

线性回归

Leon

2018 年 12 月 19 日

1 机器学习的概念

1.1 有监督学习

对训练集来说 X 对应着是确定的 $f(x)$ ，然后通过构建模型，进行超参的学习

1.2 无监督学习

大部分无监督学习都是没有确定的 $f(x)$ 的，通过一些规则，让机器自己去判断，比如knn算法，用距离来做聚类。

1.3 泛化能力

在机器学习方法中，泛化能力通俗来讲就是指学习到的模型对未知数据的预测能力。在实际情况中，我们通常通过测试误差来评价学习方法的泛化能力。

1.4 过拟合

1.4.1 概念

先谈谈过拟合，所谓过拟合，指的是模型在训练集上表现的很好，但是在交叉验证集合测试集上表现一般，也就是说模型对未知样本的预测表现一般，泛化（generalization）能力较差。

1.4.2 解决办法

一般的方法有early stopping、数据集扩增（Data augmentation）、正则化（Regularization）、Dropout等。在机器学习算法中，我们常常将原

始数据集分为三部分：training data、validation data, testing data。这个validation data是什么？它其实就是用来避免过拟合的，在训练过程中，我们通常用它来确定一些超参数（比如根据validation data上的accuracy来确定early stopping的epoch大小、根据validation data确定learning rate等等）。那为啥不直接在testing data上做这些呢？因为如果在testing data做这些，那么随着训练的进行，我们的网络实际上就是在一点一点地overfitting我们的testing data，导致最后得到的testing accuracy没有任何参考意义。

Early stopping: Early stopping便是一种迭代次数截断的方法来防止过拟合的方法，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。对模型进行训练的过程即是对模型的参数进行学习更新的过程，这个参数学习的过程往往会用到一些迭代方法，如梯度下降（Gradient descent）学习算法。这样可以有效阻止过拟合的发生，因为过拟合本质上就是对自身特点过度地学习。

正则化: 指的是在目标函数后面添加一个正则化项，一般有L1正则化与L2正则化。L1正则则是基于L1范数，即在目标函数后面加上参数的L1范数和项，即参数绝对值和与参数的积项

$$C = C_0 + \frac{\lambda}{n} \sum_w |w|$$

L2正则则是基于L2范数，即在目标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积项：

$$C = C_0 + \frac{\lambda}{2n} \sum_w w^2$$

1.5 交叉验证(cross-validation)

交叉验证，是重复的使用数据，把得到的样本数据进行切分，组合为不同的训练集和测试集，用训练集来训练模型，用测试集来评估模型预测的好坏。在此基础上可以得到多组不同的训练集和测试集，某次训练集中的某样本在下次可能成为测试集中的样本，即所谓“交叉”。有简单交叉验证、S折交叉验证、留一交叉验证。

1.6 线性回归的原理

1:函数模型(Model):

$$h_w(x^i) = \omega_0 + \omega_1 x_1 + \omega_2 x_2 + \dots + \omega_n x_n = \sum \omega^T x_i = W^T X$$

$$X = \begin{bmatrix} 1 \\ x_1 \\ \dots \\ x_n \end{bmatrix}, W = \begin{bmatrix} \omega_0 \\ \omega_2 \\ \dots \\ \omega_n \end{bmatrix} \quad (1)$$

假设有训练数据

$$D = (X_1, Y_1), (X_2, Y_2), \dots, (X_n, Y_n)$$

那么方便我们写成矩阵的形式

$$X = \begin{bmatrix} 1, x_1^1, x_2^1, \dots, x_n^1 \\ 1, x_1^2, x_2^2, \dots, x_n^2 \\ \dots \\ 1, x_1^n, x_2^n, \dots, x_n^n \end{bmatrix}, XW = h_\omega(x^i)$$

2. 损失代价函数:

$$J(W) = \frac{1}{2M} \sum_{i=0}^M (h_\omega(x^i) - y^i)^2 = \frac{1}{2M} (XW - y)^T (XW - Y)$$

3. 算法(algorithm): 求解使得损失函数最小。

1.7 优化方法

1.7.1 梯度下降法

梯度下降沿损失函数的导数方向下降，下降的步幅自己设置。

1.7.2 牛顿法

二阶下降，比梯度下降法更快，而且是求全局最优解，不是局部最优

1.7.3 拟牛顿法

没看懂，但知道适合非线性

1.8 sklearn参数

Ordinary least squares Linear Regression.

1.8.1 fit_intercept: boolean, optional, default True

whether to calculate the intercept for this model. If set to False, no intercept will be used in calculations (e.g. data is expected to be already centered).

1.8.2 normalize : boolean, optional, default False

This parameter is ignored when “fit_intercept” is set to False. If True, the regressors X will be normalized before regression by subtracting the mean and dividing by the l2-norm. If you wish to standardize, please use :class:‘sklearn.preprocessing.StandardScaler’ before calling “fit” on an estimator with “normalize=False”.

1.8.3 copy_X : boolean, optional, default True

If True, X will be copied; else, it may be overwritten.

1.8.4 n_jobs : int or None, optional (default=None)

The number of jobs to use for the computation. This will only provide speedup for n_targets \geq 1 and sufficient large problems. “None” means 1.