
“十四五”高等职业教育新形态一体化教材
人工智能技术应用
机器学习技术与应用
杜 辉 葛 鹏 赵瑞丰◎主编
中国铁道出版社有限公司
CHINA RAILWAY PUBLISHING HOUSE CO., LTD.

项目 1 运用线性回归算法实现趋势预测

1.1 项目导入

图 1-1 所示为截至 2022 年 8 月杭州近一年房价走势图，整体上来看数据走势呈现一个线性增长的趋势，那么机器学习线性回归要做的就是从已经存在的历史数据中总结规律，从而能对未来数据作出预测。如可以预测杭州 2022 年 9 月份房价大概是什么水平；预测牛肉价格会是什么走势。而这些问题，都可以通过线性回归解决，线性回归也是机器学习中解决回归问题的主要算法之一。在这个项目中，我们将会和小艾一起，解决上面提到的一些线性回归问题。

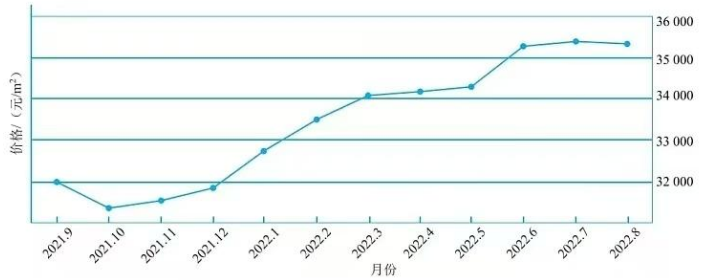

图 1-1　截至 2022 年 8 月杭州近一年房价走势

1.2 项目目标

（1）熟悉样本、特征、观测值、预测值、损失等概念。

（2）理解线性回归大致原理。

（3）能够应用线性回归模型解决实际问题。

1.3 知识导入

1.3.1 线性回归概念

线性回归（Linear Regression）是机器学习中比较基础的算法之一。简单的线性回归就是要得到一条直线去拟合已知的数据分布。

视频

线性回归算法应用的介绍

具体地，给定数据集 $D=\{(x_1,y_1),(x_2,y_2),\cdots,(x_m,y_m)\}$，其中，$x_i$ 为自变量，$y_i\in\mathbf{R}$ 为因变量，共有 m 个样本。每个样本包含 n 个特征，线性回归要做的就是要学得一个模型，在输入一个未知自变量时，尽可能准确地预测因变量的值。

如小艾因为工作原因需要在杭州市内外出，打车较多。时间久了，他把每次打车里程和打车费用记录下来，绘制成图 1-2。

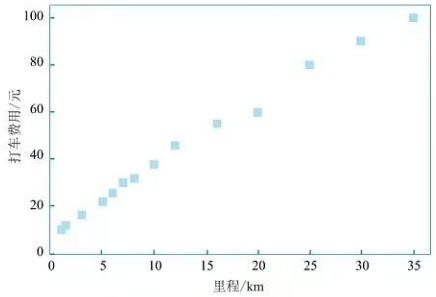

图 1-2　打车费用记录散点图

图 1-2 中的点可以看作一个数据集，$D=\{(x_1,y_1),(x_2,y_2),\cdots,(x_m,y_m)\}$，把每个数据点称为一个样本，其中，$x_i$ 只含有一个特征，表示里程；i 表示第 i 个训练样本，共 m 个样本；y_i 是数值，表示打车费。观察图 1-2 很容易发现，里程 x_i 和打车费 y_i 之间存在着一定的线性关系。也就是说可以大致画出表示 x_i 与 y_i 之间关系的一条直线。

如图 1-3 所示，在该直线中，里程 x_i 为自变量，打车费 y_i 为因变量。而线性回归的目的是利用自变量 x_i 与因变量 y_i 学习出一条能够描述两者之间关系的线。

图 1-3 的直线拟合的是里程和打车费之间的关系，属于一元线性回归，那么多元线性回归则会拟合出一个平面或者超平面。

线性回归非常简单，形式简单、思想简单，却蕴含着机器学习中的一些重要思想，许多功能强大的非线性模型是基于线性模型，并引入层级结构或者高维映射得到的。

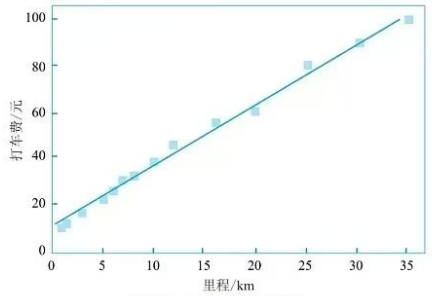

图 1-3　打车费用回归线

1.3.2　线性回归模型

将线性回归的问题进行抽象化，转换成能求解的数学问题。在上面的例子中，可以看出自变量 x_i 与因变量 $y_i(i=1,2,\cdots,m)$ 大致呈线性关系，因此可以对因变量作出假设(hypothesis)，公式如下：

$$\hat{y}_i = \theta_1 x_i + \theta_0$$

式中的 $\hat{y}_i$ 代表什么意思呢？

回想一下里程和打车费的直线，这条直线并没有完全拟合所有数据点，对每一个输入 x，对应直线上的 y' 与真实观测值 y 存在一定的误差，预测值并不完全等于观测值，$\hat{y}_i$ 则是假设函数的预测值。该假设是一元线性函数模型，其中含有两个参数 θ_1 和 θ_0。其中 θ_1 可看作斜率，θ_0 则是直线在 y 轴上的截距。

那么，二元、多元线性回归公式是什么？

回归模型中一元、二元分别代表什么含义？一元代表自变量只有一个特征属性，即上式中的 x 是一维的；二元则表示自变量有两个特征属性；依此类推，多元表示输入变量 x 有 n 个特征属性。

$$\hat{y}_i = \theta_n x_n^i + \theta_{n-1} x_{n-1}^i + \cdots + \theta_2 x_2^i + \theta_1 x_1^i + \theta_0$$

式中，$i=1,2,\cdots,m$，表示样本数，n 表示特征数。变量系数就是模型要根据已知数据学习的参数。

那么接下来的主要工作就是如何学习这些参数，也就是训练过程。在开始学习线性回归模型的求解过程之前，先完成图 1-4 所示的小测验。

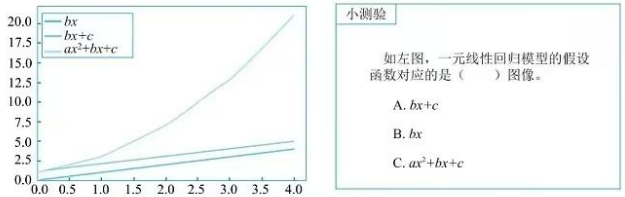

图 1-4　线性回归小测验

1.3.3　求解线性回归

假设要根据前文的里程和打车费用数据训练出一个一元线性回归模型，模型的参数要在学习数据的过程中不断更新，那么更新到值为多少才算是最优模型或者预测最准确的模型呢？这就需要定义损失，也就是当前模型预测值和真实值的差距。若损失很小，表明模型与数据真实分布很接近，则模型性能良好；若损失很大，表明模型与数据真实分布差别较大，则模型性能不佳。训练模型的主要任务是寻找损失函数最小化对应的模型参数。损失函数是机器学习算法的核心。

如线性回归是对因变量 y 和几个独立变量 x_i 之间的线性关系进行建模。因此，在空间中对这些数据拟合出一条直线或者超平面。

回归模型 $y = a_0 + a_1x_1 + a_2x_2 + \cdots + a_nx_n$，使用给定的数据点找到使模型预测最准确的最佳系数 $a_0, a_1, \cdots, a_n$。

那找到最佳参数和损失函数有什么关系？

对已知数据点 $(x_{1i}, x_{2i}, \cdots, y_i)(i = 1, 2, \cdots, n)$，使用假设函数能够计算模型预测值 $\hat{y}$，但是，真实值 y 和预测值 $\hat{y}$ 相同吗？

显然，如图 1-5 所示，预测值和真实值之间存在不同程度的误差。

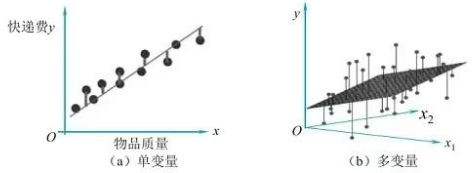

图 1-5　单变量与多变量回归预测

训练过程就是要更新函数参数值，使得预测值和真实值的差距减到最小。

下面先看几个函数的基本定义：

（1）损失函数（Loss Function）：定义在单个样本上，计算的是一个样本的误差。

（2）代价函数（Cost Function）：定义在整个训练集上，是所有样本误差的平均，也就是损失函数的平均。

（3）目标函数（Object Function）：最终需要优化的函数。即经验风险 + 结构风险（代价函数 + 正则化项），在有些实际问题中，为了防止过拟合或者其他原因会在一般的代价函数的基础上增加一个正则项。

在机器学习中，损失函数一般分为分类和回归两类，回归会预测给出一个数值结果，而分类则会给出一个标签。本项目着重介绍回归问题中常用的几种损失函数。

1. 平均绝对误差

平均绝对误差又称 L1 范数损失，它是目标值与预测值之差绝对值的和，表示了预测值的平均误差幅度，而不需要考虑误差的方向。

其公式就是所有样本预测值和真实值之间差值的绝对值之和再取平均。

$$\mathrm{MAE} = \frac{\sum_{i=1}^{n} |y_i - \hat{y}_i|}{n}$$

MAE 虽能较好衡量回归模型的好坏，但是绝对值的存在导致函数不光滑，在某些点上不能求导。考虑将绝对值改为残差的平方，就是下面的均方误差。

2. 均方误差

回归损失函数中最常用的误差，它是预测值与目标值之间差值的平方和，其公式就是所有样本预测值和真实值之间差值的平方和再取平均。

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

均方误差具有非常好的几何意义，它对应了常用的欧氏距离。均方误差又称 L2 范数损失。

3. 均方根误差

由于 MSE 与目标变量的量纲不一致，为了保证量纲一致性，需要对 MSE 进行开方。公式对 MSE 的结果进行开平方根。

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

模型训练的目标就是求解使得损失函数降到最小值的模型参数$(a_0, a_1, \cdots, a_n)$。损失函数的可视化如图 1-6 所示。

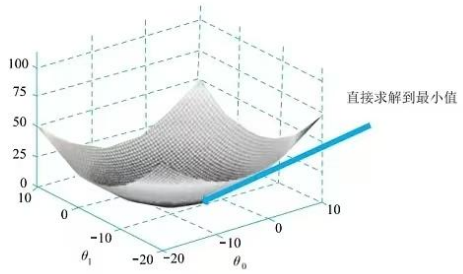

图 1-6　损失函数

那么，应该使用什么方法找到最小值呢？下面介绍最小二乘法。

基于均方误差最小化进行模型求解的方法称为“最小二乘法”。

在线性回归中，最小二乘法就是试图找到一条直线，使所有样本到直线的欧氏距离之和最小。

求解 w 和 b 使 $E(w,b) = \sum_{i=1}^{n} (y_i - (wx_i + b))^2$ 最小化的过程称为线性回归的最小二乘参数估计，将 $E(w,b)$ 分别对 w 和 b 求偏导并令其为 0，即可推出系数的解。在这里不阐述公式推

导的细节。

1.3.4　过拟合与欠拟合

视频

过拟合与欠拟合

机器学习中要评估一个模型好坏，模型的泛化能力也是要考查的标准，泛化能力强的模型才是好模型。

对于训练好的模型，若在训练集中表现差，在测试集中表现同样会很差，这可能是欠拟合导致的；若模型在训练集中表现非常好，却在测试集中差强人意，则这便是过拟合导致的。

如图 1-7 所示描述了同一训练集得到的不同模型的拟合结果。

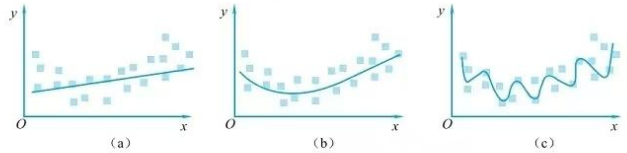

图 1-7　欠拟合与过拟合对比

在 1-7(a)中，用比较简单的函数或者说选择少量特征去拟合训练数据，能够看出，这个线性函数拟合效果并不好，有些数据的误差很大。

于是，在图 1-7(b)中，选择更多的特征也是一个复杂一些的函数模型去拟合数据，能够看出，拟合效果明显好了很多，均方误差变小了。

那么是不是模型越复杂，特征选取的越多，效果越好呢？

在图 1-7(c)中，函数模型可能是个高阶多项式，对训练集已经拟合得很好，但是，模型效果好不好，不能只看训练集上的损失，更重要的是看测试集上的损失。这种情况，模型并不能做更准确的预测，所以并不是个很好的函数模型。

1. 过拟合

过拟合就是模型过于拟合训练集的特征，而模型的泛化能力降低的表现。在对模型进行训练时，有可能遇到训练数据不够，即训练数据无法对整个数据的分布进行估计时，或者在对模型进行过度训练(Overtraining)时，常常会导致模型的过拟合(Overfitting)。随着模型训练的进行，模型在训练数据集上的训练误差会逐渐减小，但是在达到一定程度时，模型在验证集上的误差反而增大。此时便发生了过拟合。

为了防止过拟合，可以采用以下方法来解决。

1) Early stopping

Early stopping 是一种迭代次数截断的方法来防止过拟合，即在模型对训练数据集迭代收敛之前停止迭代来防止过拟合。

在每一个训练 Epoch(一个 Epoch 即为对所有训练数据的一轮遍历)结束时计算开发集的准确率，当准确率不再提高时，就停止训练。

如何判断准确率不再提高呢?

在训练过程中,记录到目前为止最好的开发集准确率,当连续 N(设定值)次 Epoch 没达到最佳准确率时,则可以认为准确率不再提高了。

2)数据集扩增

数据集扩增即需要得到更多的符合要求的数据,即和已有的数据是独立同分布或近似独立同分布的。一般有以下方法:

(1)从数据源头采集更多数据。

(2)复制原有数据并加上随机噪声。

(3)重采样。

(4)根据当前数据集估计数据分布参数,使用该分布产生更多数据等。

3)L1 正则化

L1 正则化基于 L1 范数(L1 范数是指向量中各个元素绝对值之和),即在代价函数后面加上参数的 L1 范数和项,即

$$C = C_0 + \alpha \sum_{i=1}^{k} |w|,$$

式中,C_0 是原始代价函数;α 是正则项系数;w 是模型参数。L1 正则项是为了使得那些原先处于零(即 $|w| \approx 0$)附近的参数 w 往零移动,使得部分参数为零,降低模型的复杂度,防止过拟合,提高模型的泛化能力。

如线性回归的损失函数加入 L1 正则化:

$$L(w,b) = \text{MSE} + \alpha \sum_{i=1}^{k} |w|$$

式中,MSE 表示均方误差。

4)L2 正则化

L2 正则化基于 L2 范数(L2 范数是指向量各元素的平方和的开方),即在代价函数后面加上参数的 L2 范数和项,即参数的平方和与参数的正则项,即

$$C = C_0 + \alpha \sum_{i=1}^{k} w^2$$

式中,C_0 是原始代价函数;α 是正则项系数;w 是模型参数。

2. 欠拟合

欠拟合是指模型没有能够很好地学习到数据特征,不能很好地拟合数据,表现为预测值与真实值之前存在较大的偏差。产生欠拟合的原因是模型过于简单,特征项不够。欠拟合解决方法。

(1)增加特征项:可以添加额外的属性,也可以组合当前属性,生成高阶属性,进而使学习模型的泛化能力更强。

(2)减少正则化参数:正则化的目的是防止过拟合,但是现在模型出现了欠拟合,则需要减少正则化参数。