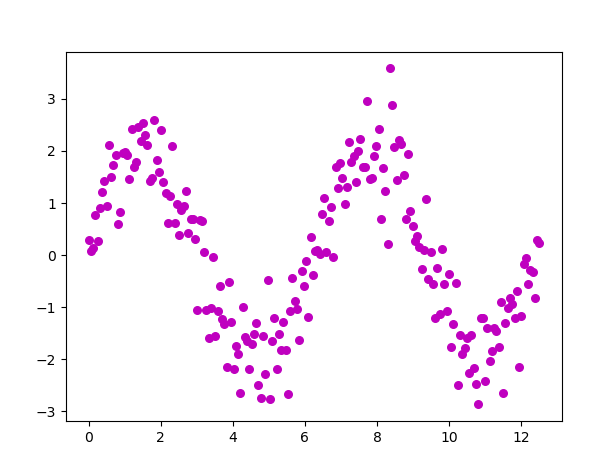
# 分类与与回归树

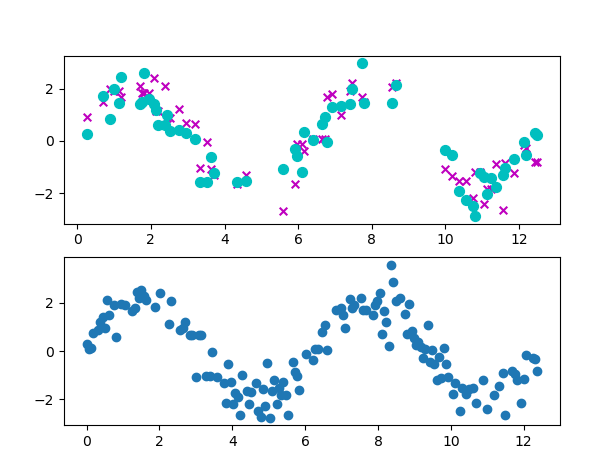
## 数据集

数据集用了2个，回归数据集和蛋白质的二分类数据集。

回归数据集：



## 实验结果



下图为训练数据的结果，上图为测试数据集，x为预测值，o为真实值。

测试集均方损失为：0.5925848568401221

对于蛋白质的分类的数据集（分类准确率）：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Train  Test | X1 | X2 | X3 | X4 |
| X1 | 1 | 0.96 | 0.61 | 0.71 |
| X2 | 0.91 | 1 | 0.65 | 0.74 |
| X3 | 0.71 | 0.65 | 1 | 0.66 |
| X4 | 0.70 | 0.65 | 0.67 | 1 |

## 实验过程

#### 2.1 实现CART(classification and regression tree)

1、定义分裂准则

（1）定义均方误差，用来做回归。def mse

（2）定义信息熵，用来做分类。def entropy

（3）定义基尼指数，Gini(D)反映了从数据集D中随机抽取两个样本，其类别标记不一致的概率。Gini(D)越小，即纯度越高。

def mse

def entropy

def gini

2、定义结点类，属性包含left,right,rule三个参数。rule[0]指的是结点的属性，rule[1]指的是阈值，（阈值理解为类别所达到的纯度，达到某个纯度后即可不再分裂）

class Node:

3、定义叶子类，属性包含value，如果是分类，value是一组类别的概率，如果是回归，则是区域的平均值。

class Leaf:

4、定义决策树。属性包含classifier,criterion。

classifier: 默认为True，默认做分类，做回归时选择False。

criterion:{mse,entropy,gini}，选择分裂时所用的准则。

class DecisionTree:

##### 4.1定义的函数

1、定义拟合函数，对数据集拟合生成二元决策树。

2、定义生长树的函数，递归生长，停止条件为所有类别为同类，则返回叶子

3、定义分割函数，寻找最优的分裂准则，即寻找到最优属性进行分割。返回分裂属性的索引及分裂阈值。

4、定义不纯增益，对于给定属性计算不纯的增益。

5、定义遍历函数（用于对新来点的类别预测）

6、定义预测函数

1. 定义拟合函数(def fit)：

即从根结点开始生长树。

self.root=self.\_grow(X,Y)

1. 定义生长树函数(def \_grow)：

确定停止条件，在类别都是同一类的时候就返回叶子。

if len(set(Y))==1:

if self.classifier:

prob=np.zeros(self.n\_classes)

prob[Y[0]]=1.0

return Leaf(prob) if self.classifirer else Leaf(Y[0])

获取当前最优的分割属性以及阈值，用贪心的方式选择最优的分割。

feat,thresh=self.\_segment(X,Y,feat\_idxs)

l=np.argwhere(X[:,feat]<=thresh).flatten()

left=self.\_grow(X[l,:],Y[l])

right同理

return Node(left,right,(feat,thresh))

1. 定义分割函数def \_segment(X,Y,feat\_idxs)

遍历feat\_idxs:

获取X的每列样本值

确定阈值和信息增益（根据增益获得属性索引并返回）

vals=X[:,i]

levels=np.unique(vals)

thresholds=(levels[:-1]+levels[1:])/2 if len(levels)>1 else levels

gains=np.array([self.\_impurity\_gain(Y,t,vals) for t in thresholds])

1. 定义不纯增益，对于给定属性计算不纯的增益（用来进行预剪枝的判断）

def \_impurity\_gain(self,Y,split\_thresh,feat\_values):

IG(split) = loss(parent) - weighted\_avg[loss(left\_child), loss(right\_child)]

判断父结点与左右子结点的loss

1. 定义遍历函数，一直到子结点结束（def \_traverse）

先判断该结点是不是子结点，不是子结点的话通过阈值判断是往左递归遍历还是往右递归遍历。

返回叶子结点的值。(return node.value)该结点已经是叶子结点了所以有值。

1. 定义预测函数(def predict(X))

将X中的每个样本，用于traverse函数的输入，将最后得出的值存入np.array中。

以探索的模式（超参数）

决策树的最大深度（防止过拟合）。

网格搜索

交叉验证

ROC、AUC