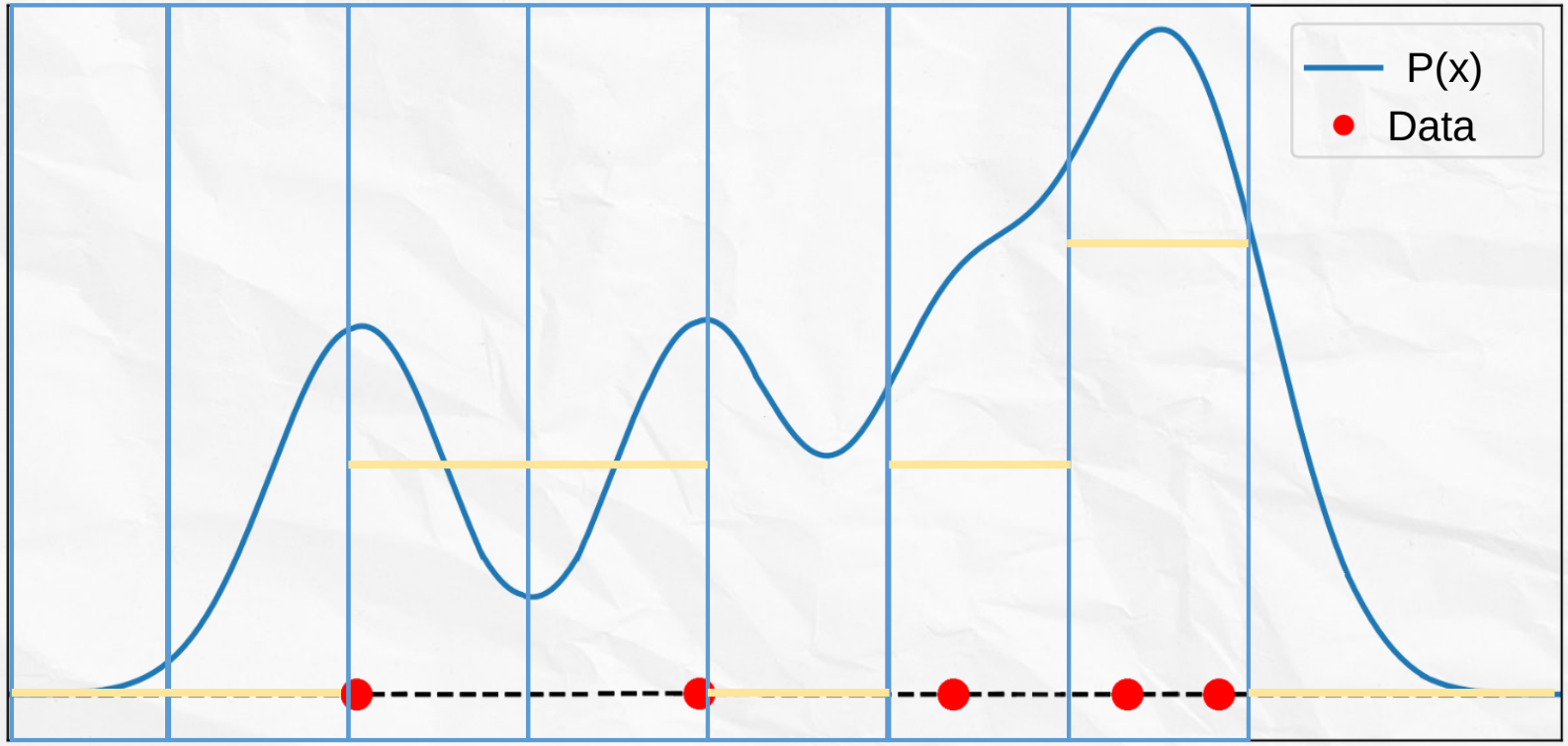
直方图法简单地把x的值域划分成不同的宽度为固定长度的箱子，然后对落在第每个箱子中的x的观测点数进行计数，然后把这种计数转换成归一化的概率密度。



密度估计

- 直方图法

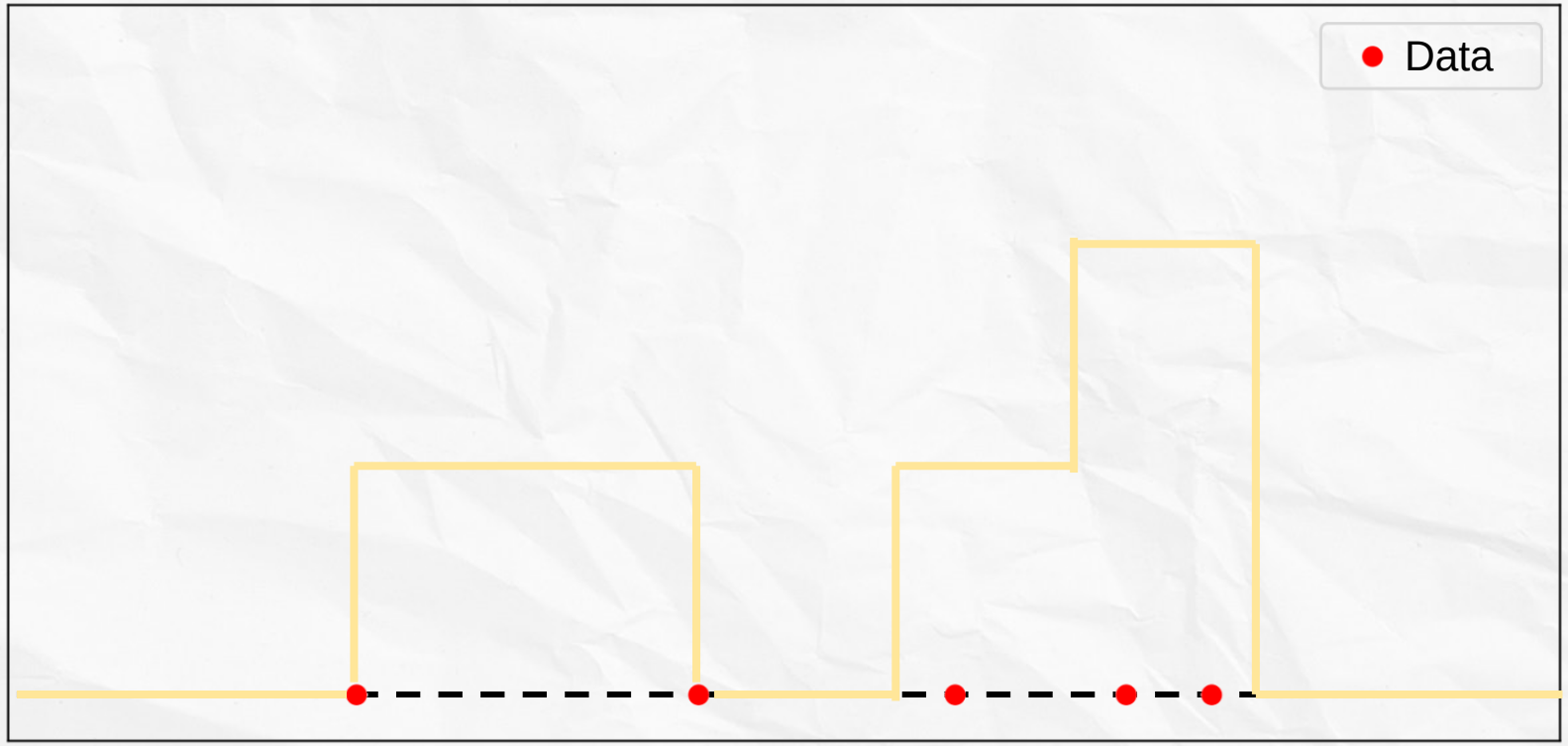
$p_i = \frac{n_i}{N\Delta}$, n_i为第i个箱子中的样本观测数量， N为观测总样本数， Δ为箱子宽度



密度估计

直方图法

直方图法对于快速地将一维或者二维的数据可视化很有用，但是并不适用于大多数概率密度估计的应用。一个明显的问题是估计的概率密度具有不连续性。



密度估计

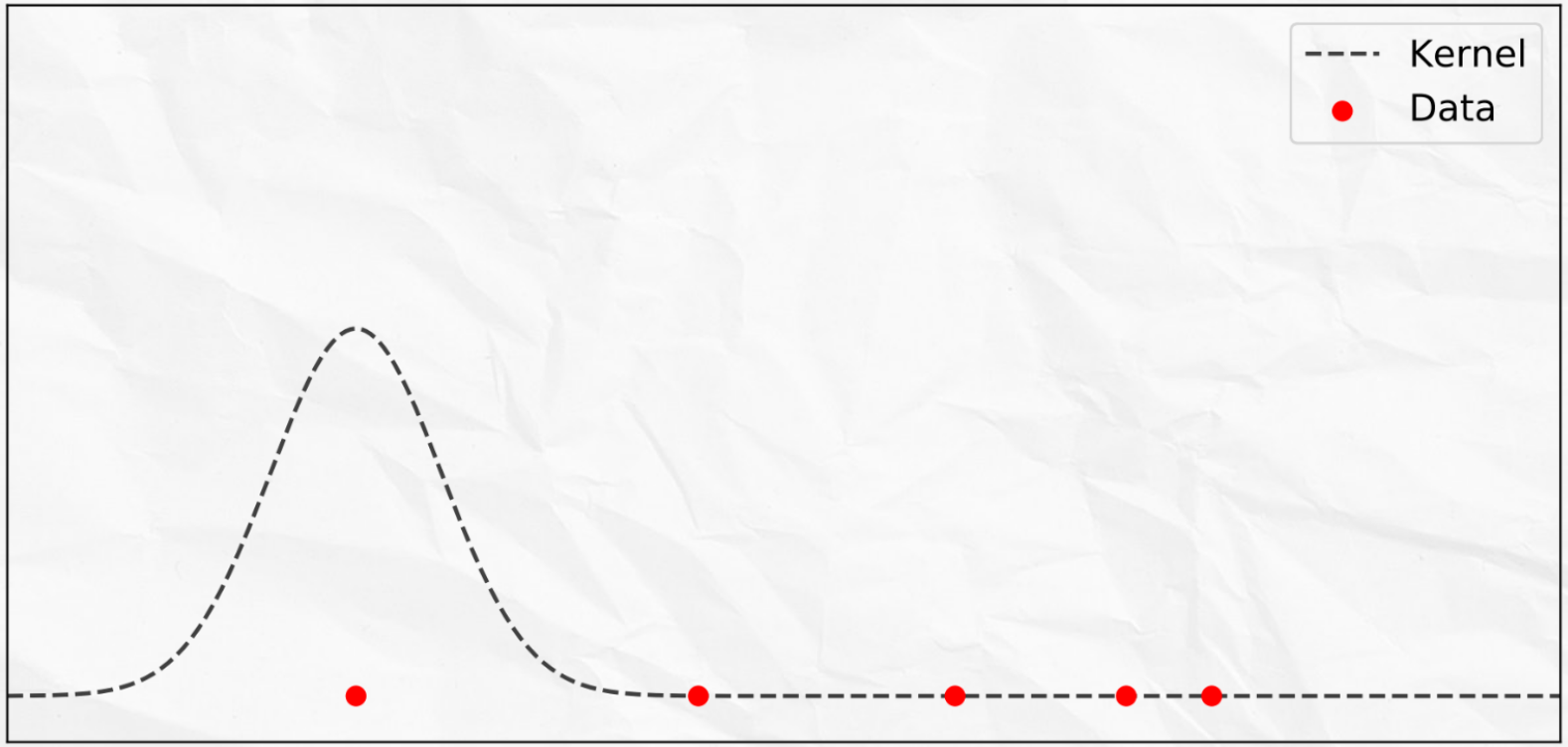
- 核密度估计法

  我们将每个样本点的贡献从箱子对应的计数器上加一转变成加入一个核函数

密度估计

- 核密度估计法

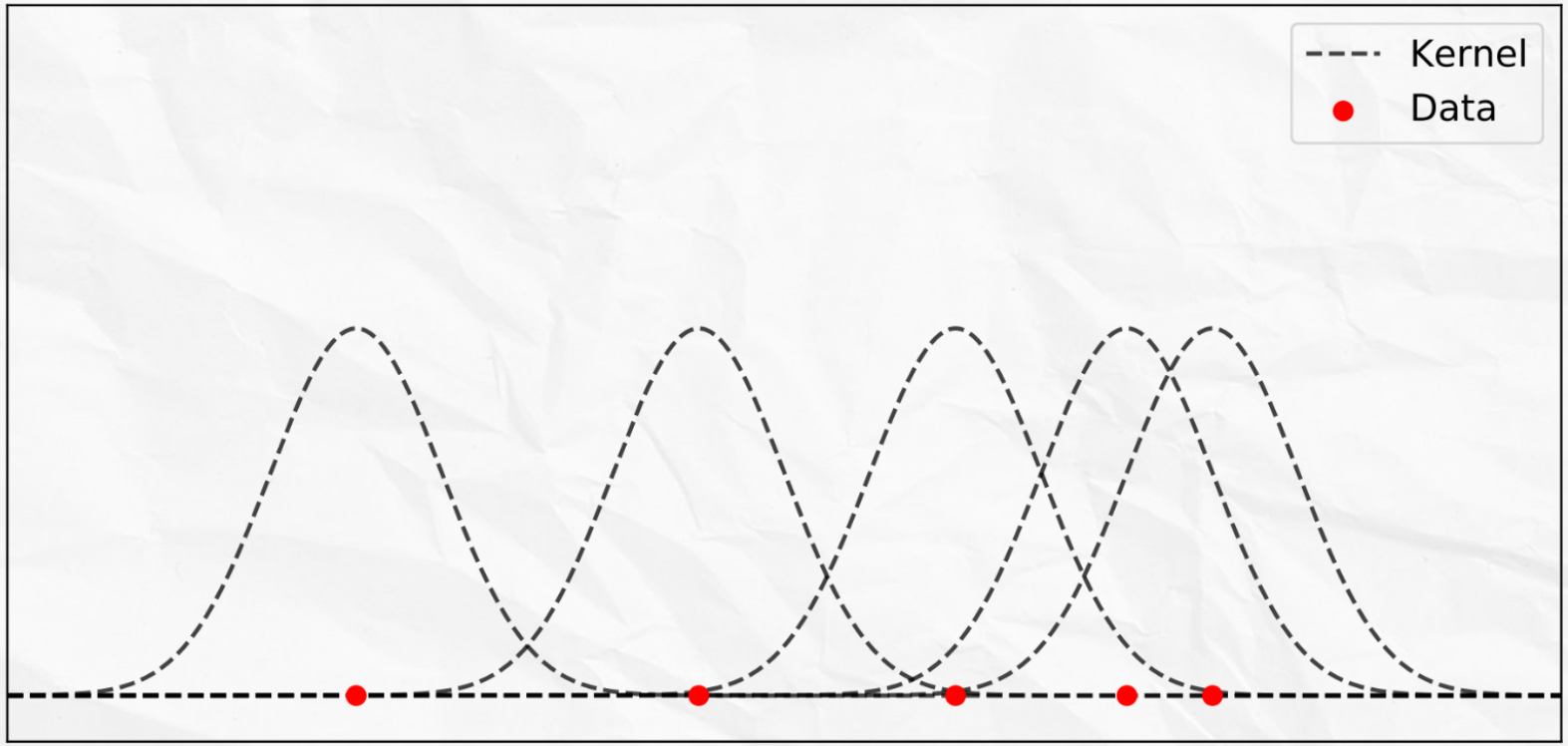
  我们将每个样本点的贡献从箱子对应的计数器上加一转变成加入一个核函数



密度估计

- 核密度估计法

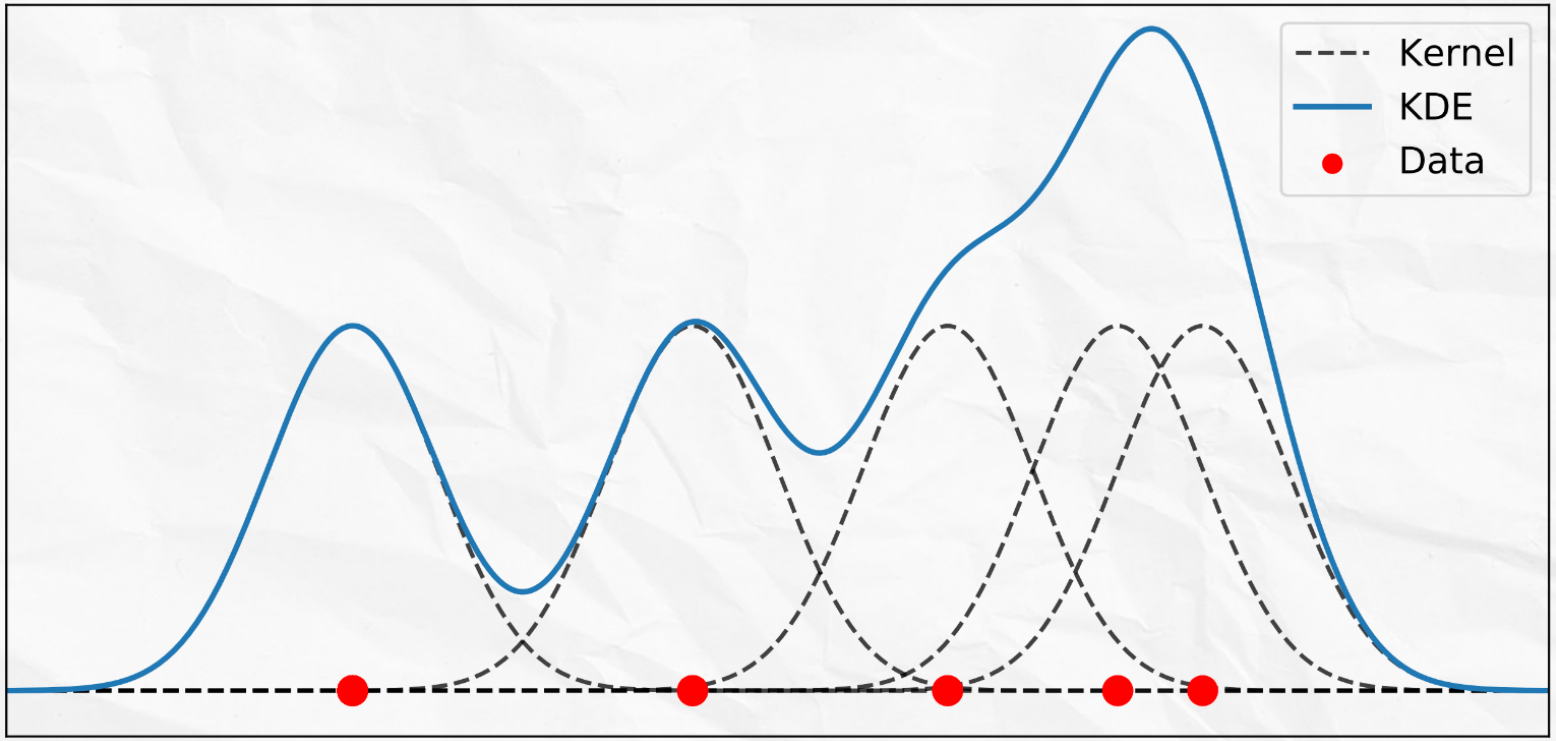
$$\hat{f}(x) = \frac{1}{N}\sum_{i=1}^{N} K(\boldsymbol{x} - \boldsymbol{x}_i)$$



密度估计

- 核密度估计法

$$\hat{f}(x) = \frac{1}{N}\sum_{i=1}^{N} K(\boldsymbol{x} - \boldsymbol{x}_i)$$

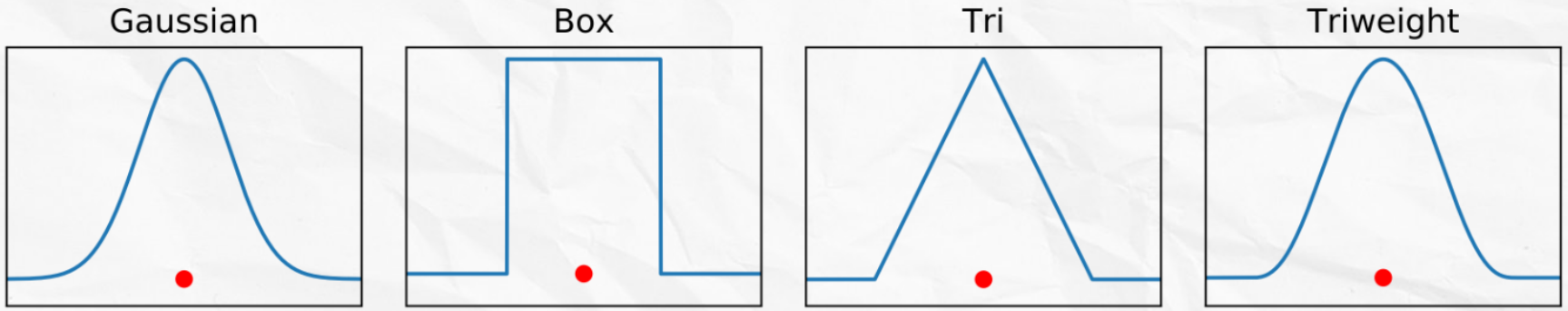


密度估计

核密度估计法

核函数符合：

- 非负性：$K(\boldsymbol{x}) \geq 0$, 对任意的$\boldsymbol{x}$
- 对称性：$K(\boldsymbol{x}) = K(-\boldsymbol{x})$, 对任意的$\boldsymbol{x}$
- 递减性：$K(\boldsymbol{x}) = K(-\boldsymbol{x})$, 对任意的$\boldsymbol{x} > 0$

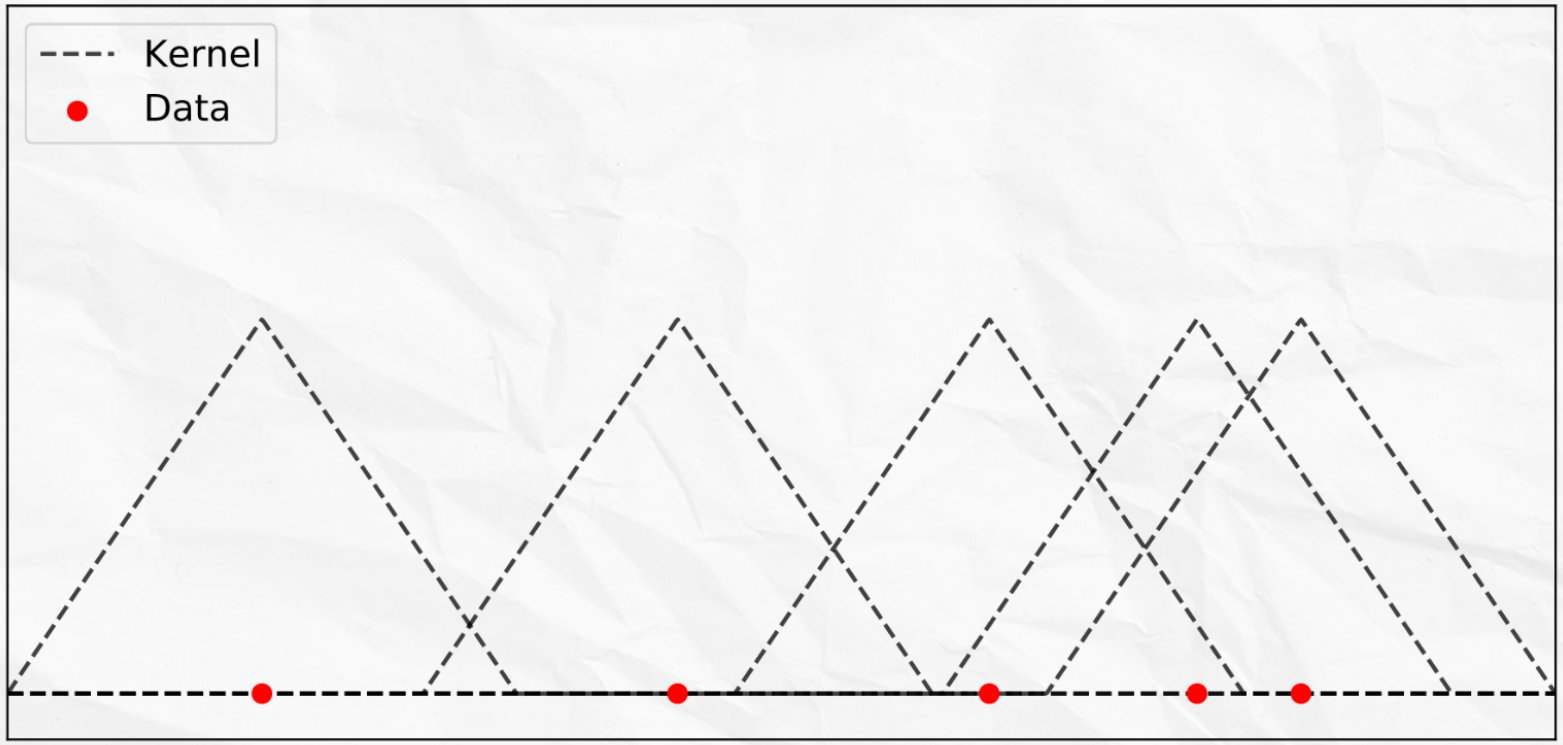


密度估计

- 核密度估计法

当我们使用*triangular* kernel

$$f(\boldsymbol{x}) \propto \max(1 - |\boldsymbol{x}|, 0)$$

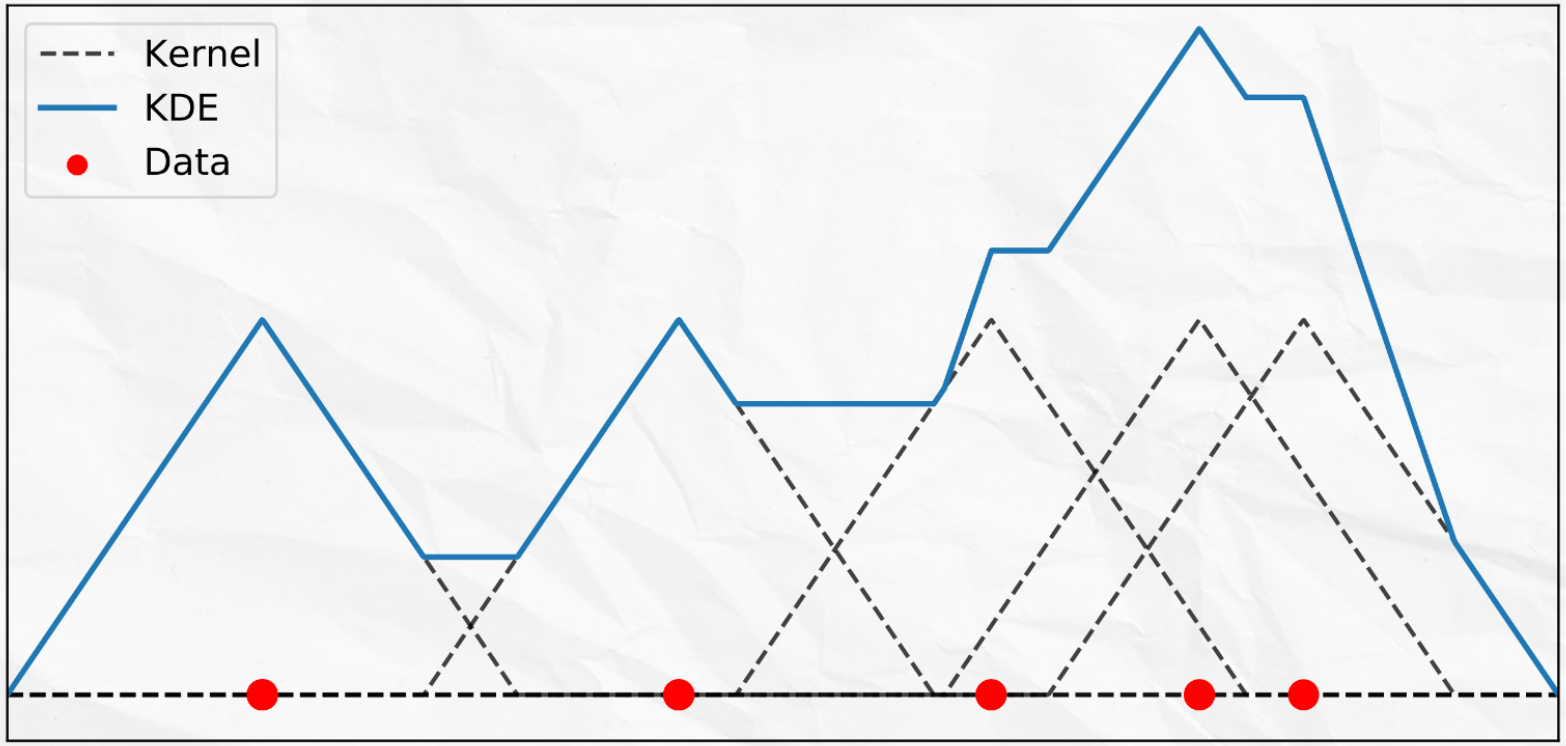


密度估计

- 核密度估计法

当我们使用*triangular* kernel

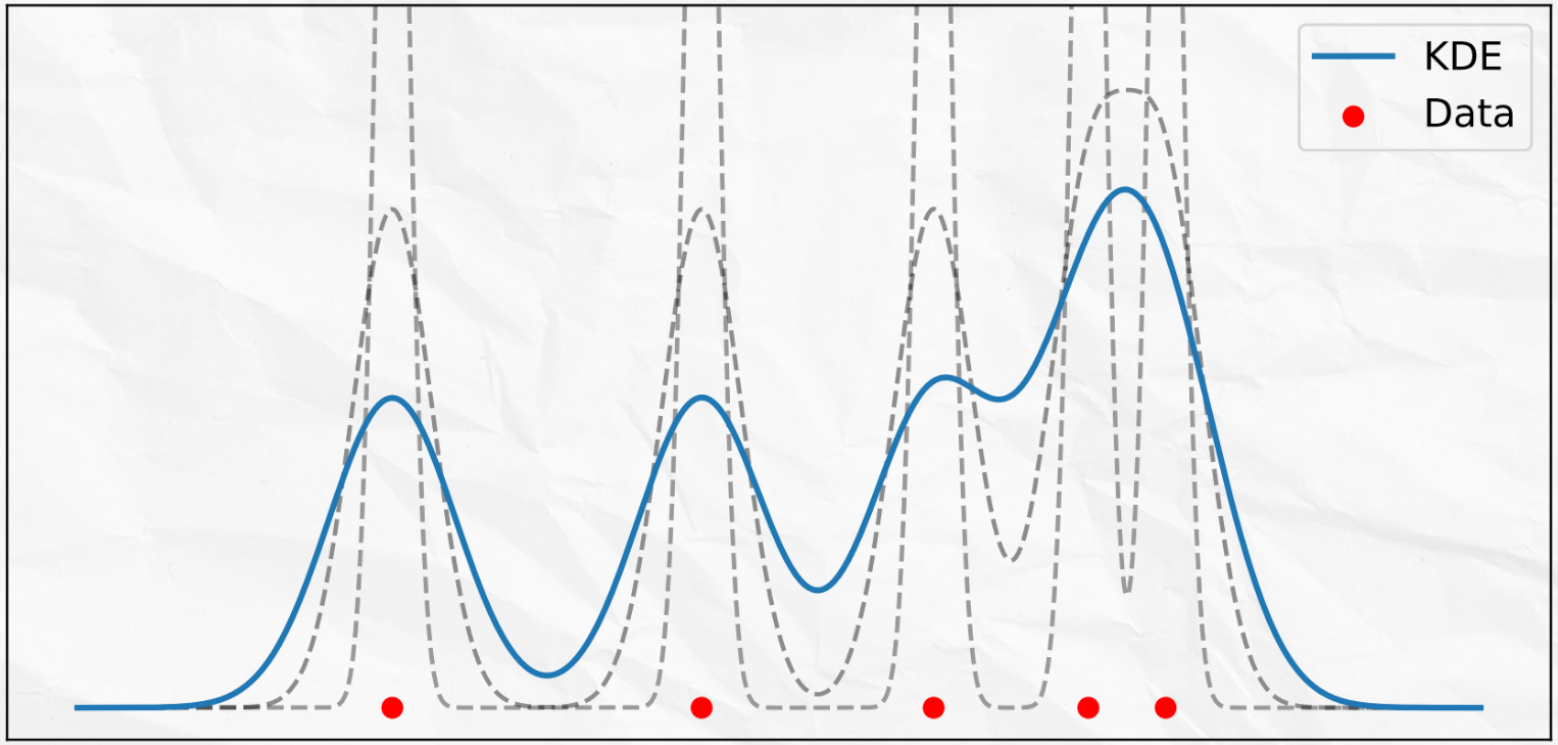
$$f(\boldsymbol{x}) \propto \max(1 - |\boldsymbol{x}|, 0)$$



密度估计

核密度估计法

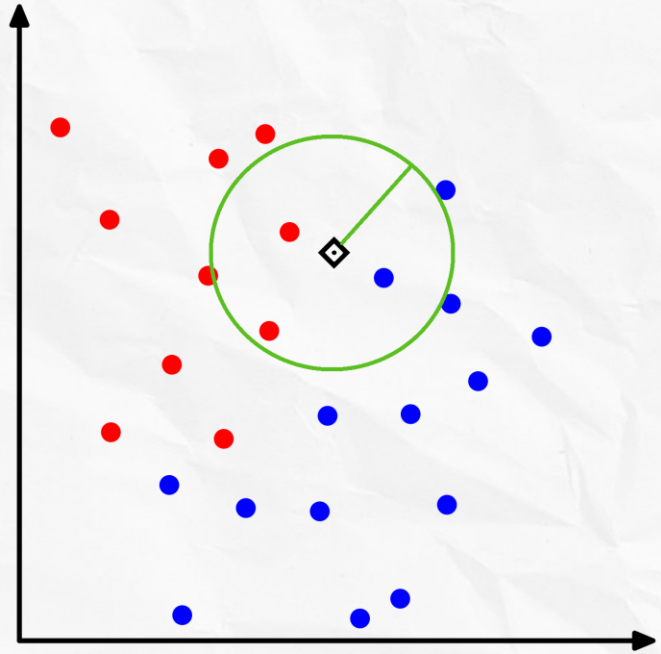
使用带宽*bandwidth* h来作为平滑参数，得到$\hat{f}(x) = \frac{1}{Nh}\sum_{i=1}^{N} K\left(\frac{x-x_i}{h}\right)$



小的h会造成模型对噪声过于敏感，而大的h会造成过度平滑

密度估计

- 从密度估计的角度解释近邻法



$$p(\boldsymbol{x}) = \frac{K}{NV}$$

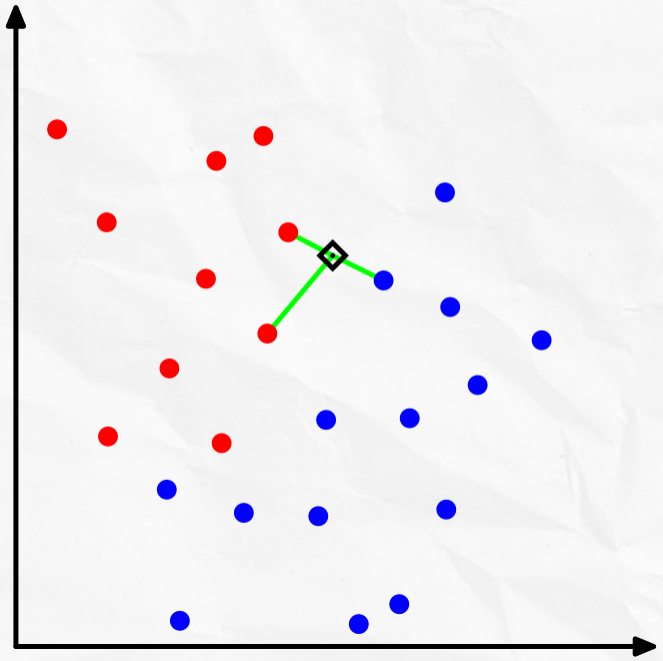
$$p(\boldsymbol{x} \mid c_k) = \frac{K_k}{N_k V}$$

$$p(c_k) = \frac{N_k}{N}$$

$$p(c_k \mid \boldsymbol{x}) = \frac{p(\boldsymbol{x} \mid c_k)p(c_k)}{p(\boldsymbol{x})} = \frac{K_k}{K}$$

密度估计

- 从密度估计的角度解释近邻法



$$p(\boldsymbol{x}) = \frac{K}{NV}$$

$$p(\boldsymbol{x} \mid c_k) = \frac{K_k}{N_k V}$$

$$p(c_k) = \frac{N_k}{N}$$

$$p(c_k \mid \boldsymbol{x}) = \frac{p(\boldsymbol{x} \mid c_k)p(c_k)}{p(\boldsymbol{x})} = \frac{K_k}{K}$$

机器学习

基于近邻算法的推荐系统

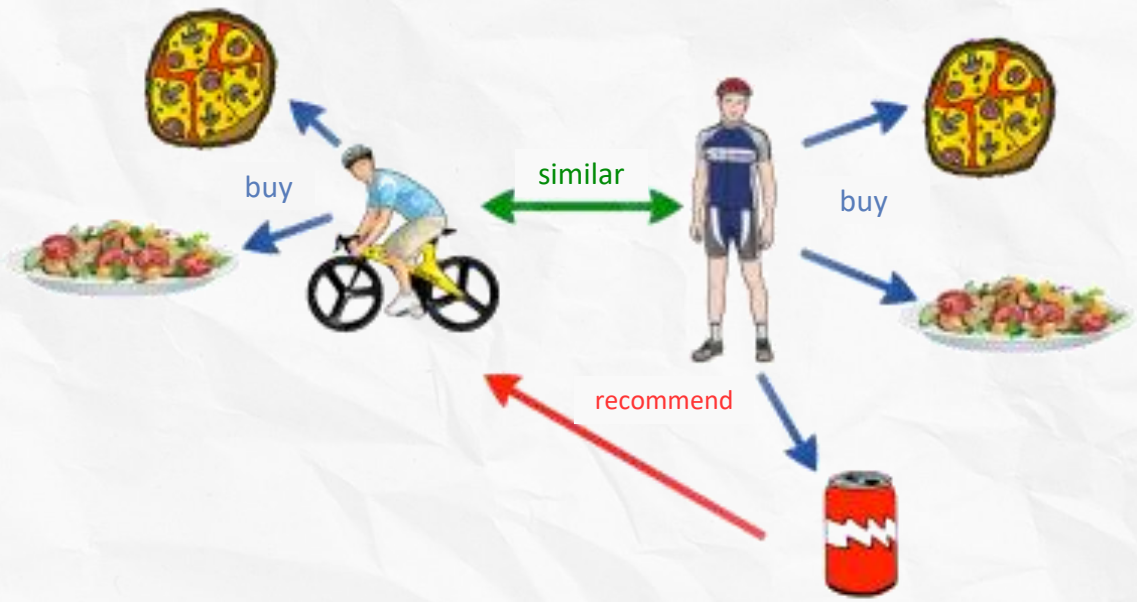
涂文婷
tu.wenting@mail.shufe.edu.cn

用近邻法实现推荐系统

- 推荐系统任务

用近邻法实现推荐系统

- 基于用户的协同过滤 UCF



用近邻法实现推荐系统

∘ 基于用户的协同过滤 UCF

假设u是目标用户，即我们想要推荐商品的对象

步骤1. 找到与u的历史打分行为相似的用户

$$R = \begin{array}{c} \begin{matrix} i_1 & i_2 & & i_j & & i_m \end{matrix} \\ \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1j} & \cdots & r_{1m} \\ r_{21} & & & & & \\ & & \ddots & & & \\ \vdots & & & r_{ij} & & \vdots \\ & & & & \ddots & \\ r_{n1} & & \cdots & & & r_{nm} \end{bmatrix} \end{array} \begin{matrix} u_1 \\ u_2 \\ \\ u_i \\ \\ u_n \end{matrix}$$

$$\text{sim}_{u,v} = \frac{\vec{r}_u \cdot \vec{r}_v}{\|\vec{r}_u\|_2 * \|\vec{r}_v\|_2} = \frac{\sum_{i \in I}(r_{u,i} - \bar{r}_u)(r_{v,i} - \bar{r}_v)}{\sqrt{\sum_{i \in I}(r_{u,i} - \bar{r}_u)^2}\sqrt{\sum_{i \in I}\left((r_{v,i} - \bar{r}_v\right)^2}}$$

挑选k个最相似的用户记为G_u

用近邻法实现推荐系统

∘ 基于用户的协同过滤 UCF

假设u是目标用户，即我们想要推荐商品的对象

步骤2. 利用这些相似用户对其他商品的打分预测u对那些他/她还未打分过的商品的打分，例如对于商品i的得分估计为：

$$\hat{r}_{u,i} = \bar{r}_u + \frac{\sum_{v \in G_u} \left(r_{v,i} - \bar{r}_v\right) \cdot \mathrm{sim}_{u,v}}{\sum_{v \in G_u} |w_{u,v}|}$$

步骤3. 排序商品得分，取得分最高的N个商品推荐给用户

用近邻法实现推荐系统

- 基于商品的协同过滤 ICF

假设我们定义商品i和商品j的相似度为：

$$\text{sim}_{i,j} = \cos(\vec{i}, \vec{j}) = \frac{\vec{i} \cdot \vec{j}}{\| \vec{i} \|_2 * \| \vec{j} \|_2}$$

$$R = \begin{array}{c} \begin{matrix} i_1 & i_2 & & i_j & & i_m \end{matrix} \\ \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1j} & \cdots & r_{1m} \\ r_{21} & & & & & \\ & & \ddots & & & \\ \vdots & & & r_{ij} & & \vdots \\ & & & & \ddots & \\ r_{n1} & & \cdots & & & r_{nm} \end{bmatrix} \end{array} \begin{matrix} u_1 \\ u_2 \\ \\ u_i \\ \\ u_n \end{matrix}$$

用户u对于商品i的得分可定义为

$$\hat{r}_{u,i} = \frac{\sum_{i \in C} r_{u,j} \cdot \text{sim}_{i,j}}{\sum_{j \in C} |\text{sim}_{i,j}|}$$