Rossmann 销售预测

Machine Learning Engineer Nanodegree

Capstone Proposal

刘冠利
2018年4月23日

领域背景

预测Rossmann商店的销售项目是一个Kaggle竞赛，预测销售使用商店，促销，和竞争对手的数据[1]。在统计学中，预测是统计推断的一部分。这种推理的一种特殊方法称为预测推理，可以通过统计推断的几种方法中进行预测。销售信息是随着时间的变化而变化时，我们需要根据给出的特定时间点的值来推断出未来某个时间点的值，这个过程被称为预测。

很明显这个预测问题是一个时间序列相关的问题。时间序列是在连续的等间隔时间点上的序列。因此，它是一个离散时间数据序列[2]。时间序列分析包括分析时间序列数据的方法，以提取有意义的统计数据和其他数据特征。它在我们的日常生活中是必不可少的，就像对天气的预测可以方便人们的出行，股票价格的预测有助于股东赚更多的钱，同时，销售预测也能帮助商家降低成本。时间序列是利用一个模型预测基于先前观测值的预测未来的值。在这个问题中，我们首先要分析数据集，然后用我们的模型来预测结果。

问题陈述

Rossmann在7个欧洲国家经营着3000多家药店。目前，Rossmann门店经理的任务是提前6周预测他们的日销量。商店的销售受到许多因素的影响，包括促销、竞争、学校和州假日、季节性和地点[2]。该项目的目标是预测位于德国不同地区的1115家门店的6周每日销售，这些门店的数据基础是2年半的历史销售。

数据集和输入

Kaggle提供了四个数据集:

> - train.csv - 包括销售的历史数据
> - test.csv - 不包括销售的历史数据
> - sample_submission.csv - 一个正确提交的示例
> - store.csv - 关于商店的补充信息

一些字段是自解释性的，比如DayOfWeek和Date。剩下的字段需要一些解释，下面是对那些剩余字段的描述。

> - Id - 表示测试集中的(存储、日期)双元的Id。
> - Store - 每个商店的Id。
> - Sales - 任何一天的营业额(这就是所预测的)。
> - Customers - 每一天的顾客数量。
> - Open - 该店是否营业的的标识:0 =关闭，1 =打开。
> - StateHoliday - 表明一个国家的节日。通常所有的商店，除了少数例外，都在国家假日休息。注意:所有学校在公共假期和周末都放假。a =公共假期，b =复活节，c =圣诞节，0 = None。
> - SchoolHoliday - (商店、日期)是否受到关闭公立学校的影响的标识。

> - StoreType - 区分4种不同的存储模式:a、b、c、d。
> - Assortment - 描述分类级别:a = basic, b = extra, c =extended。
> - CompetitionDistance - 到最近的竞争对手商店的距离。
> - CompetitionOpenSince[Month/Year] - 给出了最接近的竞争对手开门时间的大致年份和月份。
> - Promo - 该店是否在当日经营促销活动的标识。
> - Promo2 - Promo2 对一些商店持续和连续的促销的标识:0 =商店不参与，1 =商店正在参与。
> - Promo2Since[Year/Week] - 描述该商店开始参与促销活动的年份和星期。
> - PromoInterval - 描述Promo2启动的连续的间隔，如。“Feb,May,Aug,Nov”是指每年的2月、5月、8月、11月或11月开始新的一轮。

绝大部分的数据都应该被使用，我们可以从中选择一些重要的信息。客户和营业时间是至关重要的，因为它们与销售是正相关的。此外，促销信息也很有用，因为人们倾向于购买更多的商品。类型、竞争距离、节假日对销售也有一定影响。所以，在接下来的时间里会尝试这些信息来预测销售。

然而，数据集存在一些缺陷。例如，我在论坛上看622号商店在开放的列中有11个缺失值，大多数人认为商店是开放的[3]。如果它是封闭的，那么销售将是0，它不会计入分数。但是如果它是开放的，你预测0，那么你就会受到一些严重的惩罚。此外，有少量的门店的销售收入有6个月的数据缺失[4]。我们还必须估算缺失值，以得到一个稳定的结果。最后，我们还需要处理一些不适合训练模型的字段的值。例如StoreType和Assortment，它们的值都是字母，我们应该将StoreType的值变为1-4，Assortment变为1-3。

解决方案的声明

这个问题可以使用XGBoost来解决。XGBoost模型是树集成模型。它是一组分类回归决策树(CART)。CART是在给定输入随机变量X条件下输出随机变量Y的条件概率分布的学习方法。CART假设决策树是二叉树，内部结点特征的取值为“是”和“否”，左分支是取值为“是”的分支，右分支是取值为“否”的分支。这样的决策树等价于递归地二分每个特征，将输入空间即特征空间划分为有限个单元，并在这些单元上确定预测的概率分布，也就是在输入给定的条件下输出的条件概率分布[12]。用数学来准确地表示XGBoost模型，如下所示：

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i), f_k \in \mathcal{F}$$

其中K就是树的棵数，F表示所有可能的CART树，f表示一棵具体的CART树。这个模型由K棵CART树组成。确定一棵CART树需要确定两部分，第一部分就是树的结构，这个结构负责将一个样本映射到一个确定的叶子节点上，其本质上就是一个函数。第二部分就是各个叶子节点上的分数。模型的目标函数，如下所示：

$$\text{obj}(\theta) = \sum_{i}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k)$$

这个目标函数同样包含两部分，第一部分就是损失函数，第二部分就是正则项，这里的正则化项由K棵树的正则化项相加而来。正则项用于控制模型的复杂度。正则项里包含了树的叶子节点个数、每个叶子节点上输出的score的L2模的平方和。这使得学习出来的模型更加简单，防止过拟合，这也是xgboost优于传统GBDT的一个特性。

基准模型

我们选择的基准模型来自Kaggle Kernel的一份使用了XGBoost的模型[6]。该模型在Kaggle中的排名是66位，private score位0.11262。由于源码已经公布在GitHub上，因此可以运行该基准模型的代码来进行测试，并最终与我们的模型进行对比。

评价指标

由于这是一个预测相关的问题，因此评价指标不能准确或精确。有一些评价指标，如MAE(平均绝对误差)，MSE(均方误差)，RMSE[7]。至于RMSE(均方根误),它使用的平均错误,异常点的平均误差敏感,如果回到实现一定程度的回归值不是很合理,那么它的误差相对较大,从而对RMSE值有很大的影响,一般健壮性不是很好。我认为，可

以使用Kaggle官方网站中引入的RMSPE(均方根百分比误差)来评价模型[1]。RMSPE的表达式为 $RMSPE = \sqrt{\frac{1}{n} \cdot \sum_{i=1}^{n} \left(\frac{y_i - \hat{y_i}}{y_i} \right)^2}$. 其中 y_i 表示一家店在一天中的销售的实际值 $\hat{y_i}$ 是相应的预测值. 任何一天和商店的0销售都不会被计入。

项目设计

工作流包括五个部分。数据分析，数据处理，模型训练，基准模型的运行，结果比较和可视化。

首先，正如我之前提到的，这是一个时间序列预测问题。因此，在决定模型输入之前，我们应该先分析数据[8]。例如，我们需要识别最可能影响预测结果的重要因素。此外，对商店类型和相关分析的店铺活动、季节分解、趋势、自相关的分析也很重要。此外，在训练模型之前，我们应该解决的数据集存在一些缺陷。像商店622一样，在开放的列中有11个缺失值，大多数人认为商店是开放的。值0意味着它是封闭的，它不会计入分数，而它是开放的，你可以预测0，然后你就会得到一些严重的惩罚。还有少量的门店的销售收入有6个月的数据缺失，我们还必须估算缺失值，以得到一个稳定的结果。此外，我们还需要处理一些不适合列车模型的字段的值。例如StoreType和Assortment，它们的值都是字母，我们应该将StoreType的值变为1-4，Assortment变为1-3。

其次，在分析完数据之后，我们需要来划分训练集和验证集。因为这是一类时间序列的预测问题，因此我们需要按照时间顺序来划分数据。通过了解训练数据集并运行下面的R程序我们知道，数据的时间范围是2013-01-01至2015-07-31。所以我们将2015年之前的数据划分为训练集，2015年之后的数据划分为验证集。此外，在训练之前我们需要对数据进行对数处理，因为销量的尺度是不一致的，使用RMSPE是不考虑绝对尺度的。而XGBoost是使用RMSE优化的，因此对数处理后，模型的优化结果封符合预期的评价指标。

```
train = pd.read_csv("../input/train.csv", parse_dates = True, low_memory = False, index_col
= 'Date')
train.index
```

然后，在处理了数据集之后，我们使用XGBoost框架先根据训练集来得出一个初步的模型，然后使用验证集去验证模型。在验证的过程中我们可以得到所使用的特征对于模型的影响程度。因此，在初步训练的时候，我们就能够筛选出一些影响程度大的属性来进行训练。而且也可以得到使得验证结果较好的模型的参数值。在真正预测之前，我们会将训练集与测试集合并重新训练。这样得出的结果能刚好的反应真实结果。

接着，我们需要使用基准模型来训练模型，并在相同的情况下对结果进行评估。基准模型使用了随机森林的算法[6]。

最后，可视化两种算法的结果，使我们能够清楚地了解结果的好坏。结果可视化包括:数据集随着时间变化的趋势折线图和我们在其后面添加的预测值这样可以很好的观察训练集的整体趋势已经预测值是否符合整体趋势，是否在视觉上合理。由于我们并分别使用我们训练出的模型和基准模型进行了预测，因此我们可以对比两个模型的得分和结果的准度。此外，在图中应该对具有不同参数的模型的分数进行比较和对比，因为我们需要针对XGBoost模型给出最佳的参数设置。

引用

1. https://www.kaggle.com/c/rossmann-store-sales
2. https://en.wikipedia.org/wiki/Time_series
3. https://www.kaggle.com/c/rossmann-store-sales/discussion/17229
4. https://www.kaggle.com/nsecord/filling-gaps-in-the-training-set
5. http://xgboost.readthedocs.io/en/latest/model.html
6. https://github.com/mabrek/kaggle-rossman-store-sales
7. https://en.wikipedia.org/wiki/Root-mean-square_deviation
8. https://www.kaggle.com/elenapetrova/time-series-analysis-and-forecasts-with-prophet
9. https://en.wikipedia.org/wiki/Principal_component_analysis
10. http://scikit-

learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html#sklearn.ensemble.RandomForestRegressor

11. http://scikit-learn.org/stable/modules/grid_search.html#exhaustive-grid-search
12. https://baike.baidu.com/item/分类与回归树/20868547?fr=aladdin