HUAWEI

5.1 机器学习中的回归问题（上）

目录

Contents

机器学习当中的回归问题是什么

一个实例：当你准备开一家餐厅时，选址是一件非常重要的事情，你需要考虑这家餐厅开在什么地方销量会最好，这个时候你可能需要预测一下不同地点开餐厅的收入是多少，从而可以得到一个最佳的开店地址。这里我们就可以构建一个回归模型，来预测不同地点餐厅的销售额。

数据标签：回归问题属于“监督学习”的范畴，即模型建立的必要条件是训练数据当中必须包含标签（label）数据，比如：

样本编号	特征1	特征2	特征3	特征4	特征5	label
1	…	…	…	…	…	56
2	…	…	…	…	…	43

1、数据标签用于在进行回归模型训练时对于label列分布的规律进行监督式的发现，准确的数据label是构建一个准确的回归模型的非常重要的因素。

2、label是一个连续型的数值。

回归分析的过程

回归问题通常是用来预测一个值（连续性的数值），比如预测房价、商品的销量和设备的寿命等等，例如一个产品的日销量是300单，我们通过回归分析的预测值为298单，我们认为这是一个比较好的回归预测。通常，这类预测问题我们称之为回归问题，回归问题我们可以用回归模型（regression）进行解决，回归模型定义了输入与输出的关系，输入即现有知识，而输出则为预测值。

一个预测问题在回归模型下的解决步骤为：

数据收集： 我们将收集到的历史数据称之为训练集，“以史为鉴”，模型的训练都是以历史数据为基础进行分析的。

学习训练：有了历史数据之后，我们就可以开始学习训练并总结经验，以上面的产品销量预测为例，我们需要从历史数据中学习，在满足什么样输入的情况下销量是300，以此类推，我们会总结出一个对应关系。在学习阶段，应当找到最合适的学习方法。这类方法我们称之为学习算法 。

回归预测：学习完成后，当接受了新的数据（输入）后，我们就能通过学习阶段获得的对应关系来预测输出。

学习训练的过程往往不是一帆风顺的，对于训练的结果，我们要用有效的手段进行评估；当评估结果不理想时，我们需要改进学习方法来达到更好的结果。

回归实例&典型算法

典型回归方法

- 线性回归
- KNN回归
- 分类回归树（基于平方误差）
- ……

回归问题的评价

- 平均绝对误差（ mean absolute error，MAE ）

$$\frac{1}{m}\sum_{i=1}^{m}\left|(y_i - \hat{y}_i)\right|$$

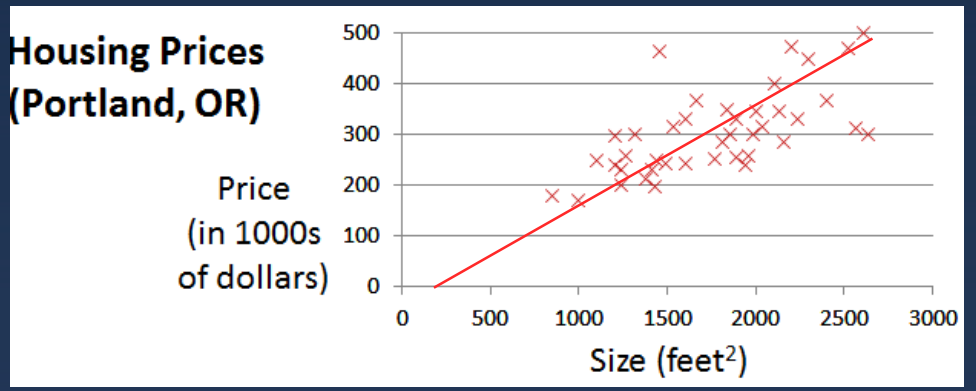
- 均方根误差（root mean squared error，RMSE）

$$\sqrt{\frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2}$$

- 均方误差（Mean Squared Error，MSE）

$$\frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2$$

Size in feet² (x)	Price ($) in 1000's (y)
2104	460
1416	232
1534	315
852	178
…	…

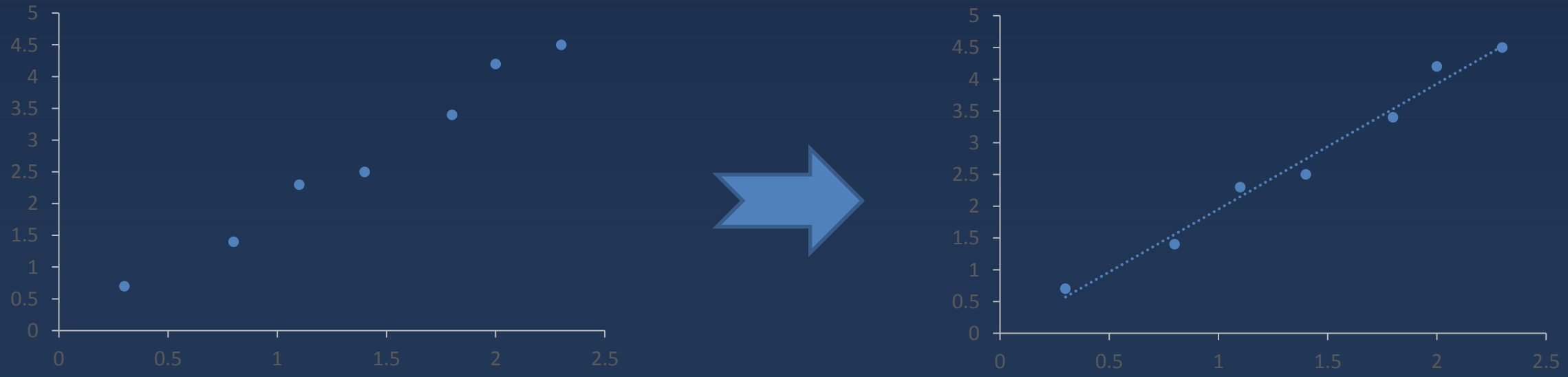


目录

Contents

常用的回归算法：线性回归

在统计学中，线性回归（Linear regression）是利用称为线性回归方程的最小二乘函数对一个或多个自变量（输入数据）和因变量（输出数据）之间关系进行建模的一种回归分析。这种函数是一个或多个称为回归系数的模型参数的线性组合。如下图所示，随着输入数据（X）的增大，输出数据（Y）也有明显的增大趋势，因此我们可以用一条线进行拟合这种变化趋势（输出数据随输入数据的增大而增大），从而可以根据新的输入数据（X）来预测对应的输出数据（Y）。回归算法中最简单和最常用的就是线性回归算法。



常用的回归算法：线性回归

回归分析中，只包括一个自变量和一个因变量，且二者的关系可用一条直线近似表示，这种回归分析称为一元线性回归分析。如果回归分析中包括两个或两个以上的自变量，且因变量和自变量之间是线性关系，则称为多元线性回归分析。

一元线性回归算法分析：

一元线性回归分析法的预测模型为：$Y_t = \emptyset_0 + \emptyset_1 x_1$

多元线性回归算法分析：

多元线性回归分析法的预测模型为：$Y_t = \emptyset_0 + \emptyset_1 x_1 + \emptyset_2 x_2 + \emptyset_3 x_3 + \cdots + \emptyset_n x_n$

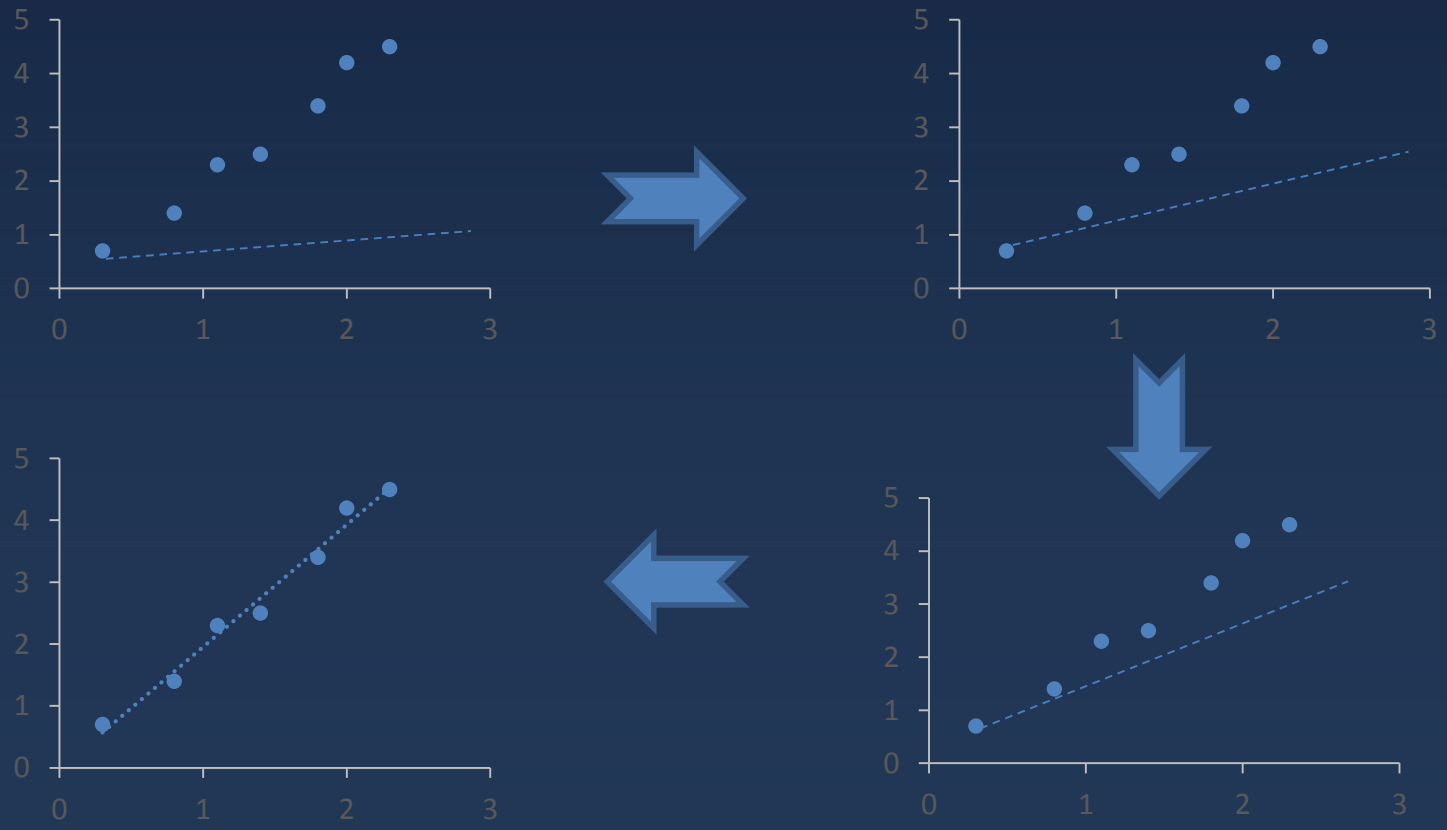
式中$\emptyset$代表线性回归方程的参数，计算公式为：$\emptyset_j = \emptyset_j - a\frac{1}{m}\sum_{i=0}^{n}\left(h_\emptyset\left(x^i\right) - y^i\right) * x_j^i$ for j=0...n

注：$h_\emptyset = \emptyset_0 + \emptyset_1 x_1 + \emptyset_2 x_2 + \emptyset_3 x_3 + \cdots + \emptyset_n x_n$，$a$*为学习率*

线性回归算法中使用系数权重来加权每个特征变量的重要性，使用梯度下降来确定这些权重和偏置。

梯度下降

梯度下降法（Gradient descent）是一个一阶最优化算法，通常也称为最速下降法。 要使用梯度下降法找到一个函数的局部最小值，必须向函数上当前点对应梯度（或者是近似梯度）的反方向的规定步长距离点进行迭代搜索。梯度下降方法基于以下的观察：如果实值函数 F（x）在点a处可微且有定义，那么函数 F（x）在a 点沿着梯度相反的方向-$\nabla F(a)$ 下降最快。过程如下图：



线性回归的几个关键点

1.适用于预测目标与特征之间线性关系强的数据集。

2.有直观的理解和解释。

3.线性回归对异常值非常敏感。

4.避免函数梯度下降到局部最优解，需要进行标准化处理。

5.梯度下降学习率a的选择不合适会出现“之字型”下降。

6.梯度下降在靠近极小值时速度减慢。

目录

Contents

什么是K最近邻回归（KNN）

如图所示，我们有两类不同的样本数据，分别是绿色的三角形和红色的正方形。图中的黄色圆形表示我们新输入带识别的数据，现在我们需要做的是把黄色圆形归类到已有的类别中。我们常说，物以类聚，识别一个事物的属性时，我们可以把该事物归类到已知属性并且和该事物最接近的事物类别中去，所以我们需要做的就是找到最接近圆形的类别是哪一类。

我们假设实线圆的半径为4（K=4）：黄色圆点最接近的3个点是2个绿色三角形和1个红色正方形，少数服从多数，我们可以认为黄色圆点是属于绿色三角形类别。

我们假设虚线圆的半径为8（K=8）：黄色圆点最接近的5个点是2个绿色三角形和3个红色正方形，少数服从多数，我们可以认为黄色圆点是属于红色正方形类别。

于此我们看到，当无法判定当前待识别数据是从属于已知类别中的哪一类别时，我们可以依据统计学的理论看它所处的位置特征，衡量它周围邻居的权重，而把它归类(或分配)到权重更大的那一类。这就是K近邻算法的核心思想。

KNN算法的核心更加像是分类问题，它不仅可以用于解决分类问题，也可以解决回归问题，KNN用在回归问题上时是采用分类思想来解决回归问题。将待测样本的值回归为从属类别的均值

K最近邻回归（KNN）距离计算

KNN计算距离的方式有很多种，最常用的就是欧式距离。假设我们有两组数据x（$x_1,\dots,x_n$）和y（$y_1,\dots,y_n$），每个数据有n个属性。那么x和y的欧式距离计算如下：

欧式距离计算公式：$d(x,y)=\sqrt{(x_1-y_1)^2+(x_2-y_2)^2+\cdots+(x_n-y_n)^2}=\sqrt{\sum_{i=1}^{n}(x_i-y_i)^2}$

在二维平面上两点a（x_1，y_1）和b（x_2，y_2）的欧式距离：

$$d_{12}=\sqrt{(x_1-x_2)^2+(y_1-y_2)^2}$$

在三维空间里两点a（x_1，y_1，z_1）和b（x_2，y_2，z_2）的欧式距离：

$$d_{12}=\sqrt{(x_1-x_2)^2+(y_1-y_2)^2+(z_1-z_2)^2}$$

两个n维向量a（x_{11}，x_{12}，…，x_{1n}）和b（x_{21}，x_{22}，…，x_{2n}）的欧式距离：

$$d_{12}=\sqrt{\sum_{k=1}^{n}(x_{1k}-x_{2k})^2}$$

K最近邻回归（KNN）的优缺点

KNN的主要优点有：

1.理论成熟，思想简单，既可以用来做分类也可以用来做回归。

2.可用于非线性分类。

3.训练时间复杂度比支持向量机之类的算法低，仅为O(n)。

4.和朴素贝叶斯之类的算法比，对数据没有假设，准确度高，对异常点不敏感。

5.由于KNN方法主要靠周围有限的邻近的样本，而不是靠判别类域的方法来确定所属类别的，因此对于类域的交叉或重叠较多的待分样本集来说，KNN方法较其他方法更为适合。

6.该算法比较适用于样本容量比较大的类域的自动分类，而那些样本容量较小的类域采用这种算法比较容易产生误分。

KNN的主要缺点有：

1.计算量大，尤其是特征数非常多的时候。

2.样本不平衡的时候，对稀有类别的预测准确率低。

3.使用懒散学习方法，基本上不学习，导致预测时速度比起逻辑回归之类的算法慢。

4.相比决策树模型，KNN模型可解释性不强。

Thank You.

华为云机器学习服务MLS
www.huaweicloud.com/product/mls.html