关于插值算法的报告

韩泽尧*　　学号：1800011314

【摘要】 很多现实中的问题需要我们通过已知的、离散的数据点，在给定的区域内推求新数据点。能实现这一目标的方法被称作插值。插值作为一种数值方法，在科学、工程等等领域有着极为广泛的应用。本文介绍了一些常用的插值数值方法，分别指出了他们适用的领域以及各自的不足之处，并对这些算法在目前已有 Python 库中的实现进行了简要介绍。

【关键词】 插值，算法，数值分析，scipy

1　问题的表述

插值问题一般可以表述为 [1]，在 N 维空间的一个区域 Ω 中，给定一些离散的数据点 $x_1, x_2, \ldots, x_n \in \Omega$，以及一个定义域在这个有限集合上的函数 f，要求找到一个插值函数 s，其定义域在整个 Ω 上，并且满足 $s(x_i) = f(x_i), \quad \forall i = 1, 2, \ldots, n.$

上述定义可以通过一个简单例子来理解——在绘制地球地形图的过程中，原则上我们需要知道球面上每一个坐标点（用经、纬度表示）处的海拔，才能获得精确无误的地形图。但在实际中，我们只可能借助仪器和有限的人力，测量出有限个坐标点 $x_1, x_2, \ldots, x_n$ 处的海拔 $h_1, h_2 \ldots, h_n$，需要从有限个数据点推知地球上任一点的海拔，这个过程便一定会用到插值。从数学上讲，满足 $f(x_i) = h_i$ 的函数是有无穷多个的，如何从无穷个函数中选择一个尽量能还原地形真实情态的函数，是我们需要解决的问题。

2　常用的插值算法

2.1　最近邻插值（**Nearest-neighbor interpolation**）

将任意一点的函数值定为与之距离最近的一个数据点的值，最常见于数码照片的显示中——使用摄像工具记录下了矩形阵列排列的有限个点的颜色，则“插值”后的显示结果为一个一个的方形“像素块”，如图1所示。

从示例中可以看出，这个方法适用于数据点较为密集，且相邻的点间数值变化剧烈（如照片中物体边缘两侧颜色可能完全不同）的情况。对于比如光滑曲线的拟合问题则

*北京大学物理学院 2018 级本科三班

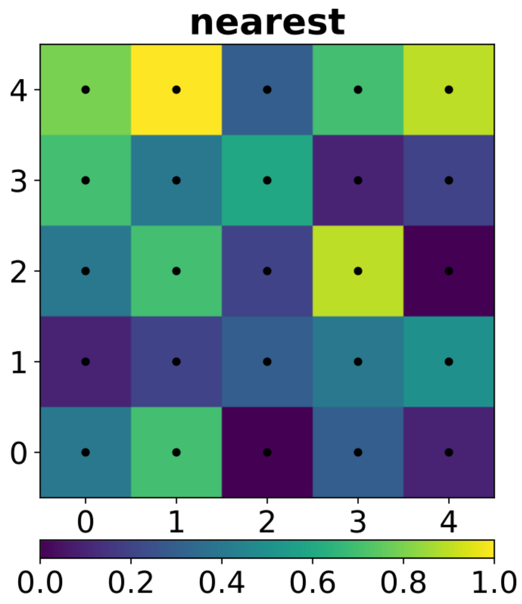


图 1: 最近邻插值在 5×5 网格上示例

并不适用。

Python 中使用 scipy.interpolate.NearestNDInterpolator 可以对 N 维空间中数据点进行最近邻插值 [3]。

2.2 多项式插值（Polynomial interpolation）

对于一维情况，由数学知识可知 [2]，过 n 个点 (x_i, y_i)，只要满足 $\forall i \neq j, x_i \neq x_j$，则一定存在一个唯一的 $n-1$ 阶多项式函数 $P(x) = \sum$，满足 $P(x_i) = y_i$

有很多种方法可以用来求出这个多项式的具体形式，其中形式最简单的是拉格朗日法——

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

拉格朗日法的一个缺点是在数据点增加时，所有的相关系数需要全部重新计算，缺乏承袭性。使用 Aitken 插值或牛顿插值方法则可以解决这个问题。

多项式插值对于一些问题，尤其是对原本就和多项式性质接近的函数来说，可以获得良好的结果，但在很多问题上会出现“过拟合”的现象，最经典的一个例子是，如果采用多项式来插值 Runge 函数 [6]：

$$f(x_i) = \frac{1}{1 + x_i^2}$$

从图2中可以看到，插值得到的结果会出现剧烈的“抖动”，且使用的插值多项式次数越高，与目标函数的差别反而越明显。并且，多项式插值的计算复杂度也很高。

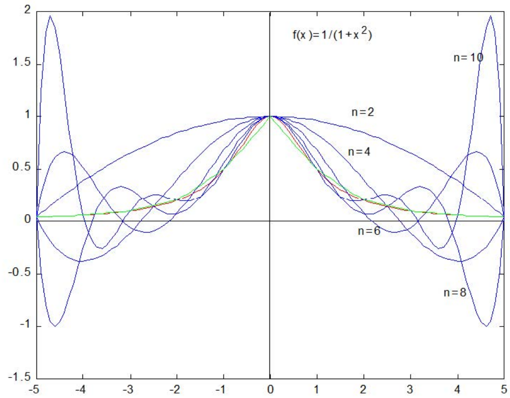


图 2: Runge 函数的多项式插值结果

在 Python 中，拉格朗日插值方法被封装为 scipy.interpolate 中的 lagrange 函数，将 x、y 坐标值作以列表形式传入该函数，返回对象为 poly1d 类型 [3]，其中包含了多项式的 $n-1$ 个系数。

2.3 样条插值（Spline interpolation）

样条插值的名称来源于过去工程中为了插值一组数据使用的一种细长、有弹性的木条，在飞机或轮船制造中被用于描绘光滑的外形曲线 [4]。如图3所示，将木条用钉子在几个位置作固定之后，木条便会自然弯曲成一条十分光滑的插值曲线。从物理原理上来讲，最终的形状会是弹性势能最低的状态。

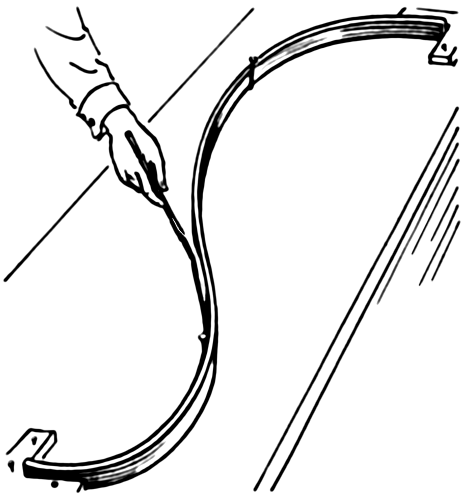


图 3: 工程制图中常用的样条

数学上的样条插值为分段低次多项式插值方法的一种——在不同的区间 (x_i, x_{i+1}) 内采用不同的、m 次的多项式，并要求在区间端点两侧多项式的函数值和直到 $m-1$ 阶

的导数都连续，通过求解一些线性方程组，可以将各段区间内的 m 次多项式都唯一确定下来，称为“m 次样条插值”。

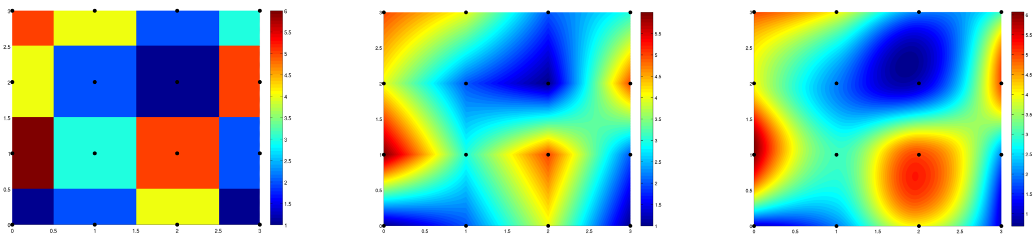
一般来说，样条插值的次数 m 不会很高，最常用的是三次样条插值，这样既可以比较好地解决高次多项式可能带来的“振荡”现象，同时函数的零、一、二阶导数均连续，使得最终的结果也能足够光滑。可以证明，该方法获得的结果恰好能使总的“弹性势能”最低。

样条插值方法是目前对于一维插值问题最常用的方法之一，后来又发展出 B-样条插值算法，在偏微分方程求解、计算机图形学等领域中有着重要的应用。

scipy.interpolate.CubicSpline(x, y, axis=0, bc_type=’not-a-knot’, extrapolate=None) 实现了三次样条插值算法，使用 x,y 两个参数传入数据数组，还可以通过修改 bc_type 参数调整边界条件。B-样条在 scipy 库中也有相应实现 [3]。

2.4 多变量函数插值（Multivariate interpolation）

对于二维空间中的双变量函数 $f(x,y)$ 插值，除了前述最简单的最近邻方法外，常用的还有双线性方法和双三次方法。篇幅所限，这里不对具体原理展开详细介绍。双变量函数插值在数据可视化中较为常用，scipy.interpolate 中也提供了 interp2d 类，实现了线性、三次和五次插值算法可以供用户调用 [3]。



(a) 最近邻插值　(b) 双线性插值　(c) 双三次插值

图 4: 双变量插值结果对比

3 总结

插值问题的研究有着很长的历史，已经发展出了很多非常成熟的算法，有相当多的现成库可以供我们调用。即便如此，了解其算法背后的原理对于我们的实际工作仍然是相当有益的。

发展出的这些关于插值的数学理论，其意义远不止于仅仅拟合一些数据。以物理学为例，B-样条函数在原子、分子物理学中的应用便是前些年的一个热门问题，有着颇广的研究空间 [5]。

参考文献

[1] Press, W. H., Flannery, B. P., Teukolsky, S. A., & Vetterling, W. T. (1989). Numerical recipes (Vol. 3). Cambridge: Cambridge University Press. Chapter 3.

[2] 见 Axler SJ. Linear algebra done right. New York: Springer; 1997 Jul 18. 第四章习题 5.

[3] scipy.interpolate官方文档：
https://docs.scipy.org/doc/scipy/reference/interpolate.html

[4] https://en.wikipedia.org/wiki/Spline_interpolation

[5] Vanne, Yulian V., and Alejandro Saenz. ”Numerical treatment of diatomic two-electron molecules using a B-spline based CI method.” Journal of Physics B: Atomic, Molecular and Optical Physics 37.20 (2004): 4101.

[6] Runge, Carl (1901), ”Über empirische Funktionen und die Interpolation zwischen äquidistanten Ordinaten”, Zeitschrift für Mathematik und Physik, 46: 224–243.