实验一：KNN算法

- ## KNN算法简述

  KNN全称K Nearest Neighbors，K个距离最近的邻居。它是一个经典的监督学习分类算法。顾名思义，**在预测一个新的值X的分类时，与它距离最近的K个点中占比最高的分类即为该点的预测分类。**显然，K应该取奇数，避免距离最近的偶数个点中各个分类的点数量相同导致无法分类。定义中，**距离**有很多选择，本实验选取**欧几里得距离**，也即**L2范数。**

- ## 实验用数据集简述

  本实验采用Python中经典的机器学习数据包sklearn中的iris dataset、wine dataset、breast_cancer dataset作为数据集。三个数据集的shape如下表所示：

Dataset_Name	Iris	Wine	Breast_Cancer
Total Sample Number	150	178	569
Feature Dimension	4	13	30
Category Number	3	3	2
Separate Sample Number	50/50/50	59/48/31	455/114

  实验中可以调节的超参数有：采样率**sample rate**（assert 0<sample reta<=1）、近邻范围**K**（K应取奇数）。

  实验中对于每一数据集，test_size设为0.2，即数据集中的**20%作为TestSet，80%作为TrainingSet。**严格来说，KNN并不是一个神经网络，没有训练参数的过程，因而也没有TrainingSet、TestSet的概念。此处**TrainingSet指已知点，TestSet指待预测的点。**sklearn中的train_test_split函数中实现了对数据集的shuffle，保证了**样本数据独立同分布。**

- ## 实验结果

  尝试不同的数据集与超参数组合，测试KNN算法预测的准确性。由于测试数据的维度大于三维，无法将分类过程进行直观的图像绘制，故使用表格记录实验结果。sklearn中的train_test_split函数每次采取不同的shuffle，这对预测结果有影响。本实验中，为了考量算法在**不同数据集下的鲁棒性**，取定超参数K和Sample Rate后，算法执行三次，**取三次Precision Rate的平均值**作为最终的Precision Rate来**评估算法表现。**

  实验结果如下：

  Iris Dataset: 150 total samples，50 for each category

Precision Rate	K=1	K=3	K=5	K=7
Sample Rate = 0.2	99.44%	100%	100%	99.44%
Sample Rate = 0.5	97.78%	97.78%	100%	97.78%
Sample Rate = 1.0	98.89%	100%	95.56%	96.67%

  每种参数组合下的算法的三次运行结果相差不大（20%以内），可以认为**该算法在Iris Dataset上鲁棒性较好**；

Wine Dataset: 178 total samples，59 for category 1，48 for category 2，31 for category 3。

Precision Rate	K=1	K=3	K=5	K=7
Sample Rate = 0.2	57.14%	71.43%	57.14%	71.43%
Sample Rate = 0.5	64.81%	62.97%	75.93%	87.03%
Sample Rate = 1.0	57.14%	71.43%	38.10%	57.14%

其中，当Sample Rate=0.2时候，算法三次运行的结果差距较大（极值差距超过40%），可以认为该算法在稀疏的Wine Dataset上鲁棒性较差。

- **Breast Cancer Dataset:** 569 total samples，455 for category 1，114 for category 2。

Precision Rate	K=1	K=3	K=5	K=7
Sample Rate = 0.2	84.06%	91.30%	86.96%	94.20%
Sample Rate = 0.5	90.64%	91.23%	96.49%	88.89%
Sample Rate = 1.0	84.06%	91.30%	86.96%	93.20%

每种参数组合下的算法的三次运行结果相差不大（20%以内），可以认为**该算法在Iris Dataset上鲁棒性较好。**

综上所述，可以得到关于KNN算法的如下结论：

1. **KNN算法适合拓扑空间中相距比较远的聚类数据。**若不同聚类之间的数据在拓扑空间中相距较近，KNN算法表现将会很差，例如Wine Dataset。
2. 在给定数据集上，**K的选取不宜过大或过小。**若过小，算法具有较大偶然性；若过大，则可能包括了过多的其它聚类中的点。例如在上面三个数据集中，算法准确率最高时的K取值是3或5而非1或7。
3. 并不是已知点越多预测效果越好。例如观察上面三个数据集，取样率为1时算法表现不是最好的。这是因为点过多时会导致不同聚类间的最短距离会变短。

实验二：高阶多项式回归的欠拟合/过拟合分析

- ## 欠拟合与过拟合

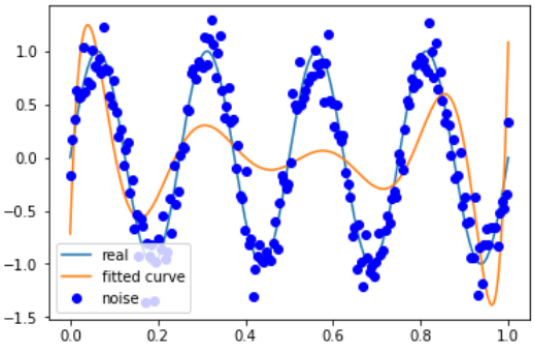
线性回归是指根据已知的离散数据拟合曲线，基于曲线对未知数据进行预测。欠拟合是指在回归过程中，拟合的曲线与已知数据符合程度很低，残差平方和很大；过拟合是指曲线为了拟合所有已知数据而过度调整参数，导致模型在Training set上表现极佳而在Test set上表现很差。

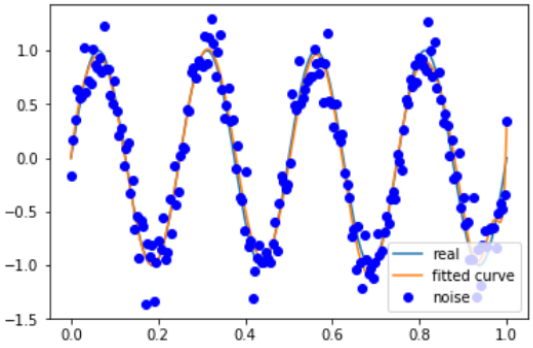
- ## 实验用数据集简述

本实验采取n次多项式对正弦曲线进行拟合，已知数据固定为200个。为增强模型的鲁棒性，对已知数据加上随机的高斯噪声。实验中可供调节的超参数有：**正弦曲线的频率ω、多项式的最高次数n、高斯噪声的方差σ。**将**残差平方和**定义为**J。**

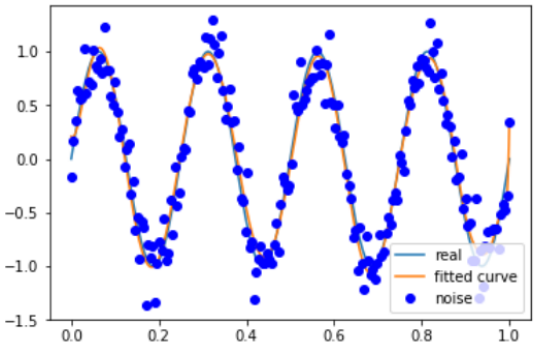
- **实验结果**

ω=8、σ=0.2： J分别取273.030 4.67 4.618 16.878

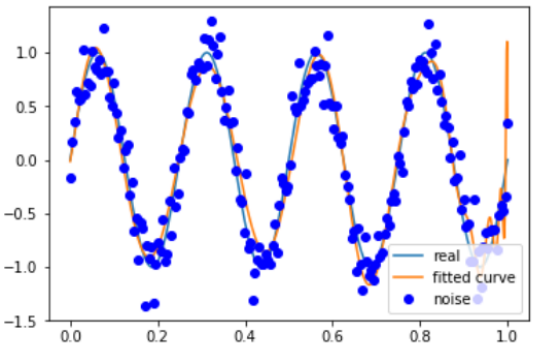




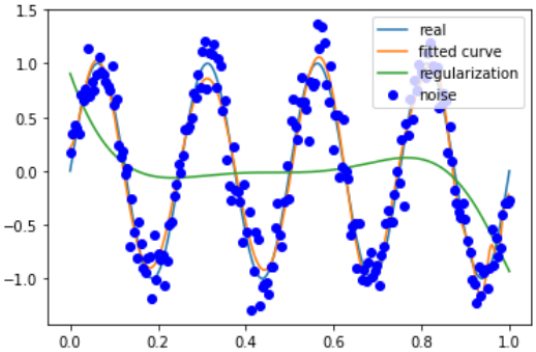
n = 10,J = 273.030



n = 30,J = 4.670



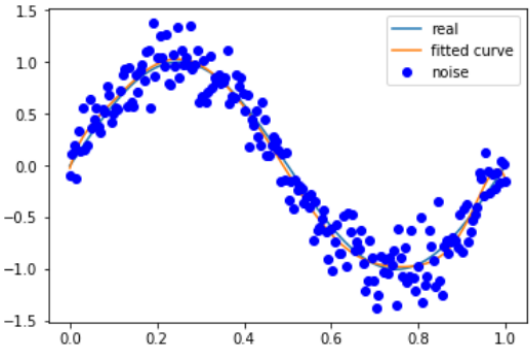
n = 50,J = 4.618

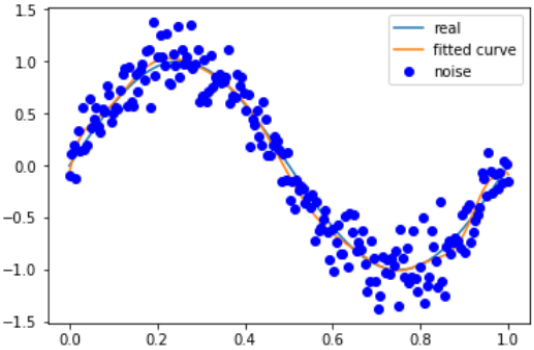


正则化

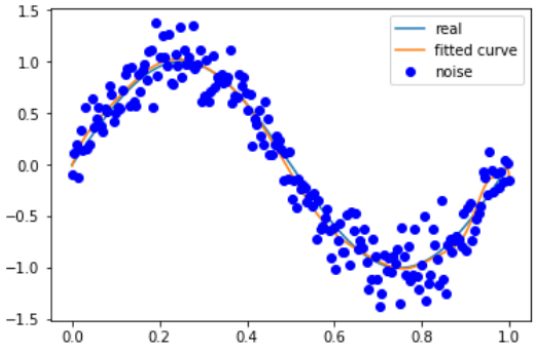
可以看出，当待拟合函数较为复杂时，n取较小值拟合效果很差，残差平方和J值很大；n取适当值，拟合效果最好；当n过大时，J值不再下降，甚至可能升高，此时模型进入过拟合阶段。加入正则化项后，过拟合程度减小，模型优化。

ω=2、σ=0.2：J分别取2.601 3.177 2.838 3.257

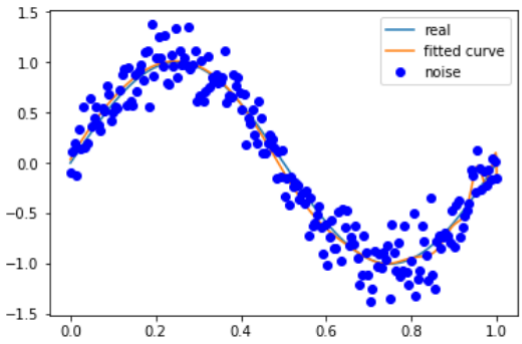




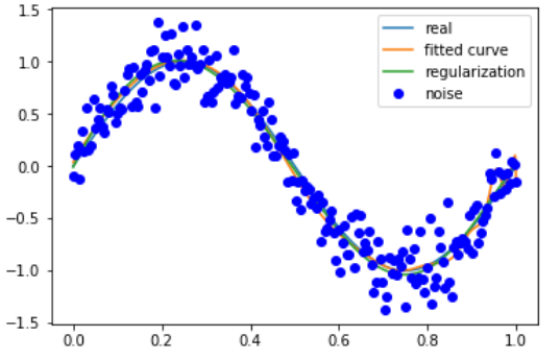
n = 10,J = 2.601



n = 30,J = 3.177



n = 50,J = 2.838



n = 180,J = 3.257

可以看出，当待拟合函数较为简单时，n取较小值时模型的J值已经很难继续下降。继续增大n会导致过拟合。

综上，待拟合曲线越复杂，所需要的拟合多项式次数越高；但当多项式次数上升到一定值后，残差平方和很难继续下降，此时模型进入过拟合阶段。加入正则化项可以减轻过拟合，使模型优化。