KDD车流量预测解决方案:
多模型融合捕捉车流量变化模式

Blackswan Team

June 7, 2017

Contents

1 Introduction

Average-Tollgate-Traffic-Volume Prediction比赛的任务是针对5个不同的收费站类型(Tollgate-Direction),在给定前两个小时的车流量数据来预测未来2小时每隔20分钟的平均车流量。我们团队的最终解决方案是多模型融合的方案。它涉及到:统计时间序列模型(SARIMA), 线性模型(Linear regression),基于树的集成模型,神经网络(Neural Network)。

SARIMAX SARIMAX是对ARIMA模型的一个拓展模型,统计角度它可以捕捉到时序模型中季节性的部分,另一方面和原始ARIMA模型相比它允许在模型引入其他回归变量。

Linear Model 线性模型在实践中需要很强的统计假设。在该任务中,x给定问题中后每隔20分中的点稳定上涨或下降,可认为线性模型可以很好地捕捉到时序点中的趋势部分。

tree-based model 树方法的基本原理是基于一定的准则将高维特征空间划分成不同的子空间,用子空间的均值或Median进行预测,基于树的集成方法如randomforest,GBDT可以大大提升其预测准确度。

Neural Network 神经网络可视为包含了很多未知参数(Weights)的数学模型,许多$y_i = f(\sum w_i x_i - \theta_j)$相互嵌套而成(给定我们在比赛中采用Quantile regression最小化MAE);

这四类模型有着其不一样的统计假设和工作原理,使得我们最终融合的模型很好地捕捉到车流量的变化模式,取得很好的效果;在第一阶段接近结尾时，我们单独融合的结果分别名列leadboard的第1和第2;第二阶段的第一天排行榜结果显示,我们的融合结果远远超过第二名的结果，名列第一。

1.1 基本解题思路

一个时序上常用的假设是我们可以把时间序列分解成三部分:季节性(Sesonal component),趋势性(Trend component)和其他部分(Remainder)。

$$Y_t = S_t + T_t + R_t$$

结合该问题的时序结构,线性模型可以很好的capture其车流量序列的趋势部分,基于SARIMAX的可以capture时序中的季节性和其他部分。在实践中我们对两者的结果做了ensemble,取得了很好的效果。

这次比赛采用的评估指标是MAPE=$\frac{|y-\hat{y}|}{y}$. 给定经典机器学习回归模型的损失函数(loss function):

- MSE(gaussian distribution) $loss = \frac{1}{2}(y - \hat{y})^2$.
- MAE(laplace(quantile) distribution): $loss = |y - \hat{y}|$.

可以尝试的方式:

Log转换 : 如果我们对目标值y进行Log转换,一个简单的数学上的近似是:

$$|log\hat{y} - logy| = |log\frac{(\hat{y})}{(y)}| = |log(1 + \frac{\hat{y} - y}{y})| \approx |\frac{\hat{y} - y}{y}|$$

给定这种情况,如果我们进行目标值转换->模型训练->预测结果指数转换,最小化MSE,MAE的机器学习模型可以近似地看作是最小化整体的$MAPE$.

Quantile-Regression : Quantile regression理论上是最小化绝对误差,用以预测分布的Median点,给定我们问题中预测的真实值对我们是未知的,理论上一个稍为小于0.5的quantile点接近于最小化整体的MAPE.

该比赛另外一个很大的挑战是数据过少,相比利用Gaussian error,在具体的实现中我们采用了另一种简单有效的方式:将时间窗口进行平移,通过时间窗口平移,我们创造了大量新的有效的数据点。

	morning	afternoon
orignal	6:00-7:40 - 8:00-9:40	15:00-16:40 - 17:00-18:40
+5	6:05-7:45 - 8:05-9:45	15:05-16:45 - 17:05-18:45
-5	5:55-7:35 - 7:55-9:35	14:55-16:35 - 16:555-18:35

Table 1: time-window moving

1.2 文件代码框架

所有团队成员的代码包含在附件压缩包里,通过读取data/datasets/里的天池提供的原始数据开始执行,进行数据处理,建模生成最终结果。

- src/model2是Linear与SARIMAX融合模型:可以通过python命令python run.py一键执行。在后面章节里我们将提供更多模型和执行的细节。
- src/model1是基于Sklearn的ExtraTreeRegression的模型1,可以通过python命令python run.py一键执行。在后面章节里我们将提供更多模型和执行的细节。
- src/model3是基于Sklearn的多模型融合的结果,可以通过python命令python run.py一键执行。在后面章节里我们将提供更多模型和执行的细节。
- src/model4是是GBRT的模型,可以通过python命令python run.py一键执行。在后面章节里我们将提供更多模型和执行的细节。
- NN文件里是基于h2o.ai的神经网络模型:可以通过Shell命令sh -x run.sh一键执行。在后面章节里我们将提供更多模型和执行的细节。
- 主文件夹下ensemble.py,对所有模型结果进行ensemble,产生最终寨；可以通过python过python命令python ensemble.py一键执行。在后面章节里我们将提供更多模型和执行的细节。

2 整体模型描述

2.1 Linear与SARIMAX融合方案

2.1.1 框架介绍

本次Ensemble解决方案主体采用线性回归模型，对于其中该模型无法最佳拟合的部分采用SARIMAX模型进行填补，以达到最优的效果。 整体代码框架介绍:

- linear_regression.py，线性回归模型训练及预测文件。
- sarimax_preparation.py，sarimax模型特征提取文件。
- sarimax_model.py，sarimax模型训练以及持久化文件。
- sarimax_predict.py，sarimax模型预测文件。
- sarimax_run.py，sarimax模型运行文件。
- Linear_sarimax_ensemble.py，线性模型和sarimax模型融合文件。
- run.py，整体框架运行文件。

2.1.2 线性回归模型

1. 数据预处理: 对数据进行划分，训练集train1为9月20号到10月24号共5周的数据，且只取其中每天6:00 8:00，15:00 17:00四个小时的数据，train2为9月20号到10月24号共5周的数据，且只取其中每天8:00 10:00，

17:00 19:00四个小时的数据，测试集test为10月25号到10月31号每天6:00 8:00，15:00 17:00四个小时的数据。异常点处理方面，直接去除国庆节7天的数据结果，train1，train2去除国庆7天的数据后，都剩余4周的数据结果。

2. 模型训练及预测：对train1的四周数据进行线性加权去拟合test的结果，其中mape_loss最低的四个权重直接赋给train2，拟合出10月25号到10月31号每天8:00 10:00，17:00 19:00四个小时的结果。
3. 异常处理：其中tollgate_id为2，direction为0的收费站需要另外考虑，因为其最后两天，即10月30号和10月31号只允许etc车辆通过，所以将前四周数据划分为(周日和周一)，(周二到周六)两类，分别用四个参数去做线性拟合。

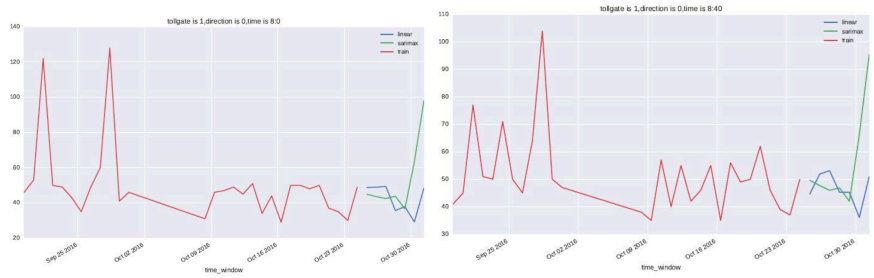
2.1.3 SARIMAX模型

sarimax输入的数据为两部分，一部分是时序数据，另一部分是外部数据，所以对五个id的收费站分别进行建模,时序数据取log，外部数据特征分>为几个部分:

1. 第一部分是ID,EarlyPeakTime,LatePeakTime,NormalTime的onehot，EarlyPeakTime对应7到10点之间都是早高峰，17点到20点对应晚高峰，其他时间都是NormalTime。
2. BeforeNationalDay,NationalDaySart,NationalDayEnd,Weekend,WorkingDay,WorkingWeekend，BeforeNationalDay对应9月30号，NationalDayStart对应国庆节前四天，NationalDayEnd对应国庆节后三天，WorkingWeekend对应10月8,9两天的工作日，Weekend是周末，WorkingDay是工作日。
3. BigRain,MediumRain,SmallRain,Sunny。利用训练集中天气数据乘以4，然后后向填补NaN值，使每20分钟都有天气数据，降雨量为0是Sunny，0 5是SmallRain，5 10是MediumRain，10以上是BigRain。

确定好时序模型的相关参数，对每个id分别建模然后训练，取出最后7天的结果即可。

2.1.4 基于线性回归和SARIMAX模型的Ensemble

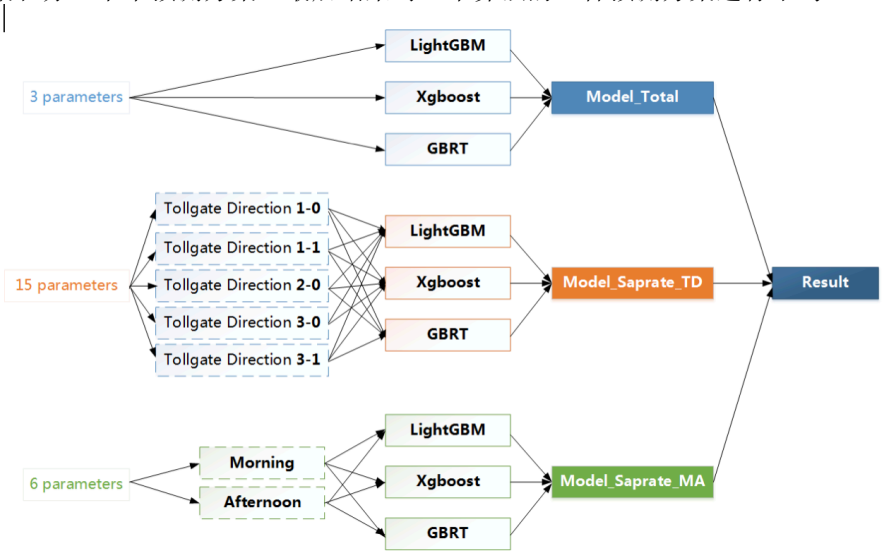


上图中绿线为sarimax模型预测结果，蓝线为线性回归模型预测结果，不难发现，对于id为1，direction为0的收费站线性回归并没有很好的捕捉到月末变化这一趋势，所以对于id为1，direction为0的收费站，我们采用sarimax模型的预测结果来进行替换，从而弥补了线性模型对该收费站的预测不准确的问题，达到了最优的Ensemble效果。

2.2 GBDT Model3

2.2.1 模型思想

回归预测的思想，统计历史时间窗口的流量，构造特征，预测未来的流量。使用了LightGBM、Xgboost和GradientBoostingRegressor三个GBDT的算法库。在模型训练和预测阶段，除了整体预测方案外，还使用了分Tollgate-Direction预测方案和分上下午预测方案。最后结果对三个算法的三种预测方案进行平均融合。



2.2.2 运行环境与代码结构

1. 语言：Python2.7及以上
2. 运行库：numpy1.12.1、pandas0.19.2、sklearn0.18.1、lightgbm0.1、xgboost0.6
3. 代码结构：BlackSwan/src/model_4/rice_run.py
    - slice_windows_sample()滑动窗口生成样本.
    - train_filter()样本选取与过滤.
    - lightgbm()微软开源LightGBM模型训练预测.
    - xgboost()Xgboost模型训练预测.
    - gbrt()sklearn中GradientBoostingRegressor模型训练预测.
4. 注意：三个模型都设置sample_rate=0.9，在本机可以复原结果。在其他机器上受随机种子seed影响，结果会略微不同。如有疑问随时联系本人scut_linhao@qq.com

2.2.3 模型细节

1. 滑动窗口构造样本，每1分钟滑动窗口，构造出20倍的样本.
2. label转换：y=log(1+y)。评估指标是mape，需要对小流量预测精度要求高
3. 特征：尝试加过很多特征，但是线上表现不好。选取最重要的特征：[month,day_of_week,day,holiday,hour,minute_point,slice_point,tollgate_id,direction]

4. 样本的多样性、参数的多样性filter_param=slice_window:1,day_filter_0930:[9,30],day_filter_1001:[10,1],tollgate_direction_filter:[0,0],slice_point_min:240,slice_point_max:1420,volume_min:1,volume_max:1000
5. 以上参数用于筛选样本，表示：1分钟滑动窗口；过滤0930、1001全天的样本；不过滤线路-方向的样本；保留[240,1420]分钟点的样本；保留流量为[1,1000]的样本
6. model_param=[lr:0.1,depth:6,tree:500,leaf:400,sample:0.9,seed:3]
7. 以上为模型重要参数，表示：学习率；深度、树棵树；最大叶子节点数量；抽样率；随机种子。

模型离线结果:

序号	模型	融合参数	离线成绩
1	lightgbm_result		0.1213
2	xgboost_result		0.1216
3	gbrt_result		0.1210
4	**ensemble_3model**	**1，2，3平均加权**	**0.1187**
5	lightgbm_separate_tollgate_direction		0.1154
6	xgboost_separate_tollgate_direction		0.1170
7	gbrt_separate_tollgate_direction		0.1149
8	**ensemble_separate_tollgate_direction**	**5，6，7平均加权**	**0.1133**
9	lightgbm_separate_morning_afternoon		0.1180
10	xgboost_separate_morning_afternoon		0.1181
11	gbrt_separate_morning_afternoon		0.1214
12	**ensemble_separate_morning_afternoon**	**9，10，11平均加权**	**0.1160**
13	**ensemble_3ensemble**	**4，8，12平均加权**	**0.1125**

2.3 ExtraTreeRegressionModel1

2.3.1 基本思路

我们将数据分为上下午两部分，分别利用上午6,7小时预测8,9小时,下午15,16小时预测17,18小时。以上午为例：由于预测的值为8:00-8:20,8:20-8:40,8:40-9:00,9:00:9:20,9:20-9:40,9:40-10:00,一共6个值，会导致误差较

parameters	list
n_estimators	[10, 20, 30, 40, 50, 80, 100]
max_depth	[2, 5, 8, 10, 15]
min_samples_split	[2, 5, 10, 15]
min_samples_leaf	[1, 2, 5, 10, 15]
max_features	[auto, sqrt, log2, None]

Table 2: grid search parameters list

大，所以我们选择预测8：00-9：00的均值，9：00-10：00的均值，在利用历史的值来计算8:00-8:20,8:20-8:40,8:40-9:00分别对于8点均值的比例ratio，这样我们只要预测两个值就可以。同时我们选择了sklearn里面的ExtraTreesRegressor模型来连续预测多个标签。

2.3.2 流量Ratio计算

我们使用历史对应天的平均值来计算对应的比例，而且对于每一个收费站，每一个方向都有各种的比例。所以得到的比例的个数是5*7*6个值。比如算周一8：00-8：20相对于8：00-9：00之间车流量的比例：算出历史中每一个周一8：00-8：20的值求一个平均值，算和每一个周一8：00：9：00的均值的比例。
该模型使用的特征如下:

- tollgateId的onehot特征;direction;
- 周几的onehot特征;
- 前两个小时每20分钟的车流量;
- 两个小时6个车流量对应的max,min,median,max,min,std;
- 原始一个小时内3个值的max，mean，median， max，min，std;
- 两个小时的均值的二项式展开;
- 是否特殊天，比如工作日，工作第一天，放假第一天，上班前一天，是否节假日等。

2.3.3 最优参数选择

我们利用网格搜索选择最优的参数，该过程大概需要3个小时来选择参数，所以代码中直接加载最优参数，如需验证参数，可以跑代码params来获取参数。
参数的初始设定范围为： 上午的模型的最优参数是:
best_params1 = max_features:log2, min_samples_split:2, n_estimators:30, max_depth:15, min_samples_leaf:2
下午的模型的最优参数为:
best_params2 = max_features:auto, min_samples_split:15, n_estimators:10, max_depth:15,min_samples_leaf:1

2.3.4 模型优势

利用ratio后我们只需要连续预测两个值，大大减小误差，同时我们会发现这个模型的预测结果的波动会更加大一些，而且数据越多，ratio的计算会更加的准确，模型效果也就越好，在复赛中的结果明显比初赛的成绩好很多。由于模型中random种子的设定与机子有关，所以结果可能会有小浮动的偏差，在本地可完全复现实验。

2.4 Sklearn多模型融合

该结果整合了GBRT,RandomForest,KNN三个模型的结果。主要思想是用前6个点来预测后6个点.在特征处理中,我们提取了天气，节假日，对利用时序的decomposition方式提取车流量的季节性特征，取得了良好的效果。

2.5 神经网络模型

所有Neural Network模型的代码是用R语言写的,采用h2o.ai公司开源的h2o.deeplearning框架:

- $installPackages.R$, 我们提供了所有运行模型所需要的第三方包的安装。
- $basicPreprocess.R$, 读取原始volume和weather数据，进行基本数据处理转换; 将结果存到”basicPreprocess.rda”(R的二进制, 类似python的pkl文件);
- $advancedPreprocess.R$ and $ATVHelper.R$: 特征工程和数据扩展(通过时间窗口平移k创造更多有效训练数据)。
- $cleanFeatureEngine.R$ and $ATVHelper.R$: 异常值过滤和天气,节假日特征;
- $DLNumericExtend.R$ and $DLHelper.R$: NN模型:训练单层神经网络,通过grid-search和early-stopping寻找最优的k个模型,取bagging的结果。 由于h2o.deeplearning自身工具的情况,在多线程情况下每次结果都有所不同 [1],在grid search情况下,每次选择的最优网络结构和参数列表也有所不同,但bagging结果不影响最终模型的准确度，可联系yitian1988@outlook.com或qq:935491064

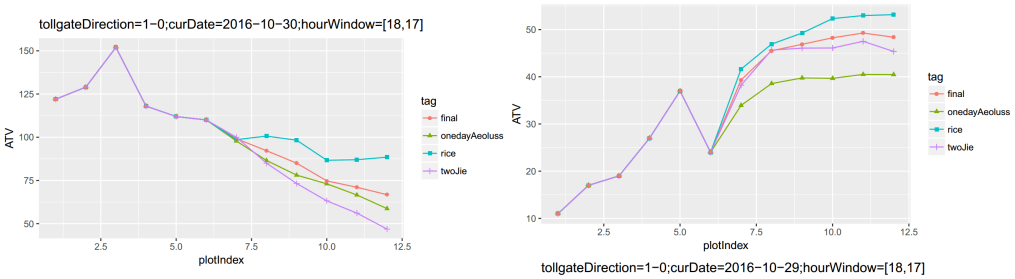
3 模型融合

我们模型融合包含两部分:一部分是基于第一阶段成绩和可视化结果的加权融合(weightedaveraging); 另一部分包含我们基于ensemble结果(差异很大),规则推理,对2-0的最后两天选择部分模型结果，我们将详细描述我们的分析和推理过程。

3.1 加权融合

BlackSwan-team是由初赛中三个独立的组合并而成,在融合中我们整体遵循初赛的融合框架:先小组内部进行融合,再进行三者之间的整体融合;在融合过程中,我们对三者的结果和融合的结果进行了可视化展示,根据可视化的结果和每次排行榜的结果在初赛的框架基础上进行了微调。
以下是三个子模型结果方案:

1. SARIMAX模型，LinearRegression模型，ExtraTreeRegression模型三个。具体的模型的思路在模型部分已经说明。在SARIMAX模型，LinearRegression模型融合的基础上，和ExtraTreeRegression模型结果按7:3融合，得到我们第一组的融合结果result_model1_model2_0.7_0.3.csv.，在分析图中的结果如线条twoJie所示，记为twoJie。
2. 使用lightgbm模型，模型介绍如模型章节描述，降多个类似的结果通过求平均来得到第二组的结果predict_rice_final.csv.,在分析图中的结果如线条rice所示,记为rice。

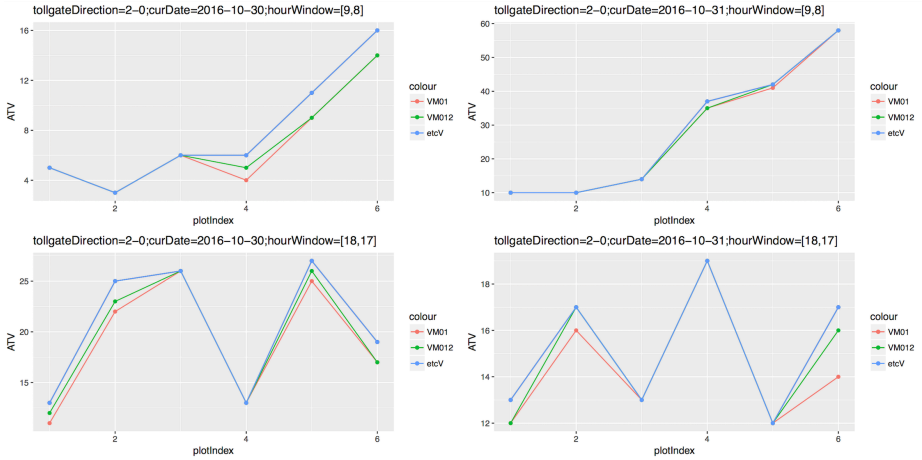


3. Sklearn模型和NeuralNetwork两个模型。两个模型采用1:1融合的方式。但是通过图发现，2-0收费站的10月30号和10月31号两天只有ETC的车流量，而我们的NeuralNetwork模型对于这两天的预测较差，所以针对改收费站的这几天我们只用了RandomForst模型的结果，得到的结果为result_NN_model3_0.5_0.5.csv，在分析图中的结果如线条onedayAeoluss所示,记为onedayAeoluss。

根据可视化结果和每个路口的绝对值分析,我们在最后一次提交时选择0.45:0.45:0.1的比例融合twoJie,rice和onedayAeoluss的结果。

3.2 规则推理优化

我们融合的结果绝大部分结果都十分接近,除了tollgate-direction=’2-0’和’1-0’的最后两天”2016-10-30”和”2016-10-31”部分结果,我们对其进行了细致分析发现这两天只允许有etc的车辆通过;



同时我们发现1-0的流量大幅度增长,根据出题方提供的道路拓扑图,我们做了如下商业假设:

- 平常时间1-0和2-0车流量和整体保持稳定;
- 当2-0进行限行(只限etc基于数据分析的结果),其他车辆将选择1-0道路通行;

基于这个假设,我们选择模型结果里最接近2-0历史上etc车的流量;同理我们也对1-0同期车流量进行了优化(总量-tollgate-direction=2-0)。

4 团队成员介绍

周杰，华东师范大学博士研究生，主要研究机器学习，深度学习和自然语言处理。在本次比赛中负责部分模型实现和模型融合。
林杰，南京理工大学硕士研究生，主要研究方向是深度学习，计算机视觉方向，在本次比赛中负责部分模型的实现工作。
林浩，腾讯社交网络事业群算法工程师，主要从事推荐系统、基础算法研究等工作。在本次比赛中负责部分模型的实现和模型融合。
郭阳，京东Y事业部算法工程师，在京东主要从事统计建模及模拟分析平台开发相关工作。在本次比赛中负责部分模型的实现和第二阶段数据分析。
陈艺天,京东Y事业部资深算法工程师,在京东主要从事统计建模与优化决策相关的工作,在本次比赛中担任队长，负责部分模型实现和整体数据分析。

References

[1] DeepLearning_Vignette.pdf Appendix A:Complete parameter list, page-24, h2o.ai offical website

[2] MD Zeiler, Adadelta: An adaptive learning rate method, 2012.