2021-09-08

> 相关章节：2.5

Nonparametric methods

介绍了三种 Nonparametric methods for density estimation：histogram method (直方图法)，kernel method (核方法)，K-nearest-neighbour method (K 近邻算法).

一方面 simple parametric models are very restricted in terms of the forms of distribution that they can represent (简单参数模型的表达能力不强)，另一方面，只要数据量足够，nonparametric methods 的表达能力总能满足要求，但模型的复杂度会随着训练集的增大而不断增大，we therefore need to find density models that are very flexible (也就是表达能力强的意思) and yet for which the complexity of the models can be controlled independently of the size of the training set, and we shall see in subsequent chapters how to achieve this (因此，下面的章节，我们会学习表达能力强，模型复杂度可控且不依赖训练集大小的模型).

Histogram density estimator

Standard histograms simply partition x into distinct bins of width $\triangle_i$ and then count the number n_i of observations of x falling
in bin i. Probability value for each bin is given by:

$$p_i = \frac{n_i}{N\triangle_i}$$

This gives a model for the density $p(x)$ that is constant over the width of each bin, and often the bins are chosen to have
the same width $\triangle_i = \triangle$.

优点：

- Note that the histogram method has the property (unlike the methods to be discussed shortly) that, once the histogram has been computed, the data set itself can be discarded (histogram method 的一个好处是，一旦 histogram 构建完毕，样本集本身就不再需要了，可以丢弃，而下面介绍的核方法和 K 近邻方法需要存储所有样本数据).
- the histogram approach is easily applied if the data points are arriving sequentially.

缺点：

- the estimated density has discontinuities that are due to the bin edges rather than any property of the underlying distribution that generated the data.
- the curse of dimensionality.

Kernel density estimator & Nearest-neighbour method

we obtain our density estimate in the form:

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

上述公式的不同使用方式就得到了不同的估计方法:

- K-nearest-neighbour technique: fix K and determine the value of V from the data,
- kernel approach: fix V and determine K from the data.

收敛性：It can be shown that both the K-nearest-neighbour density estimator and the kernel density estimator converge to the true probability density in the limit $N \rightarrow \infty$ provided V shrinks suitably with N, and K grows with N.

Kernel density estimator

Example 1：

取核函数：

$$k(u)=\begin{cases}1, & \left|u_i\right| \leqslant \frac{1}{2}, \quad i=1,\cdots,D \\ 0, & \text{otherwise}\end{cases}$$

则

$$k\left(\frac{x-x_n}{h}\right)$$

给出了样本点 x_n 是否在以 x 为中心，边长为 h 的超立方体中，其中 D 为这个立方体的维度，也就是数据点的维度。此外，根据对称性，上式也给出了点 x 是否在以 x_n 为中心，边长为 h 的超立方体中。

此时：

$$K=\sum_{n=1}^{N} k\left(\frac{x-x_n}{h}\right)$$
$$V=h^D$$

则点 x 处的概率密度：

$$p(x)=\frac{1}{N} \sum_{n=1}^{N} \frac{1}{h^D} k\left(\frac{x-x_n}{h}\right)$$

Example 2：

取高斯核函数时：

$$p(x)=\frac{1}{N} \sum_{n=1}^{N} \frac{1}{\left(2 \pi h^2\right)^{D/2}} \exp \left\{-\frac{\left\|x-x_n\right\|^2}{2 h^2}\right\}$$

where h represents the standard deviation of the Gaussian components. Thus our density model is obtained by placing a Gaussian over each data point and then adding up the contributions over the whole data set, and then dividing by N so that the density is correctly normalized.

In general:

核函数满足如下的条件：

$$
\begin{gathered}
k(u) \geqslant 0, \\
\int k(u) \mathrm{d} u=1
\end{gathered}
$$

上面是书中给出的条件，事实上，个人认为更一般地应该是：

$$
\begin{gathered}
k(u) \geqslant 0, \\
\int k(u) \mathrm{d} u=C
\end{gathered}
$$

其中，C 为常数，由下可知，事实上 $V=C$。

可以看到，example 1 和 histogram method 很类似，histogram method 也是 fix V and determine K from data，其与 kernel method 的区别在于 histogram method 被动地划分统计区间，而 kernel method 则主动地以要计算概率密度的点 x 为中心构建统计区间。

对于核函数一般形式的理解：

1. 还是从 $p(x)=\frac{K}{NV}$ 的角度，example 1 是这一公式的直接应用，很好理解，但对于 example 2 我们应该如何从这一公式出发进行理解呢？事实上，我们可以将 example 1 看做是该公式的 hard 模式，而 example 2 则是 soft 模式。

   可以看到，$k(u)$ 实际上就是一个概率密度函数 (或者说未归一化的概率密度函数)，核函数 $k\left(\frac{x-x_n}{h}\right)$ 以待评估的点 x 为中心，对空间中的各点赋予不同的权重。 example 1 中，在以 x 为中心，边长为 h 的超立方体空间中点的权重为 1，否则为 0；example 2 中，越靠近 x 的空间，权重越大，权重分布的集中度由标准差 h 控制，因此，$p(x)=\frac{K}{NV}$ 更一般地应该是：

   $$
   p(x)=\frac{1}{N} \cdot \frac{\sum_{n=1}^{N} \text{第 } n \text{ 个样本点的权重}}{\text{整个空间的加权体积}}
   $$

   当 $k(u)$ 已经归一化时，加权体积 $V=1$，当 $k(u)$ 未归一化时，加权体积 $V=C$。example 1 上述公式的 hard 版本，权重非 0 即 1，example 2 则是 soft 版本。

   进一步，若 $k(u)$ 已归一化，$k\left(\frac{x-x_n}{h}\right)$ 可表示以 x 为中心点，x_n 处的概率密度，因此，$p(x)=\frac{1}{N} \sum_{n=1}^{N} k\left(\frac{x-x_n}{h}\right)$ 实际上就是通过计算各样本点 x_n ($n=1,2,\cdots,N$) 处概率密度的算术平均来估计 x 处的概率密度。

2. 我们从混合模型的角度来解释 kernel method。由对称性，$k\left(\frac{x-x_n}{h}\right)$ 又可表示以 x_n 为中心，x 为随机变量的概率密度函数。由此，我们可得 N 个同类型的概率分布 (函数图像形状相同，但位置不同，平移后可完全重合) $k\left(\frac{x-x_n}{h}\right)$ ($n=1,2,\cdots,N$)，而 $p(x)$ 则是由这 N 个概率模型组成的混合模型：

   $$
   p(x)=\frac{1}{N} \sum_{n=1}^{N} k\left(\frac{x-x_n}{h}\right)
   $$

   各子模型的权重相同，为 $\frac{1}{N}$。

从混合模型的角度，很容易看到 example 1 得到的概率密度和 histogram 一样不连续，但 example 2 产生的是连续的概率密度函数。

核方法的优缺点：It has a great merit that there is no computation involved in the ‘training’ phase because this simply requires storage of the training set. However, this is also one of its great weaknesses because the computational cost of evaluating the density grows linearly with the size of the data set.

Nearest-neighbour method

One of the difficulties with the kernel approach to density estimation is that the parameter h governing the kernel width is fixed for all kernels (即从混合模型的角度，所有子模型都采用相同的概率密度函数，只是中心点的位置不同而已). The optimal choice for h may be dependent on location within the data space. This issue is addressed by nearest-neighbour methods for density estimation.

K-nearest-neighbour method for **local density estimation**： we consider a small sphere centred on the point x at which we wish to estimate the density $p(x)$, and we allow the radius of the sphere to grow until it contains precisely K data points. The estimate of the density $p(x)$ is then given by $p(x) = \frac{K}{NV}$ with V set to the volume of the resulting sphere.

优缺点：

- Note that the model produced by K nearest neighbours is not a true density model because the integral over all space diverges (K-nearest-neighbour density estimator 得到的并不是一个 proper 的概率密度函数，其在整个空间中的积分是发散的).
- both the K-nearest-neighbour method, and the kernel density estimator, require the entire training data set to be stored, leading to expensive computation if the data set is large. This effect can be offset, at the expense of some additional one-off computation, by constructing tree-based search structures to allow (approximate) near neighbours to be found efficiently without doing an exhaustive search of the data set.