

# 16

## 2D Contours

# 平面等高线

每条曲线上的点具有相同的数值



征服你自己，而不是世界。

*Conquer yourself rather than the world.*

—— 勒内·笛卡尔 (René Descartes) | 法国哲学家、数学家、物理学家 | 1596 ~ 1650



- ▶ `matplotlib.pyplot.contour()` 绘制等高线图
- ▶ `matplotlib.pyplot.contourf()` 绘制平面填充等高线
- ▶ `numpy.diag()` 如果 `A` 为方阵，`numpy.diag(A)` 函数提取对角线元素，以向量形式输入结果；如果 `a` 为向量，`numpy.diag(a)` 函数将向量展开成方阵，方阵对角线元素为 `a` 向量元素
- ▶ `numpy.dot()` 计算向量标量积。值得注意的是，如果输入为一维数组，`numpy.dot()` 输出结果为标量积；如果输入为矩阵，`numpy.dot()` 输出结果为矩阵乘积，相当于矩阵运算符@
- ▶ `numpy.linalg.det()` 计算行列式值
- ▶ `numpy.linalg.inv()` 矩阵求逆
- ▶ `numpy.meshgrid()` 创建网格化数据
- ▶ `numpy.sqrt()` 计算平方根
- ▶ `numpy.vstack()` 返回竖直堆叠后的数组
- ▶ `numpy.zeros_like()` 用来生成和输入矩阵形状相同的零矩阵
- ▶ `sympy.diff()` 求解符号导数和偏导解析式
- ▶ `sympy.exp()` 符号自然指数
- ▶ `sympy.lambdify()` 将符号表达式转化为函数

本 PDF 文件为作者草稿，发布目的为方便读者在移动终端学习，终稿内容以清华大学出版社纸质出版物为准。

版权归清华大学出版社所有，请勿商用，引用请注明出处。

代码及 PDF 文件下载：<https://github.com/Visualize-ML>

本书配套微课视频均发布在 B 站——生姜 DrGinger：<https://space.bilibili.com/513194466>

欢迎大家批评指教，本书专属邮箱：[jiang.visualize.ml@gmail.com](mailto:jiang.visualize.ml@gmail.com)

## 16.1 等高线

平面等高线、三维等高线是“鸢尾花书”中非常重要的可视化方案。我们常用等高线可视化二元乃至多元函数、概率密度函数、机器学习中的决策边界等等。

这个话题主要介绍平面等高线，本书后续将专门介绍三维等高线。

### 等高线原理

等高线图是一种展示三维数据的方式，其中相同数值的数据点被连接成曲线，形成轮廓线。

形象地说，如图 1 所示，二元函数相当于一座山峰。在平行于  $x_1x_2$  平面在特定高度切一刀，得到的轮廓线就是一条等高线。这是一条三维空间等高线。然后，将等高线投影到  $x_1x_2$  平面，我们便得到一条平面等高线。

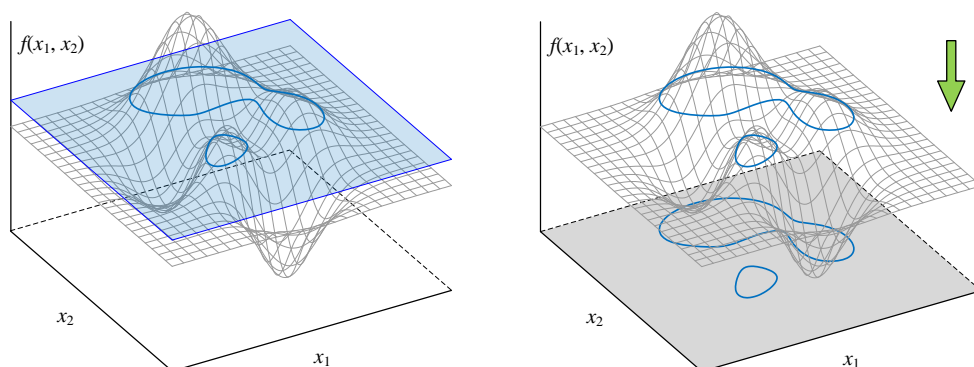


图 1. 平行  $x_1x_2$  平面切  $f(x_1, x_2)$ ，然后再投影到  $x_1x_2$  平面

一系列轮廓线的高度一般用不同的颜色或线型表示，使得我们可以通过视觉化方式看到数据的分布情况。如图 2 所示，将一组不同高度的等高线投影到平面便得到如图 3 所示的平面等高线。图 3 的右侧还增加了色谱条，用来展示不同等高线对应的具体高度。这一系列高度可以是一组用户输入的数值。大家可能已经发现，等高线图和海拔高度图原理完全相同。类似的图还有，等温线、等降水线、等距线等等。

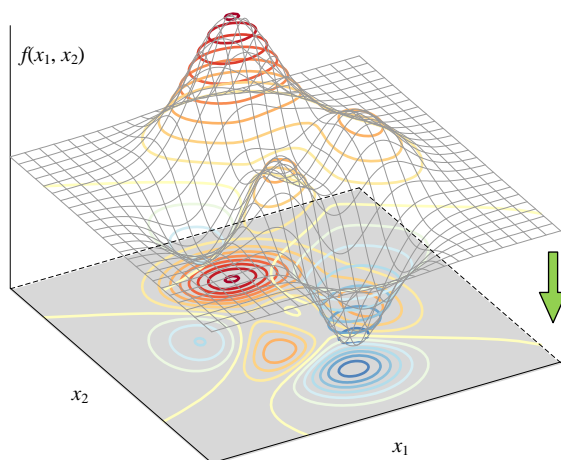


图 2. 将不同高度 (值) 对应的一组等高线投影到  $x_1x_2$  平面

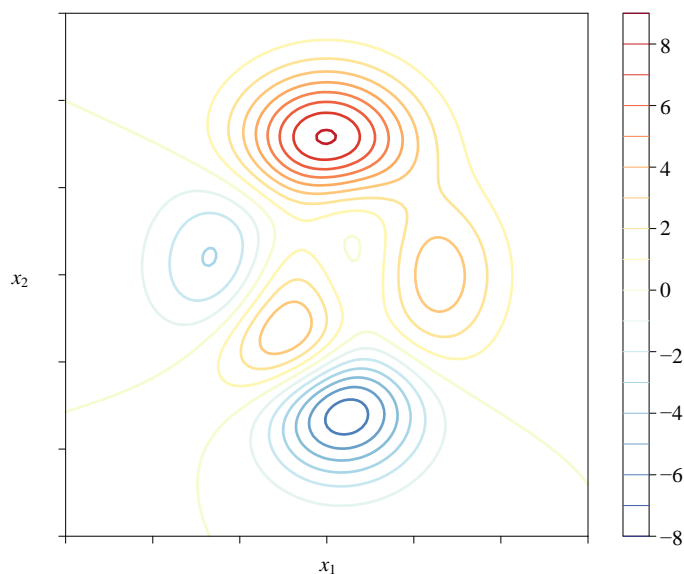


图 3. 平面等高线

### 步骤

具体来说，Matplotlib 中绘制等高线图需要以下步骤：

**准备数据：**等高线图需要的数据是一个二维数组，其中每个元素都表示一个点的数值。通常这个二维数组被称为“网格”。

**计算轮廓线：**Matplotlib 会通过对数据进行插值，计算出一组轮廓线的值，并把它们绘制在二维平面上。轮廓线的数量取决于我们指定的等高线的数量。

**绘制轮廓线：**Matplotlib 会根据轮廓线的高度在不同的颜色或线型中表示，使得我们可以通过视觉化方式看到数据的分布情况。通常使用 `plt.contour()` 函数进行绘制。

### 等高线设置

Matplotlib 还提供了许多与等高线图相关的函数和选项，例如设置轮廓线的样式、标签、标记等。这些选项可以帮助我们更好地展示数据。

图 4 所示为给每条等高线增加高度注释。图 5 所示为单色等高线，matplotlib 会用虚线代表负值。

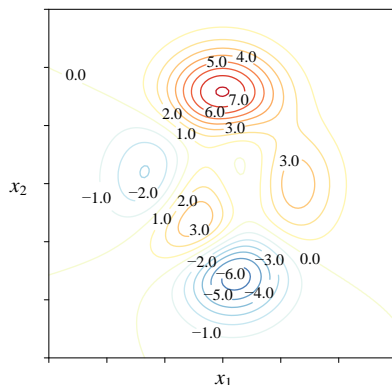


图 4. 给等高线加注释

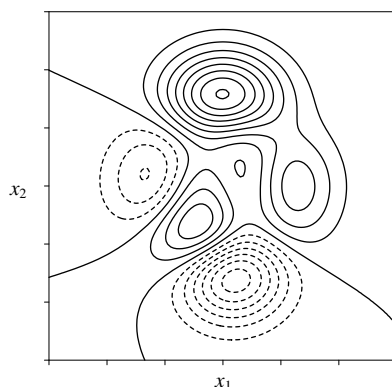


图 5. 单色等高线

### 填充等高线

`contourf` 是 Matplotlib 库中用于绘制等高线填充图的函数。其原理是通过将数据转换为等高线线段的集合，然后通过填充线段之间的空隙来创建颜色填充图。具体地说，`contourf` 函数首先根据数据生成一组等高线，这些等高线可以使用 `contour` 函数绘制。然后 `contourf` 函数会根据这些等高线，将图像中每个等高线所围成的区域填充上颜色。填充的颜色根据指定的 colormap 进行选择，可以通过设置 `cmap` 参数来控制。

如图 6 所示，在三维空间中，我们可以把填充等高线想象成是“梯田”。每个颜色代表一定的高度区间。将图 6 填充等高线画在水平面上，我们便得到图 7。图 7 中，我们还绘制一条指定高度 (0.0) 的等高线。

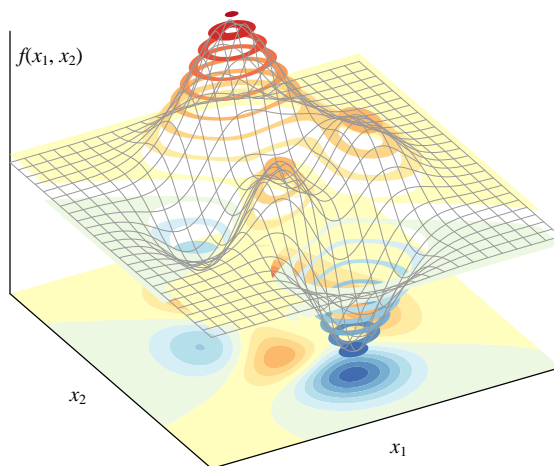


图 6. 填充等高线原理

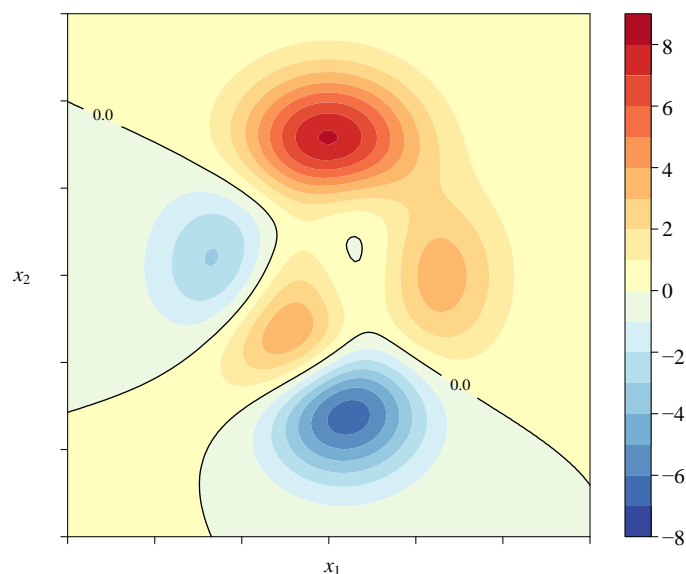


图 7. 平面填充等高线

此外，`contourf` 还可以通过调整填充级别的数量和间隔来控制填充效果，可以通过设置 `levels` 参数来控制填充级别的数量和大小。此外，还可以通过设置 `alpha` 参数来控制填充颜色的透明度，以及通过设置 `extend` 参数来控制 `levels` 范围之外的值应如何处理。

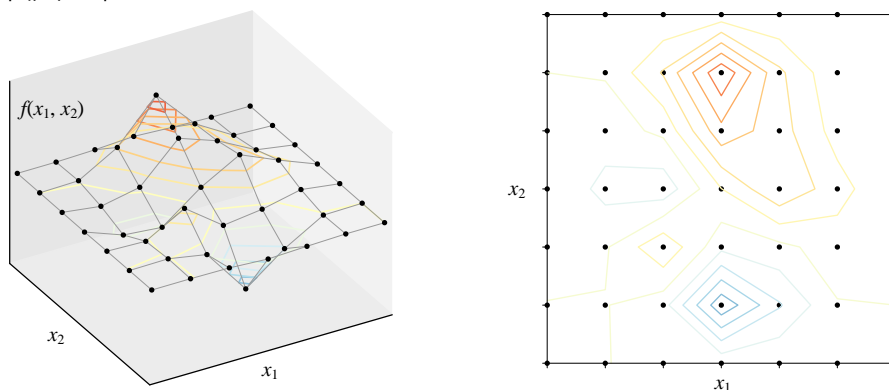


Jupyter 笔记 BK\_2\_Ch16\_1.ipynb 绘制本话题前几幅图。

## 16.2 颗粒度

前文提过，绘制连续等高线基于插值，因此等高线是否“平滑”，也取决于网格数据的颗粒度是否足够细腻。

如图 8 所示，绘制等高线采用的网格数据显然过于粗糙，这导致不管是三维等高线，还是平面等高线都非常毛糙。



本 PDF 文件为作者草稿，发布目的为方便读者在移动终端学习，终稿内容以清华大学出版社纸质出版物为准。

版权归清华大学出版社所有，请勿商用，引用请注明出处。

代码及 PDF 文件下载：<https://github.com/Visualize-ML>

本书配套微课视频均发布在 B 站——生姜 DrGinger：<https://space.bilibili.com/513194466>

欢迎大家批评指教，本书专属邮箱：[jiang.visualize.ml@gmail.com](mailto:jiang.visualize.ml@gmail.com)

图 8. 颗粒度粗糙

图 9 则采用更为细腻的网络数据。显然，插值得到的三维等高线、平面等高线看上去更加平滑。

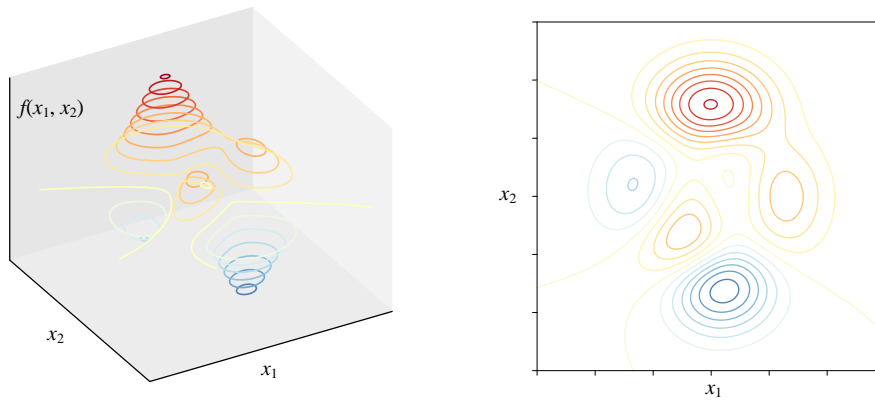


图 9. 颗粒度细腻



Jupyter 笔记 BK\_2\_Ch16\_2.ipynb 绘制图 8、图 9。

网格不是越密越好。网格能够取得多密，还受到算力制约。比如，图 10 所示马氏距离、高斯分布概率密度等高线，进一步提高网格密度会受到内存空间限制。

本书后续还会介绍等高线的更广泛的用途。



Jupyter 笔记 BK\_2\_Ch16\_3.ipynb 绘制图 10。

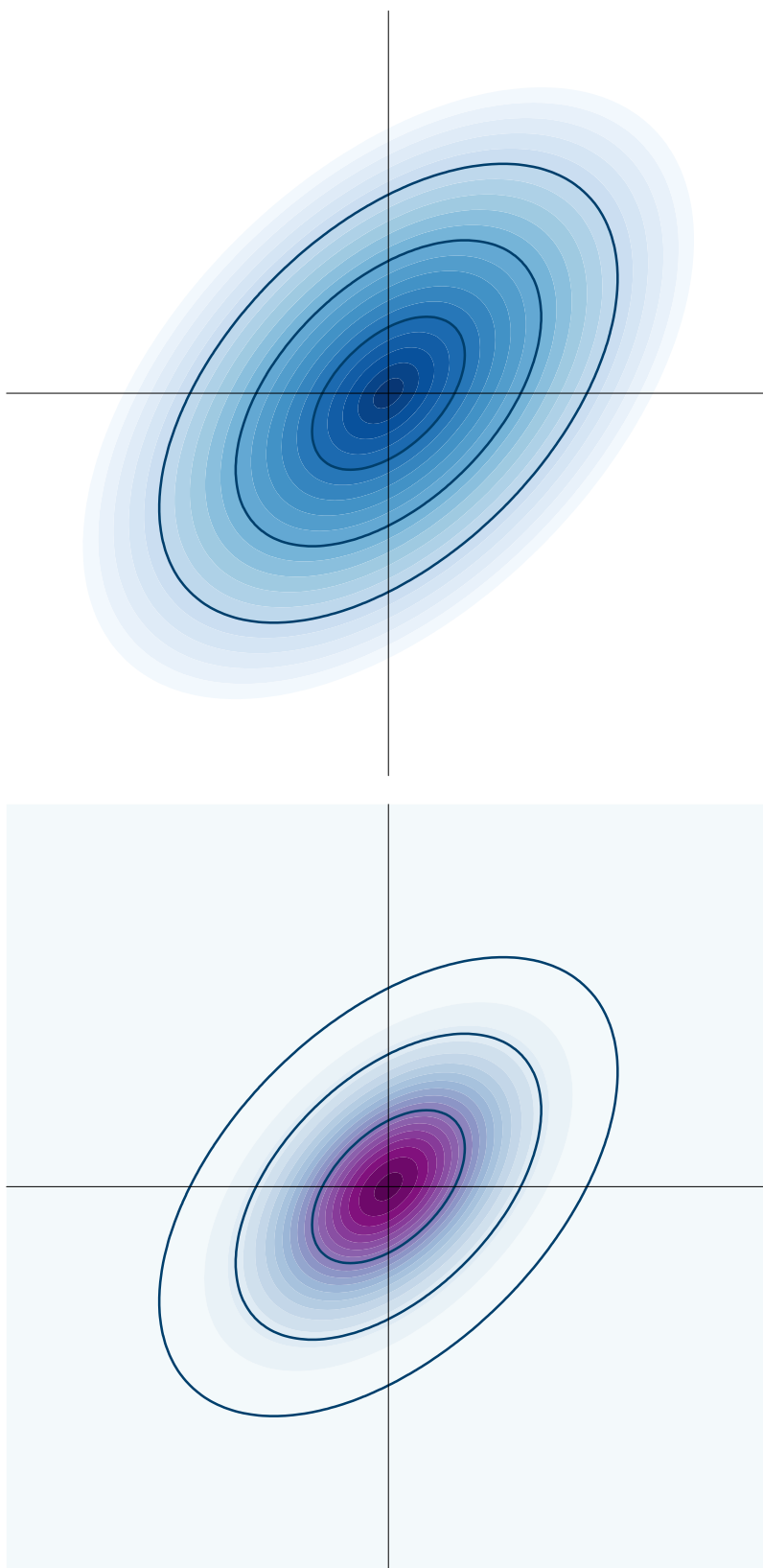


图 10. 马氏距离、高斯概率密度函数等高线