基于线性回归模型的研究

熊舒鹏

（昆明理工大学 机器人工程 昆明市）

# 线性回归 (Simple Linear Regression)

## 1.1 线性回归定义

**(Gemini-2.0-Flash-Thinking(experimental))**

假设我们有一个包含个数据点的训练集，其中是输入特征，是对应的输出标签。一元线性回归的目标是找到一条直线，能够最好地拟合这些数据点。这条直线的方程可以表示为：



其中，是截距(bias)偏置，是斜率(weight)权重。

我们的目标是找到合适的和使得预测值 尽可能接近真实值。

**(Grok3)**

线性回归假设目标变量与特征变量之间存在线性关系。对于简单线性回归（只有一个特征），模型可以表示为：



其中，是截距(当时的预测值)。

斜率（特征对的影响）。

：误差项，表示模型无法解释的随机噪声。

对于多元线性回归（多个特征），假设有个特征，模型扩展为：



**(Deepseek-R1)**

线性回归假设目标变量是输入特征的线性组合，模型形式为：



向量化表示为：



其中，，（含截距项）。

## 1.2 损失函数、代价函数 (Cost Function)

**(Gemini-2.0-Flash-Thinking(experimental))**

我们使用均方误差 (Mean Squared Error, MSE) 作为衡量模型预测性能的代价函数：



将线性回归模型的方程代入代价函数：



我们的目标是找到使最小的和。

**梯度下降 (Gradient Descent)**

梯度下降是一种迭代优化算法，通过不断更新参数来最小化代价函数。更新规则如下：



其中，是学习率，控制着每次迭代参数更新的步长。我们需要计算代价函数对和的偏导数。

计算过程（偏导数）：







**正规方程 (Normal Equation)**

正规方程提供了一种直接计算出最优参数和的解析方法，不需要迭代。对于一元线性回归，可以直接通过对代价函数求偏导并令其等于零来求解。

将代价函数对和的偏导数设置为零：



因此，斜率的最优解为：



截距的最优解为：



**多元线性回归 (Multiple Linear Regression)**

假设我们有一个包含*n*个数据点的训练集，其中每个输入特征是一个*m*维向量，是对应的输出标签。多元线性回归模型的方程可以表示为：



为了方便使用矩阵表示，我们通常会给每个输入特征向量添加一个额外的维度，其值为1，对应于截距项。因此，我们将输入特征向量表示为 ，参数向量表示为。

这样，模型方程可以简洁地表示为：



多元线性回归的代价函数与一元线性回归类似，仍然使用均方误差：



使用矩阵表示，我们可以将所有训练数据表示为一个*n×(m+1)*的矩阵*X*，其中每一行对应一个训练样本的特征向量（添加了截距项1）：



将所有输出标签表示为一个*n×1*的向量*y*：



那么，所有预测值可以表示为，代价函数可以使用矩阵形式表示为：



利用梯度下降得到：



**(Grok3)**

为了方便推导，我们使用矩阵形式表示。假设有个样本，每个样本有个特征，则：

特征矩阵*X*（大小为）：



其中，第一列全为1，用于表示截距项。

目标向量（大小为）：



参数向量（大小为）：



误差向量（大小为）。

因此，模型的矩阵形式为：



**参数估计：最小二乘法**

线性回归的目标是找到参数，使模型的预测值与真实值之间的误差最小。常用的方法是最小二乘法，即最小化残差平方和（Residual Sum of Squares, RSS）：



用矩阵形式表示：



为了求解，我们需要最小化*RSS*。具体步骤如下：

1. 展开*RSS*





注意到是一个标量，且（标量的转置等于其本身），因此：



1. 对求导

为了找到*RSS*的极值点，对求偏导数：



其中，是常数，对的导数为0。

是导数为（因为是常量）。

是二次型，其导数（矩阵微分规则）

因此：



1. 令导数为0

为了找到极值点，令导数为0：





1. 解出

假设是可逆的（即满秩），两边左乘：



**(Deepseek-R1)**

损失函数定义为所有训练样本预测值与真实值的平方误差均值：



矩阵形式为：



**梯度下降：**

梯度下降通过迭代更新参数以最小化损失函数。对每个参数求偏导数：



向量化梯度表达式：



参数更新规则（学习率为）：



**正规方程：**

令梯度为零求得闭式解：

