第四次实验报告

——————梯度下降算法实验求最小值

SA20218084 林睿江

一、实验目的

完成一个梯度下降算法来解决线性规划的优化问题。利用梯度下降算法思想进行求函数最小值方法的构造，然后求输入函数的最小值。

二、实验原理

提供一个数据集，样本数 n 为 16087，样本数特性 d 的值是 10013。假设 x 是输入特征矩阵 y，是对应的响应向量。我们使用线性模型来拟合数据，因此我们可以将优化问题表述为

$$\arg\min_{\mathbf{w}} \frac{1}{n}\|y - \bar{x}\mathbf{w}\|_2^2,$$

使用梯度下降算法求解上面的优化问题，迭代到|f(w_k) - f(w_0*)|< 0.1, where f(w) = (1/n) ||y - x_w|| 2。

梯度下降法(gradient descent)是一个最优化算法，通常也称为最速下降法。常用于机器学习和人工智能当中用来递归性地逼近最小偏差模型。顾名思义，梯度下降法的计算过程就是沿梯度下降的方向求解极小值（也可以沿梯度上升方向求解极大值）。

其迭代公式为$a_{k+1} = a_k + \rho_k \bar{s}^{(k)}$,其中$\bar{s}^{(k)}$ 代表梯度负方向，ρ_k表示梯度方向上的搜索步长。梯度方向我们可以通过对函数求导得到，步长的确定比较麻烦，太大了的话可能会发散，太小收敛速度又太慢。一般确定步长的方法是由线性搜索算法来确定，即把下一个点的坐标 ak+1 看做是的函数，然后求满足 f(ak+1)的最小值的 即可。

因为一般情况下，梯度向量为 0 的话说明是到了一个极值点，此时梯度的幅值也为 0.而采用梯度下降算法进行最优化求解时，算法迭代的终止条件是梯度向量的幅值接近 0 即可，可以设置个非常小的常数阈值。

三、实验步骤

1.先对实验训练数据进行预处理，将 180 列数据转换成对应的 60 个碱基。

即为 getData()函数。

2.再将训练数据集运用递归的方法建立决策树。

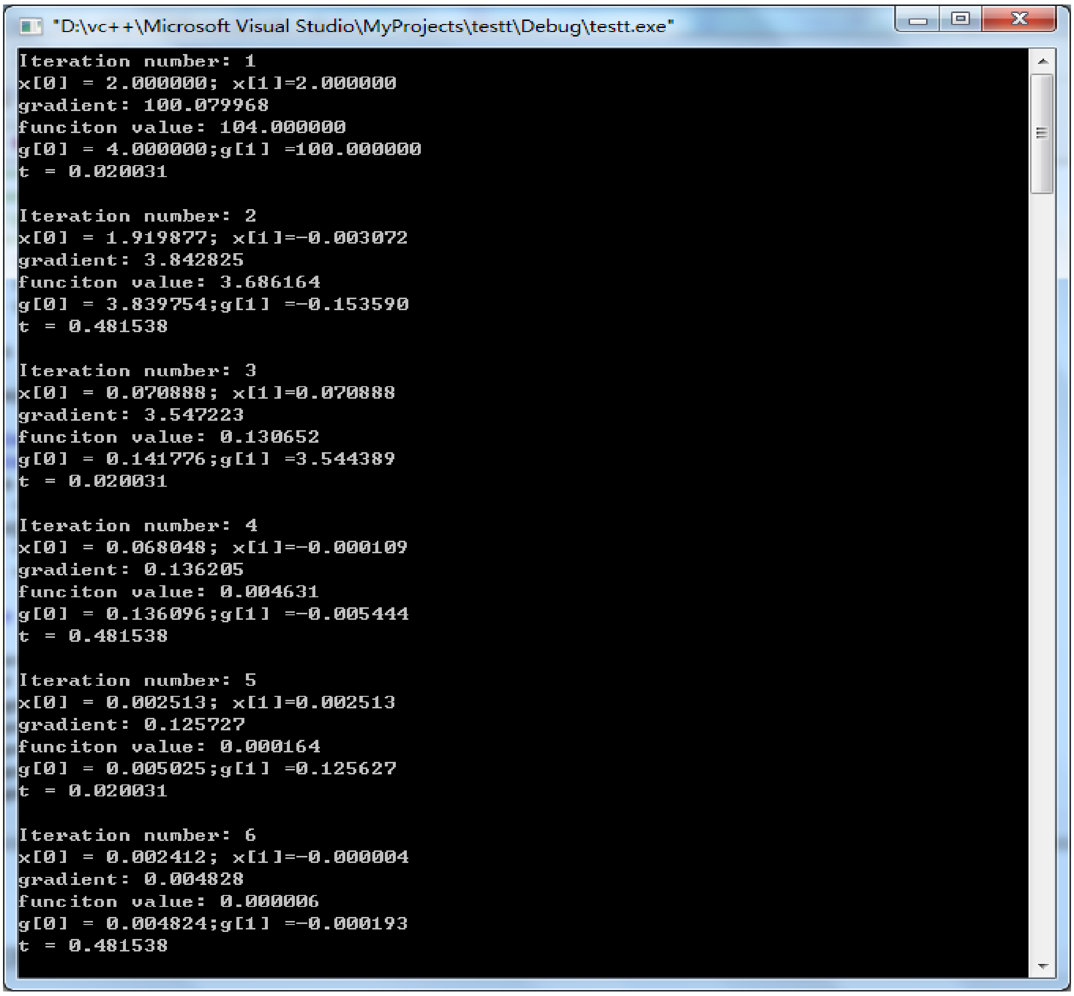
即调用 id3Tree()函数。

3.将所得到的决策树保存，并绘图。

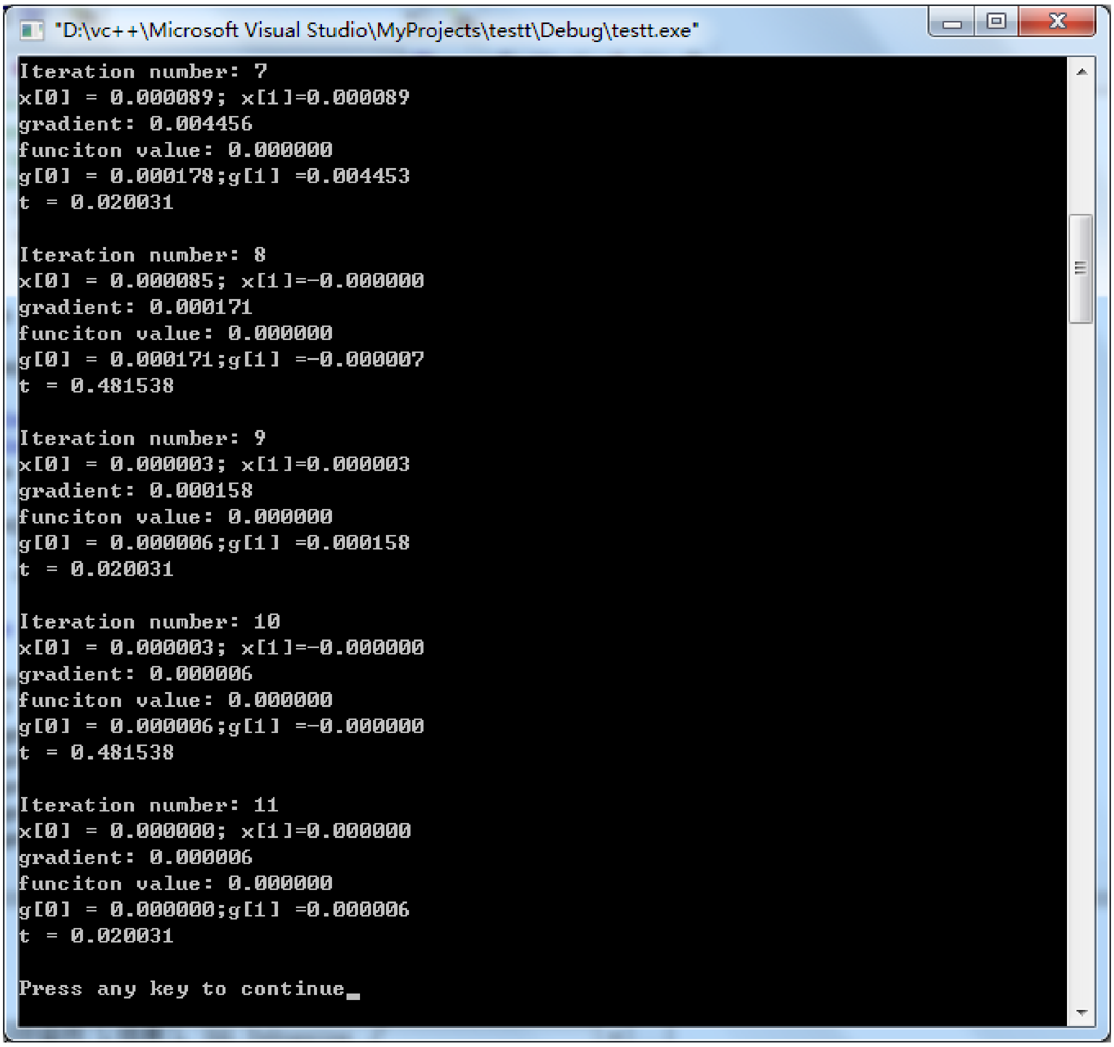
即调用 treeplotter 文件的 createPlot()函数

4.最后调用 Classify()函数，依赖刚刚所得到的决策树来对验证数据集进行分类，将所得分类结果与真实数据结果进行对比，统计分类正确的数值，得出最终准确率。

四、实验结果



实验截图 1



实验截图 2

五、实验结果分析

梯度下降算法找到的不一定是最小值，有可能是极小值，有时候甚至是鞍点。其实梯度下降只是不动点迭代的一种，梯度下降找到的其实是不动点，而不是直接寻找极小值。在可导的区间上，梯度下降迭代的不动点（梯度为 0 的点）有三类——极大值，极小值，鞍点。对于梯度下降来说，极大值是不稳定的（再小得误差都可能导致迭代从不动点上逃逸，并且，除非你初始值就是极大值，否则迭代过程几乎不可能到达极大值），而鞍点不稳定性次之（在某侧的误差会导致逃逸），而极小值是梯度下降过程最稳定的不动点。迭代过程可以参照下雨的时候水的流向，水总是会聚集在坑（极小值）里面。并不是所有不动点迭代都是收敛的。对于梯度下降来说，梯度下降只是在点得足够小的邻域内，负梯度方向让函数值减小，如果参数不合适，迭代过程总是超过了这个足

够小的邻域，那迭代可能会发散。如果函数是凸的，那么梯度下降会收敛到最小值（因为只有一个极小值，它就是最小值）。对于一般问题来说，梯度下降能找到最小值是个概率事件。虽然有很多优化方法，但它仍然是个概率事件。有很多概率方法，试图让你从不稳定的不动点附近“跳出去”（比如，对迭代的过程增加一些扰动），这样得到的不动点往往更加稳定。通常，这些稳定的不动点即便不是最优值，性质也足够好了。所以，在很多时候我们也并不是必须要找到最优值。大部分迭代算法其实都是不动点迭代。构造这个过程的精髓在于——解就是不动点，但不动点未必是解。对于某些特定的问题，不动点就是解。