**Prophet模型预测新冠肺炎疫情影响下的城市供水总量**

谢杰

（广州市自来水有限公司，广州 510600）

摘要：该文以Prophet模型为工具，详细介绍了如何糅合时间趋势，季节趋势，假期因素，特殊事件，以及相关因素来对新冠疫情期间的城市供水总量的变化进行预测，并且检验预测效果，判断供水总量何时回复正常。

关键词：时间序列；供水总量；新冠肺炎疫情；预测；Prophet。

**Title**

XieJie

(Guangzhou Water Supply Co., Ltd，GuangZhou 510600，China)

**Abstract:**Forecasting is a common data science task that helps organizations with capacity planning, goal setting, and anomaly detection. Despite its importance, there are serious challenges associated with producing reliable and high quality forecasts — especially when there are a variety of time series and analysts with expertise in time series modeling are relatively rare. To address these challenges, we describe a practical approach to forecasting “at scale” that combines configurable models with analyst-in-the-loop performance analysis. We propose a modular regression model with interpretable parameters that can be intuitively adjusted by analysts with domain knowledge about the time series. We describe performance analyses to compare and evaluate forecasting procedures, and automatically flag forecasts for manual review and adjustment. Tools that help analysts to use their expertise most effectively enable reliable, practical forecasting of business time series.

**Key words:** Time Series, Statistical Practice, Nonlinear Regression

1.引言

新冠肺炎疫情传播给社会造成了生产生活造成了极大的影响，同时也对供水总量的预测提出了心得挑战，在水资源日益紧缺的今天，准确预测供水总量的变化，对判断疫情对城市供水的影响量，合理下达供水生产计划都有重要意义。

2.模型介绍

Prophet是Facebook公司2018开源的一个时间序列预测的算法，其工作原理发表在论文《Forecasting at scale》(Taylor S J, LethamB. Forecasting at scale[J]. The American Statistician, 2018, 72(1): 37-45)中。在实际生活和生产环节中，除了季节项，趋势项，剩余项之外，通常还有节假日的效应和相关影响因素的回归量。所以，在 prophet 算法里面同时考虑了以上五项：

1587300914(1)

其中g(t)为趋势项，回归量可以附加在趋势项中。s(t)为季节量，h(t)为假期项，为随机项。

2.1趋势项模型分量

趋势项有两个重要的函数，一个是基于逻辑回归函数（logistic function）的，逻辑回归用于模拟非线性的饱和增长（减少）趋势。另一个是基于分段线性函数（piecewise linear function）的，在线性函数中不需要指定自然极限或者饱和值。

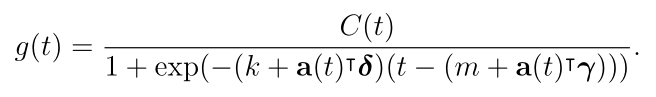
逻辑回归可以写成：

1587299049(1)

C就是 Capacity，也就是所谓的承载能力，k，m

在 Prophet 里面，如果趋势发生多次改变则是需要设置变点的位置，每一段的趋势和走势也是会根据变点的情况而改变的

设置完变点后的分段逻辑回归方程：



C(t)意味着随t变化的回归曲线增长的上限度。

线性回归方程中所周知为：

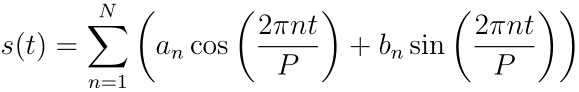
1587300414(1)

设置完变点后的分段线性回归方程为：

1587299702(1)

其中κ表示增长率（growth rate），δ表示增长率的变化量，ｍ表示 offset parameter。

2.1 季节性趋势。模拟线性趋势或者还能够几乎全自动地预测时间序列未来的趋势。针对中长期由很多帮助。



P为季节周期天数，比如年为365.25天，周为7天。N为傅里叶级数。N越大模拟越细腻当容易过拟合。一般年为10，周为3。

2.3 假期因素

1587300670(1)

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）

1. 应用实例

为了方便使用，Prophet算法同时提供了 R 语言和 Python 语言的接口。本文采用Python语言为工具实现

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。

* 1. 修复离群值

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))]

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。

* 1. 找出有代表性的回归量

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']])

* 1. 构建模型超参数

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。

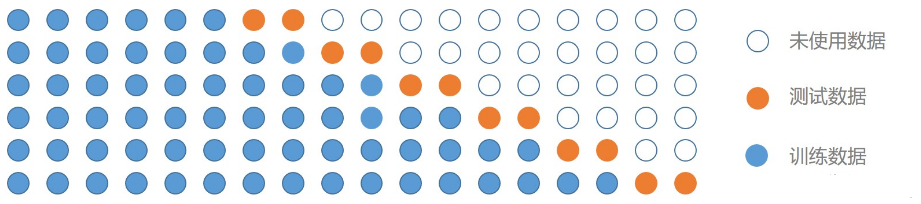
同样的历史数据集使用的是月数据。只有每个月第一天的数据，对于其他天的周期效应是不可测的。因此当可以通过在 make\_future\_dataframe 中传入频率参数的频率必须和历史训练数据集保持一致。

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。

3.4 交叉验证评估模型

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，如上图黄色点为预测范围。

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点为黄色开始的点。

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')

4.结论

m.predict (df = futrue, …)

参考文献

[]

[]