摘要

关键词

供水总量预测的背景

下达生产计划，供水调度工作。

模型简介

Prophet是Facebook公司2018开源的一个时间序列预测的算法，针对对时间序列基于 Python 和R语言的数据预测工具，其工作发表在论文《Forecasting at scale》(Taylor S J, LethamB. Forecasting at scale[J]. The American Statistician, 2018, 72(1): 37-45)。prophet 算法不仅可以处理时间序列存在一些异常值的情况，也可以处理部分缺失值的情形，还能够几乎全自动地预测时间序列未来的走势。除此之外，为了方便统计学家，机器学习从业者等人群的使用，prophet 同时提供了 R 语言和 Python 语言的接口。从整体的介绍来看，如果是一般的商业分析或者数据分析的需求，都可以尝试使用这个开源算法来预测未来时间序列的走势。用于预测的时间任务常具有以下特征：

对于历史在至少几个月（最好是一年）的每小时、每天或每周的观察

强大的多次的「人类规模级」的季节性：每周的一些天和每年的一些时候

事先知道的以不定期的间隔发生的重要节假日或者特殊事件（如，十一、五一、如疫情）

合理数量的缺失的观察或大量异常

历史趋势改变，比如因为产品发布或记录变化

非线性增长曲线的趋势，其中有的趋势达到了自然极限或饱和。针对中长期由很多帮助。

模型原理

模型的建立

空值、判断

#判断数据框里面所有不是有限数据的数，可以排除null，np.nan,np.inf

sup\_water[~(np.isfinite(sup\_water).all(axis=1))]

特征工程：扩增特征变量，或者浓缩特征变量，或者对特征变量进行选择。将文字便来给你转化为编码。具体编码转换方式见数据处理笔记。

from sklearn.preprocessing import PolynomialFeatures

poly\_features = PolynomialFeatures(degree=3,include\_bias=1)

X\_poly = poly\_features.fit\_transform(sup\_water[['最低温度','最高温度','平均温度']])

时间序列独特的离群值判断

am = sup\_water['最低温度'].rolling(window=3,min\_periods=0,center=True).mean()

diff = sup\_water['最低温度'].values-am

sup\_water.loc[diff.abs()>diff.std(ddof=0)\*3,]

sup\_water['上限']=am+diff.std(ddof=0)\*3

sup\_water['下限']=am-diff.std(ddof=0)\*3

异常值最好的方法是移除它们，而 Prophet 是能够处理缺失数据的。如果在历史数据中某行的值为空（ NA ），但是在待预测日期数据框 future 中仍保留这个日期，那么 Prophet 依旧可以给出该行的预测值。过大的异常值会破坏季节性效应的估计，最好移除。

模型构建中的参数确定

初始周期应该足够长，以捕获模型的所有特性，特别是季节性和额外的回归变量:对年的季节性至少保证一年，对周的季节性至少保证一周，等等。

相关关系检验。多项式

增加回归量比如天气温度可以在网上爬取

变化速率灵活性更大时（通过增大参数 changepoint\_prior\_scale 的值），预测的不确定性也会随之增大。原因在于如果将历史数据中更多的变化速率加入了模型，也就代表我们认为未来也会变化得更多，就会使得预测区间成为反映过拟合的标志。

节假日特殊因素标识

对于春节固定农历日可以采用涉及以农历节日的假期，需要用以下代码或者阳历日期

import sxtwl。新冠疫情影响时长具有不确定性，对市民活动的限制程度也不一致。因素疫情影响分为三个阶段，武汉封城为开始点，广州市降级为二级响应、武汉解封三个时间讲疫情影响分为三个阶段。分别计算影响度。

季节性周期值，调整傅里叶级数让周期平滑，或者过拟合。

如果训练集的周期有规律的不完整，比如每天只有0-6点的数据，那么天的季节性在一天剩下的时间里是不受约束的，估计也不准确。周期性影响量的判断不完整，未来dataframe里面的时间窗和历史数据也应该保持一致。比如每周只有工作日的数据，未来也应只有工作日。

同样的历史数据集使用的是月数据。只有每个月第一天的数据，对于其他天的周期效应是不可测的。因此当可以通过在 make\_future\_dataframe 中传入频率参数的频率必须和历史训练数据集保持一致。

Prophet 只会返回趋势中的不确定性和观测值噪声的影响。你必须使用贝叶斯取样的方法来得到季节效应的不确定性，可通过设置 mcmc.samples 参数（默认下取 0 ）来实现。

交叉验证

时间序列交叉验证原理

因为时间序列的样本之间是无法交换的，所以没办法像KFold交叉验证一样把数据集切分成若干份训练集和测试集（每份训练集的模型超参数是一样的）。一个比较好的思路是按照时间顺序设置。这里有三个参数：

horizon: 模型预测的范围，如从cutoff点开始数未来30天

period: 每两个 cutoff 点之间的间隔，也就是间隔多少期开始新的一份训练集。cutoff为horizon的开始点。

initial: 用于训练的日期范围，如730天

有自动交叉验证函数

#划分数据

from fbprophet.diagnostics import cross\_validation

df\_cv = cross\_validation(m, initial='730 days', period='7 days', horizon = '365 days')

rmse\_ave = df\_cv.groupby('cutoff').apply(fun\_rmse).mean()

df1.loc[i-1] = [param,rmse\_ave]

把每个超参数以及 rmse\_ave 记录进入数据框以供选择

df1 = df1.sort\_values(['rmse']).head(10)

for param,rmse in zip(df1['param'],df1['rmse']):

print(param,rmse)

查看从预测未来1days到365days的rmse，mape，coverage等参数

from fbprophet.diagnostics import performance\_metrics

df\_p = performance\_metrics(df\_cv)

可以指定某个参数随着预测时期的增长的每个数值和平均值画出来

from fbprophet.plot import plot\_cross\_validation\_metric

fig = plot\_cross\_validation\_metric(df\_cv, metric='mape')

预测效果评估

m.predict (df = futrue, …)

意义