4.2 2D Line Plot 平面线图

点动成线，线动成面。散点顺序连线的结果就是线图。

颗粒度

绘制线图时，大家首先注意颗粒度 (granularity)，即采样，的问题。

多数情况，在绘制一元函数线图时，我们用 numpy.linspace() 生成自变量的等差数列。图 1 的两幅图中的散点都来自于正弦函数 $f(x) = \sin(x)$。显然，颗粒度粗糙时，用线图可视化一元函数可能会误导读者。

等差数列的公差越小，曲线的颗粒度越高，这样平面线图看上去“光滑”。如图 2 (a) 所示，等差数列有 101 个元素。将这些散点顺序连接便得到图 2 (b)。对于 $f(x) = \sin(x)$ 这个并不复杂的一元函数，图 2 (a) 的颗粒度显然足够用了。

如图 3 所示，为了可视化 $f(x) = 1/\sin(x)$ 在靠近 0 附近的振荡，我们需要极其细腻的颗粒度。

Jupyter 笔记 BK_2_Topic_4.02_1.ipynb 绘制图 1、图 2、图 3。

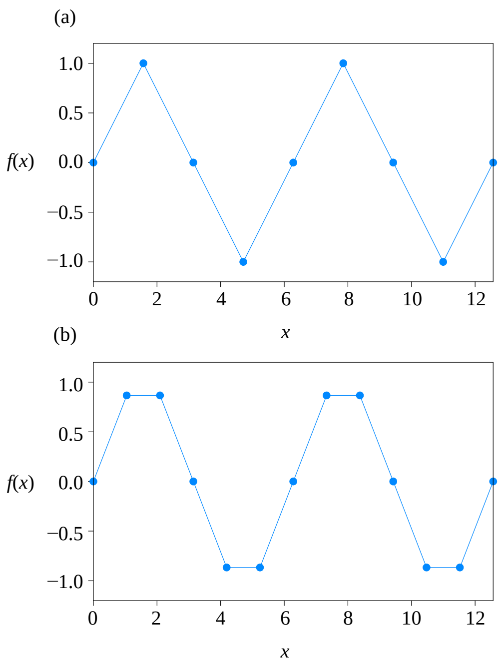

图 1. 颗粒度粗糙

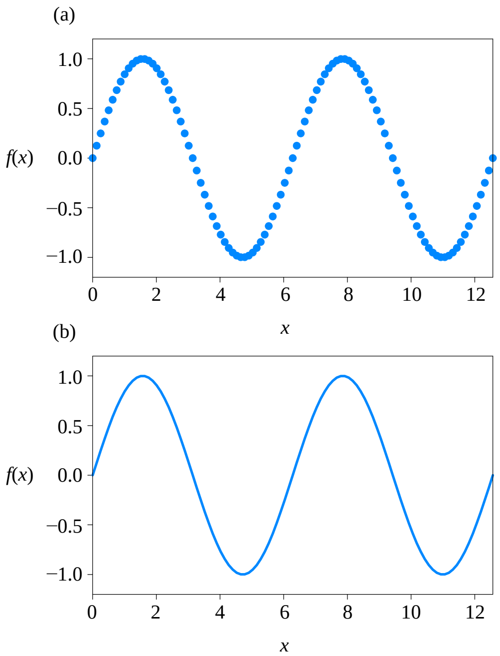

图 2. 颗粒度合适

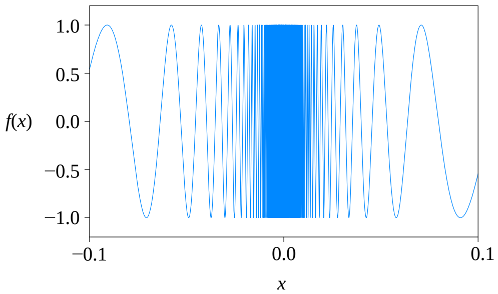

图 3. 特殊函数需要极其细腻的颗粒度

但是，颗粒度过高也不可取，也就是等差数列的公差过小，会增大计算量。这一点在一元函数上并不明显，但是用 numpy.meshgrid() 生成网格时，大家就会发现维数灾难 (curse of dimensionality)。

维数灾难是指在高维空间中，数据变得非常稀疏，而且距离变得非常远，使得许多常用的数据分析技术和算法无法有效地处理和分析数据。通俗点讲，假设我们有一个只有两个特征 (比如，鸢尾花花萼长度、宽度) 的数据集，我们可以很容易地将其可视化成二维平面上的点。但是如果我们有许多特征，比如几百个，那么我们将无法在三维或更高维空间中可视化数据。

此外，保持每个特征的采样数量，当维度增加时会导致数据量急剧增长。比如，单一维度的采样点数为 100，两个特征的网格点数就变成了 10000 (100^2)，三个特征的网格点数就增大到了惊人的 1000000 (100^3)。本书后文还会遇到这个问题。

注意，如果绘图采用对数坐标，建议采用 numpy.logspace() 生成数列。

阶跃

再次强调，在绘制线图时，默认散点之间两点顺序连线。这就意味着，任意顺序两点之间的线段是通过线性插值方法得到。

但是，有很多场合，我们需要避免“线性插值”，而采用阶跃方法绘制线图。

matplotlib.pyplot.plot() 函数本身可以设定阶跃绘图。此外，matplotlib.pyplot.step() 函数是专门绘制阶跃线图的函数。这个函数有三种设置：'pre'、'post'、'mid'。

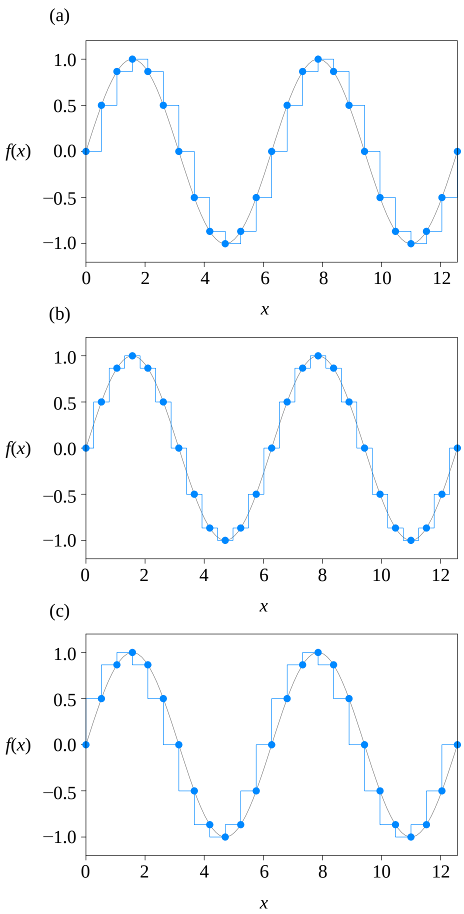

图 4. 三种阶跃

Jupyter 笔记 BK_2_Topic_4.02_2.ipynb 绘制图 4。

连接两点的插值方法有很多，《数据有道》第 5 章专门介绍。此外，本书后文会介绍贝塞尔曲线 (Bézier curve)。贝塞尔曲线是一种平滑曲线，在计算机图形学、工程和设计领域中广泛应用。

举个例子，贝塞尔二次曲线由三个控制点组成，其中两个控制点定义曲线的端点，第三个控制点定义曲线在端点之间的弯曲。

火柴图

火柴图 (stem plot)，也称火柴梗图、脊柱图，常用来可视化离散数据序列和趋势。火柴图垂直线所在横轴位置代表样本点的位置，圆点纵轴高度表示样本点的值。

本系列图册中，火柴图常用来可视化数列、离散随机变量概率质量函数 (Probability Mass Function, PMF)，如图 5 所示。

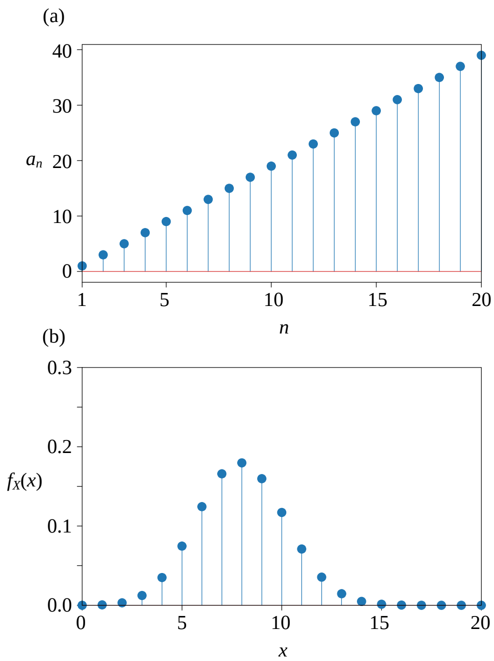

图 5. 火柴图可视化数列、概率质量函数

图 5 的两个子图都可以看成是离散函数。而前文的 $f(x) = \sin(x)$ 则是连续函数。离散函数、连续函数的主要区别在于自变量取值方式不同。离散函数自变量只能取有限或可数无限个值。也就是说，离散函数的函数图像是一系列散点。

例如，一个函数 $f(x)$ 表示了投掷一枚骰子后得到点数。因为骰子点数是有限的，所以自变量 x 的取值为 1、2、3、4、5、6 这几个离散值。而连续函数的定义域是一个连续的区间，比如 $(-\infty, \infty)$、$[0, 2]$。

Jupyter 笔记 BK_2_Topic_4.02_3.ipynb 绘制图 5。

参考线

水平线图中，我们经常需要添加水平或竖直参考线。图 6 所示两种不同绘制参考线的方法。

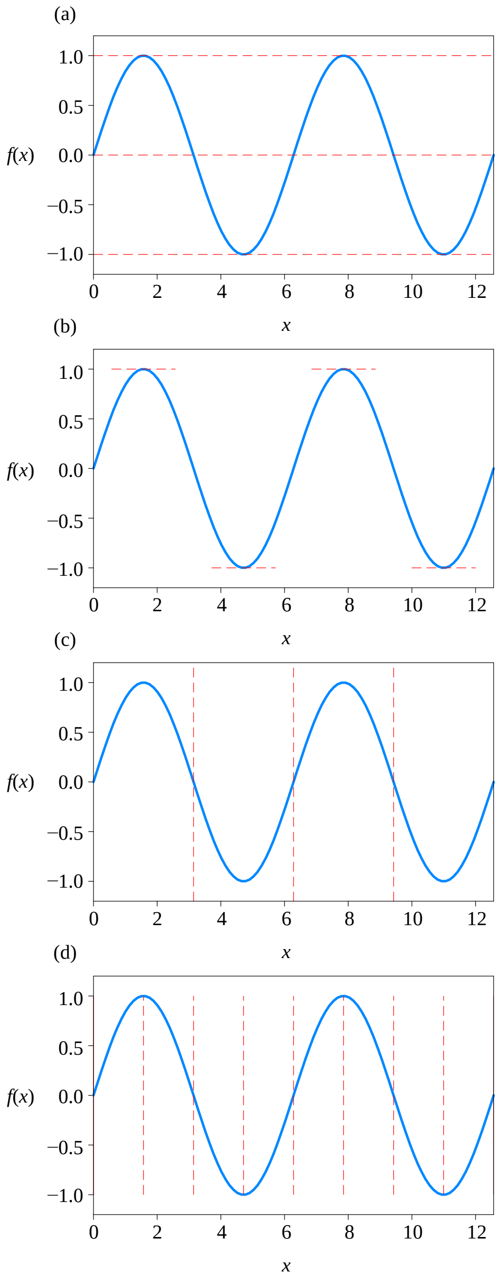

图 6. 两种绘制参考线的方法

Jupyter 笔记 BK_2_Topic_4.02_4.ipynb 绘制图 6。

使用面具

图 7 所示使用面具 (mask) 分段渲染线图。采用的方法和上一个话题一致。

Jupyter 笔记 BK_2_Topic_4.02_5.ipynb 绘制图 7。

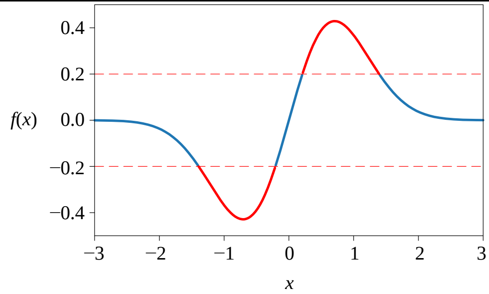

图 7. 分段渲染线图

交点

如图 8 所示，通过寻找 $f_1(x) - f_2(x)$ 的正负号变号的位置，我们可以估计 $f_1(x)$、$f_2(x)$ 的交点。

Jupyter 笔记 BK_2_Topic_4.02_6.ipynb 绘制图 8。

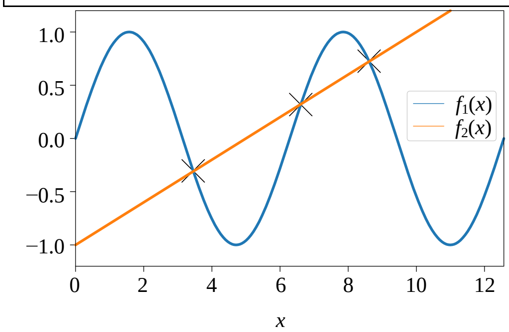

图 8. 可视化交点

极大、极小值

numpy.argmax()、numpy.argmin() 可以寻找数组中的极大、极小值，如图 9 所示。

Jupyter 笔记 BK_2_Topic_4.02_7.ipynb 绘制图 9。

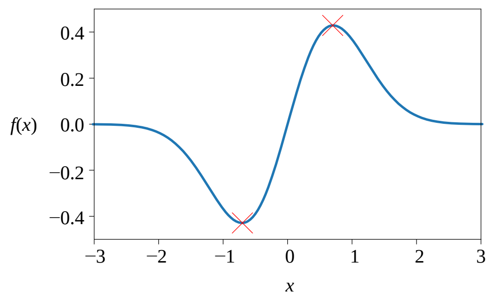

图 9. 可视化极值点

两点连线

如图 10 所示，我们还会经常可视化两点连线，比如散点和质心的连线，比如散点和投影点的连线。

Jupyter 笔记 BK_2_Topic_4.02_8.ipynb 绘制图 10。

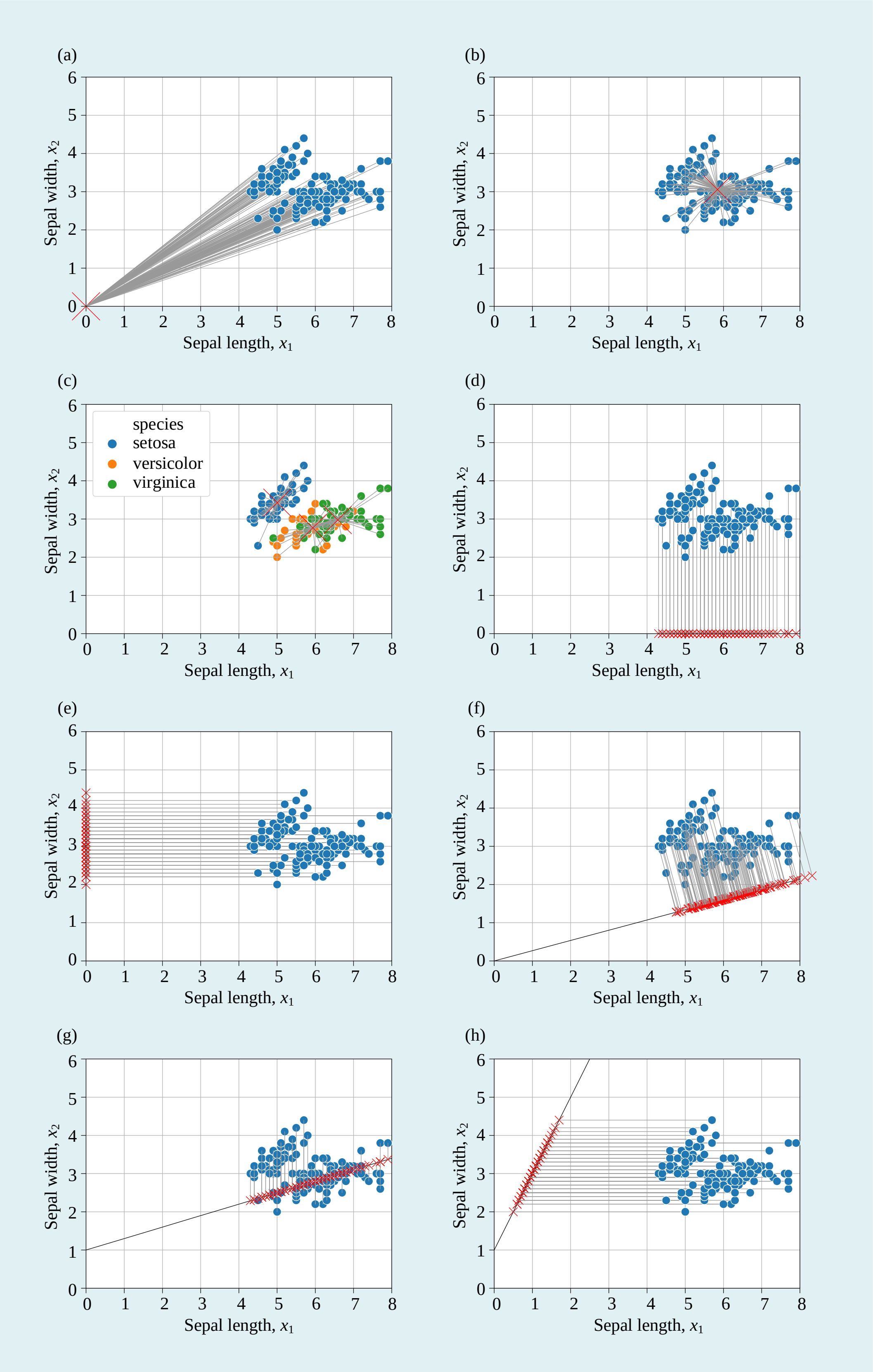

图 10. 几种两点连线的情况