实验六 线性回归

线性回归是利用数理统计中回归分析，来确定两种或两种以上变量间相互依赖的定量关系的一种统计分析方法。线性回归中的数据呈现线性关系，其表达形式为y = w（T）x+b。其中w为系数向量组，x为特征值向量组，b为常值系数。  
  
     我们通常将数据集分为训练集和测试集，使用训练集来确定待定系数，测试集来测试误差，通过代价函数进行优化。  
  
     训练模型的一些方法：正规方程法，梯度下降法等。  
     测试模型的一些方法：留出法，交叉验证法，自助法等。  
     此次实现主要使用的是梯度下降法和留一法。

试验代码：

#include <iostream>

#include<fstream>

#include<cmath>

const int sample = 28;//样本数 fish样本为44 house\_price样本为28

const int feature = 13;//特征值数 fish样本为4 house\_price样本为13

const int scaling = 10;//特征缩放大小 fish样本为100 house\_price样本为10

double a =0.001;//学习率 fish样本为0.0002 house\_price样本为0.001

#define path "D:\house\_price.txt"

using namespace std;

int main()

{

double x[sample][feature] = { 0 };

ifstream infile;

infile.open(path);

int i, j;

if (!infile)

{

cout << "cannot open file" << endl;

return -1;

}

for ( i = 0; i < sample ; i++)

{

for ( j = 0; j < feature; j++)

{

infile >> x[i][j];

if (j == feature-1)

x[i][j] = x[i][j] /scaling;

if (j == 0)

x[i][j] = 1;

cout << x[i][j] << " ";

}

cout << endl;

}

infile.close();

double J=0;//代价函数

double Y[sample] = { 0 };//预测函数

double temp[feature - 1] = { 0 }, t = 1, p[feature] = { 1 };//置换变量

double w[feature-1] = {0};//系数数组

double hype;//验证值

double A=0;//验证值与真实值的平均差异

int cnt = 0;//计数变量

double sum = 0, s[feature - 1] = { 0 };//求和变量

cout << endl;

cout <<"输出验证结果："<< endl;

for (int m = 0; m < sample; m++)

{

for (int n = 0; n < feature; n++)

{

p[n] = x[m][n];

x[m][n] = 0;

//抽取一个为测试集

while (fabs(J - t) > 0.001)//当代价函数变化极小时退出循环

{

t = J;//等于上一个代价函数的值

J = 0;//代价函数归零

for (i = 0; i < sample; i++)//求每个样本的预测函数

{

sum = 0;

for (j = 1; j < feature - 1; j++)//除去序值

{

sum += w[j] \* x[i][j];

}

Y[i] = sum;

}

for (j = 0; j < feature - 1; j++)//梯度下降法

{

sum = 0;

for (i = 0; i < sample; i++)

{

sum += (1.0 / (sample))\*(Y[i] - x[i][feature - 1])\*x[i][j];

cnt++;

}

temp[j] = w[j] - a \* sum;

}

for (int i = 0; i < feature; i++)

{

w[i] = temp[i];

}

for (i = 0; i < sample; i++)//求代价函数

{

for (j = 1; j < feature - 1; j++)

{

J += (1.0 / 2.0 / (double)(sample))\*pow((Y[i] - x[i][feature - 1]),2);

}

}

}

}

sum = 0;

for (i = 0; i < feature-1; i++)//开始使用留一法进行测试

{

x[m][i] = p[i];

sum += w[i] \* x[m][i];

}

hype = sum;

cout << "第" << m+1<<"个测试结果：" << endl;

cout << "验证值为" << hype << endl;

cout << "真实值为" << p[3] << endl;

cout << "验证值与真实值的差为" << fabs(p[3]-hype) << endl<<endl;

A += fabs(p[3] - hype);

//求系数均值

for (i = 0; i < feature - 1; i++)//求平均系数值

s[i]+= w[i];

}

cout << endl;

cout << "线性回归系数值：" << endl;

for (i = 0; i < feature - 1; i++)

{

w[i] =s[i] / sample;

cout << "第" << i + 1 << "个系数为" << w[i] << endl;

}

cout <<endl<< "代价函数的值为"<< J << endl;

cout <<endl<<"梯度下降法一共运行了"<< cnt<<"次"<<endl << endl;

cout << "验证值与真实值的平均差为" << A/(sample)<< endl<<endl;

cout << "该线性数据集学习率为" << a << endl;

cout << "该线性数据集样本数为" << sample << endl;

cout << "该线性数据集特征数为" << feature << endl;

cout<< "更换数据集请记得调试特征缩放以及以上参数!" << endl;

cout << endl;

system("pause");

return 0;

}

运行结果：











