Lasso+Elastic Net+Xgboost 房价预测模型

自动化学院 陈欣星 D201677594

一、 方法概要

本文介绍了采用 Lasso 算法、Elastic Net 算法、Xgboost 算法三者的混合模型进行 Kaggle 平台上的 House Prices: Advanced Regression Techniques 房价预测竞赛。代码主要参考了 https://github.com/kuangmeng/HousePrices，我仅在最后训练模型的环节选取了更优的 alpha，得到的结果比直接采用原作者的代码前进两百多名。此外我优化了一些冗余的代码。目前的 Leaderboard 上我排名为第 55 名，分数为 0.11401。

二、 引言

1. 竞赛介绍

在 House Prices: Advanced Regression Techniques 这个 Kaggle 竞赛中，我们需要根据训练数据集中给出的房子的若干个特征和房价建立房价预测模型，针对训练集给出的房子特征进行房价预测。

2. 认识数据

通过观察数据，我们可以得知训练集中有 1460 条样本（1460 间房子），每条样本有 80 种特征。其中 LotFrontage、Alley、MasVnrType、MasVnrArea、BsmtQual、BsmtCond、BsmtExposure、BsmtFinType1、BsmtFinType2、Electrical、FireplaceQu、GarageType、GarageYrBlt、GarageFinish、GarageQual、GarageCond、PoolQC、Fence、MiscFeature 这几个特征存在数据缺失。此外还存在大量字符型特征，需要进行替换。

三、 方法

1. 数据预处理

首先进行缺失数据填充。

LotArea 与 LotFrontage 存在线性相关，故采用二次拟合利用 LotArea 数据计算缺失的 LotFrontage 数据。

剩余缺失数据的特征采用 0 或者 NoXXX 等字符填充。部分数据与某与缺失项存在依赖关系，比如某样本缺失 MasVnrType 特征下的数据，则说明该房屋缺少这部分结构，与之相关的 MasVnrArea 也需要设置为 0。

字符型数据替换为离散型数值。

例如 XXXQu，XXXCond 等特征下的 Ex，Gd，Ta，Fa，Po 表示品质的字符可替换为数值 5，4，3，2，1。

2. 新建特征

利用 XXXQual 和 XXXCond 创建 XXX_good（poor）_qu 和 XXX_good（poor）_cond 特征。这些特征组合成为 qu_list,代表房屋品质。用新建的 bad_heating 特征的 0，1 二值代表暖气品质好坏。MasVnrType_Any 特征的 1，0 二值代表表层砌体有无（大概指是否是毛坯房）。SaleCondition 特征中有些导致房价下降的因素，用新建特征 SaleCondition_PriceDown 的 1 表示，其他为 0。新建 Neighborhood_Good 特征，用 1 代表可提升房价的房屋周围环境。新建 price_category 特征，训练 svm 模型，将 price_category 分为 0，1，2 三类，代表价格低，中，高。新建 season 特征，用 0，1 值代表房价的淡季旺季。新建 reconstruct，recon_after_buy 特征，用 0，1 值代表是否改造过。新建 build_eq_buy 特征，用 0，1 值代表是否建成前就售出。等等。

3. 特征处理

将偏斜度大于 0.75 的特征做一个 log 转换，使之尽量符合正态分布。去除一些异常值。

4. Lasso 算法

在统计学和机器学习中，Lasso 算法（英语：least absolute shrinkage and selection operator，又译最小绝对值收敛和选择算子、套索算法）是一种同时进行特征选择和正规化的特征选择方法，旨在增强统计模型的预测准确性和可解释性，最初由斯坦福大学统计学教授 Robert Tibshirani 于 1996 年提出[1]。

Lasso 的思想就是在传统的最小二乘估计上对模型的系数施加一个 L_1 惩罚。

模型形式如下

$$\hat{\beta}^{lasso} = \operatorname{argmin}_{\beta} \sum_{i=1}^{N}(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 \quad \text{subject to } \sum_{j=1}^{p} |\beta_j| \le t.$$

上式等价于

$$\hat{\beta}^{lasso} = \operatorname{argmin}_{\beta} \left\{ \sum_{i=1}^{N}(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\}$$

它与岭回归(ridge regression)非常相似，只是将 L_2 惩罚换成了 L_1 惩罚。利用这个惩罚项函数有一些局限性，例如，当数据类型是一个高纬数据，即特征显著多于样本时，Lasso 通常在饱和之前选择 N 个变量。同样的，如果存在一组高度相关的变量时，Lasso 倾向于选择其中的一个变量，而忽视其他所有的变量。

在选取参数 alpha 时我采用了 LassoCV 函数多次尝试，寻找到了最优 alpha=0.0002，这一步使得代码运行的结果大大提升。

5. Elastic Net 算法

在 2005 年，Zou 提出了一种在 ridge regression 和 the lasso 之间折衷的方法：Elastic Net[2]。Elastic Net 在惩罚项上增加了一个二次方，当独立使用时类似于岭回归的用法。Elastic Net 的数学实现算法如下

$$\hat{\beta} = \operatorname*{argmin}_{\beta}(\|y - X\beta\|^2 + \lambda_2\|\beta\|^2 + \lambda_1\|\beta\|_1).$$

Elastic net 作为一个线性回归模型，通过 L_1，L_2 两个优先级作为正规化矩阵。这个结合允许模型处理稀少参数的数据。Elastic net 在对于多特征并且特征之间彼此高度相关的时候非常有用，Lasso 很可能随机的选择其中一个，而丢弃其他所有特征，起到降维的目的，而 Elastic Net 很可能选择多个。Elastic Net 算法一个实际的优势是可以在数据轮转下集成 ridge 稳定性。

用 ElasticNetCV 函数调参时我发现对应的最优参数应该是 alpha=0.0002，l1_ratio=1.2，但在 ensemble 后发现效果并不是最好的，最终只能选择 alpha=0.0004。

6. Xgboost 算法

XGBoost 是由 Tianqi Chen 最初开发的实现可扩展，便携，分布式 gradient boosting (GBDT, GBRT or GBM) 算法的一个库[3]。XGBoost 所应用的算法就是 gradient boosting decision tree，既可以用于分类也可 Gradient boosting

是 boosting 的其中一种方法，所谓 boosting，就是将若干弱分离器组合起来形成强分类器的一种方法。

XGBoost 的特点为计算速度快，模型表现好，这得益于该模型的三个设计。Parallelization：训练时可以用所有的 CPU 内核来并行化建树。Distributed Computing：用分布式计算来训练非常大的模型。Out-of-Core Computing：对于非常大的数据集还可以进行 Out-of-Core Computing(当然这个比赛应该用不到)。Cache Optimization of data structures and algorithms：提升速度。

我将原作者的 max_depth 由 20 改成了 10，以减小过拟合。

7. 模型组合

模型组合即 ensemble，通过将多个训练好的模型糅合在一起以减小单个模型带来的误差或过拟合。

最终的组合模型采用 0.45*Lasso 预测结果+0.25*Xgboost 预测结果+0.30*ElasticNet 预测结果。该权重组合是我尝试过的效果最好的组合。

四、 代码

发布在 https://www.kaggle.com/cxxacxx/have-a-try-2 上，可直接运行。目前的 Leaderboard 上我排名为第 55 名，分数为 0.11401。

参考文献

[1] Tibshirani R. Regression shrinkage and selection via the lasso[J]. Journal of the Royal Statistical Society. Series B (Methodological), 1996: 267-288.

[2] Zou H, Hastie T. Regularization and variable selection via the elastic net[J]. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 2005, 67(2): 301-320.

[3] Chen T, Guestrin C. Xgboost: A scalable tree boosting system[C]//Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2016: 785-794.