# 初识机器学习

## 监督学习

监督学习（Supervised Learning）,输入样本已被标记。有两类问题，回归(Regression)和分类（Classification）。

Regression的输出是连续的，Classification输出离散。

**典型案例：**

Regression根据已知楼盘房价推测新楼盘房价。

Classification根据肿瘤样本的良性/恶性，推测新肿瘤是良性还是恶性。

## 无监督学习

无监督学习（Unsupervised Learning），输入样本没有被标记，需要根据样本间的相似性对样本集分类。

**典型案例：**

用户细分

混合音频的剥离

# 线性回归

分为单变量线性回归（Linear regression with one variable）和多变量线性回归（(Linear Regression with Multiple Variables）。

一般公式为：

其中，，为变量数量，也叫特征数量。

矩阵公式：

## 代价函数

1. **函数原型**

文本

描述已自动生成

1. **python实现**

|  |
| --- |
| #代价函数  #x, y, theta为矩阵变量  def costFunctionJ(x, y, theta) :  #计算(x\*theta - y) ^ 2  inner = np.power(((x.dot(theta.T)) - y), 2)  #计算  累加和/2m  return np.sum(inner) / (2 \* len(x)) |

1. **原理**

假定训练实例原本为97x1的矩阵，为了计算矩阵相乘，前面插入一列1，为97x2矩阵。

矩阵为每个实例的输出，也是97x1矩阵。

m为实例数量，97。

矩阵为2x1矩阵，即，其中 = 1，每次运算有两个值。

回到代价函数公式，其中：

，

换成矩阵形式，即矩阵和矩阵相乘，，所得结果是97x1矩阵，包含了m个训练实例的每一个计算结果。

矩阵运算包含了每一个训练实例的计算结果与实际结构的差值。

python代码np.power((x.dot(theta.T), 2)，对差值矩阵的每个元素求平方。numpy使用dot进行矩阵乘法，使用multiply进行矩阵点乘。不要使用\*，容易乱。

最后的np.sum(inner) / (2 \* len(x))，也就是公式中功能。

## 批量梯度下降

1. **函数原型**

文本

描述已自动生成

1. **python实现**

|  |
| --- |
| def gradientDescent(x, y, theta, alpha, iters):      #theta矩阵清零      temp = np.matrix(np.zeros(theta.shape))      #revel()将多维数组降为一维      #返回theta元素个数      parameters = int(theta.ravel().shape[1])      #比如iters=5,则cost为长度为5的数组，值为0      cost = np.zeros(iters)        for i in range(iters):          error = (x.dot(theta.T)) - y          print(error)            for j in range(parameters):              term = np.multiply(error, x[:,j])              temp[0,j] = theta[0,j] - ((alpha / len(x)) \* np.sum(term))            theta = temp          cost[i] = costFunctionJ(x, y, theta)        return theta, cost |

1. **原理**

对于，梯度下降公式：

其中：

公式展开：

其中：

n为的特征数，本例中n=2。

求导后：

其中，代表矩阵中第行、第列，也就是第个训练实例的第个特征。

对于：

设矩阵R为计算结果，实际上是矩阵R和矩阵X的第一列相乘矩阵的累加。

回到python：

代码error = (x.dot(theta.T)) – y即为。

代码term = np.multiply(error, x[:,j])，即为，其中为矩阵的第列。

代码np.sum(term)即为。

代码temp[0,j] = theta[0,j] - ((alpha / len(x)) \* np.sum(term))即为对的迭代过程，其中在这里的取值为0，1。

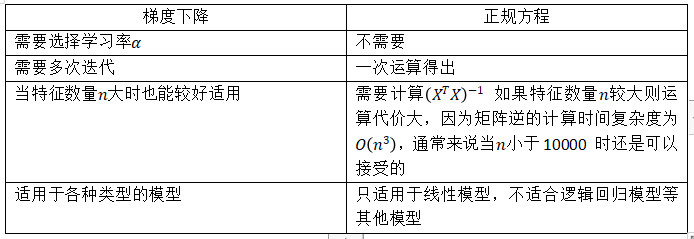
## 正规方程

正规方程（Normal Equation）是通过求解下面的方程来找出使得代价函数最小的参数的： 。

解得

TODO:推导过程设计矩阵求导，以后再推吧。

梯度下降与正规方程的比较：



总结，只要特征变量的数目并不大，标准方程是一个很好的计算参数的替代方法。具体地说，只要特征变量数量小于一万，我通常使用标准方程法，而不使用梯度下降法。