

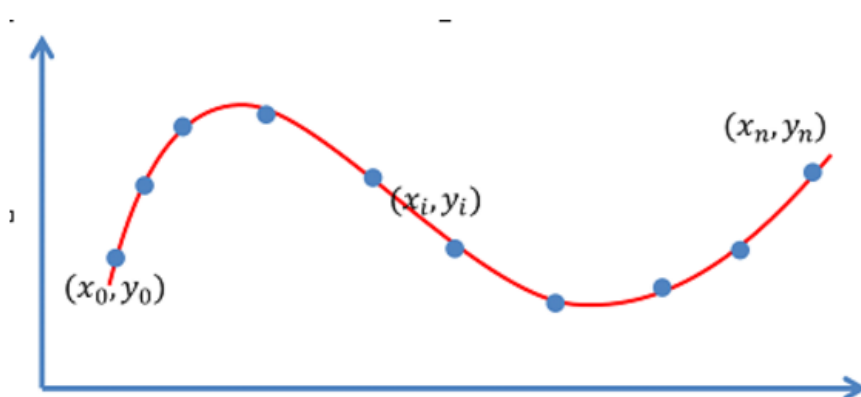
homework 1

问题

Input: 已知平面内 n 个点 $P_j(x_j, y_j), j = 1, 2, \dots, n$ 。

Output: 拟合这些点的函数。

要求：实现不同的拟合方法，并进行比较。输入点集可以进行交互式鼠标指定，或者其他方法生成。

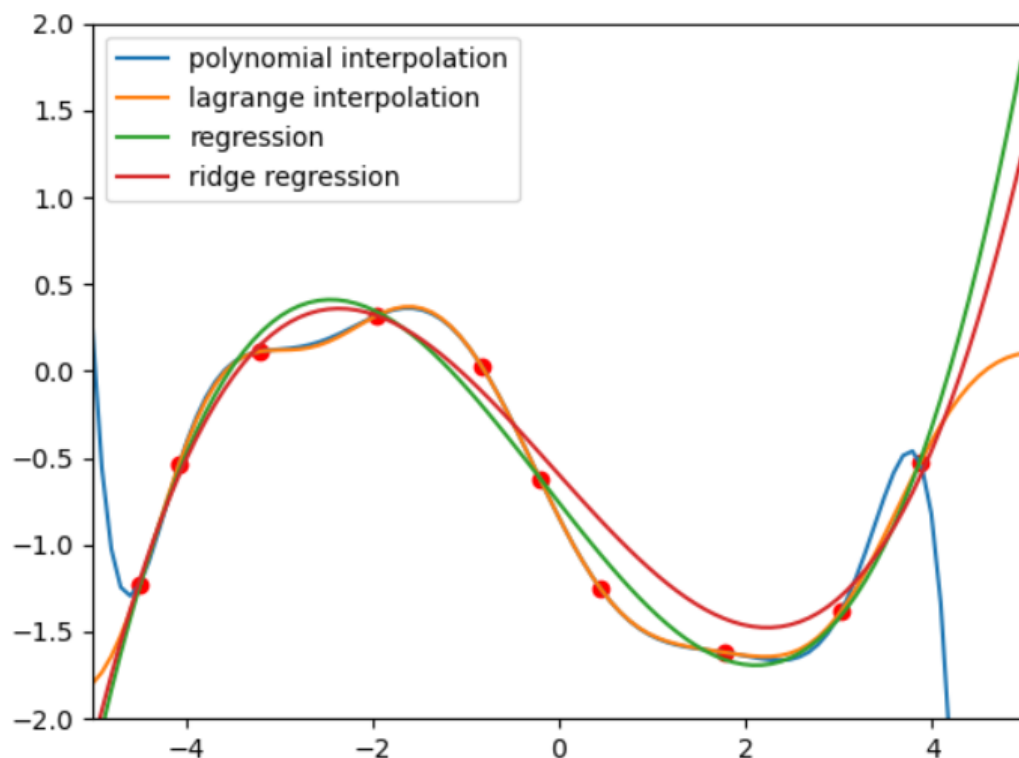


- 实现语言：python，使用matplotlib，numpy框架
- 实现内容：交互式放置需要拟合的点，并使用多项式插值，高斯插值，普通回归，岭回归来进行函数拟合
- 使用方法：输入下面指令：

```
python fit.py
```

弹出一个带有坐标系的窗口，点击坐标系就会记录需要拟合的点。点的放置结束后，关闭该窗口，即可进行拟合，并会弹出一个窗口可视化拟合结果。

- 注意： n 个点，多项式插值中，系数最高次等于 $n - 1$ ，而在普通回归与岭回归中，需要指定 n ，默认为4，也就是三次曲线拟合
- 拟合结果：



从上图可以看出，插值算法误差为0，但是由于点的个数比较多，插值得到的曲线非常奇怪。绿色的曲线与红色曲线是通过回归得到的，其中红色曲线使用了岭回归，它使用了二范式正则化，可以避免对数据的过拟合。

在实现过程中，我专门写了一个函数来解线性方程组。现在有线性方程组：

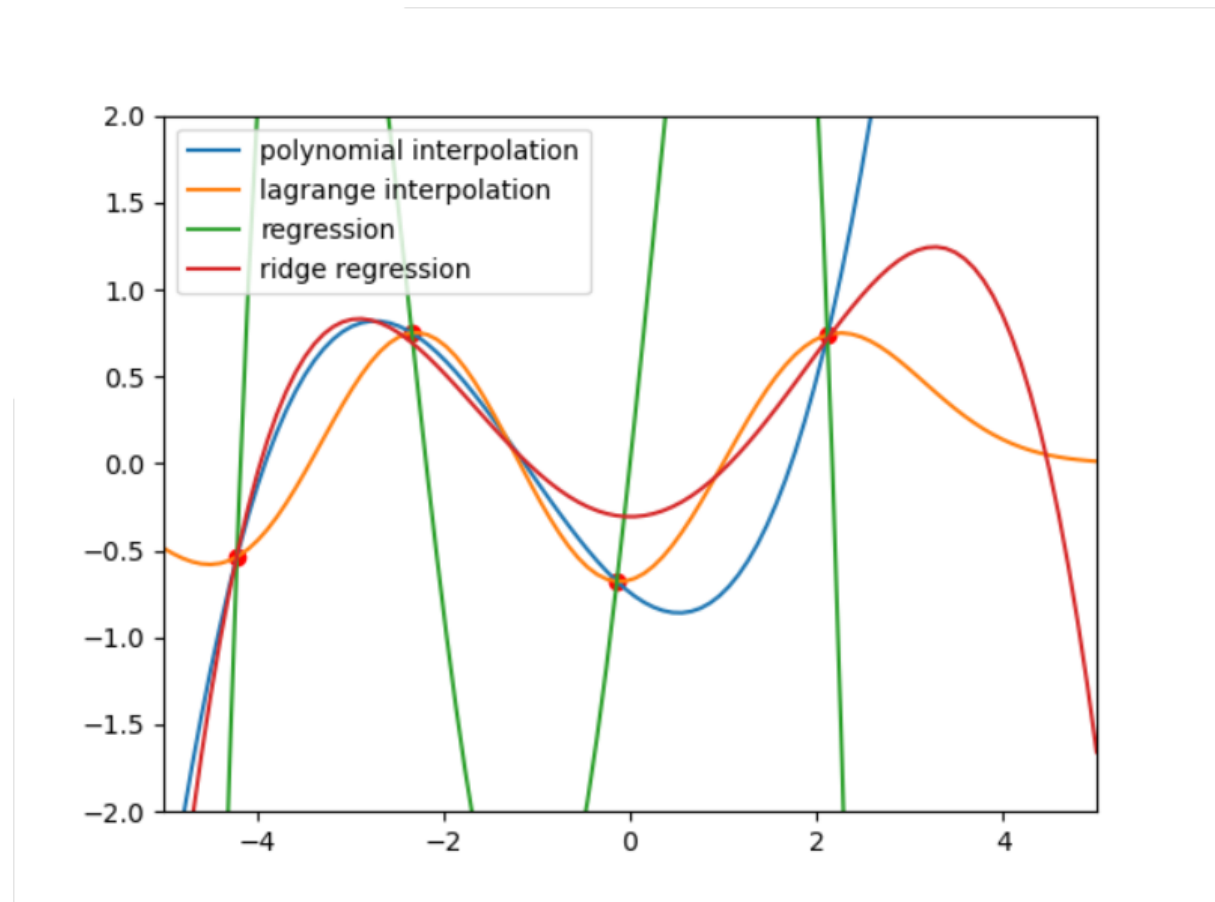
$$Ax = b$$

其中 A 是一个 $m \times n$ 的矩阵。如果 $m > n$ ，说明方程个数大于变量个数，是超定方程组，可以使用SVD分解来求得最小二乘解，如果方程个数小于变量个数，也就是 $m < n$ ，该方程组为欠定方程组，若 $m = n$ ，该方程组有唯一解（这个描述并不准确，实际上还需要根据矩阵 A 的秩来决定）。

如果我们将回归拟合中 n 的个数调大，也就是参数个数变多，甚至超过了方程的个数，那么解方程需要约束条件。我在代码中使用的约束条件是将前面多余的参数置0（也可将后面的参数置0，也就是将高次数的参数变量置0，这样得到结果将与多项

式插值一致，为了与多项式插值区分，我选择将前面低次数的参数置0），这时候回归也可以得到插值般的精确解。

如果设定的回归参数个数为5，而需要拟合的点只有四个，得到下图：



可以看到用高次多项式来回归拟合会得到很夸张的曲线，这时候岭回归与插值方法显得更好。