**参赛队员1**

**姓名**：

**游乐园客流疏导方案及酒店房间预定量预测**

摘要

面对日益拥堵的游乐园，建立起合适的模型来模拟客流，提供高效的方案及时疏导游乐园中游客，将会成为合理规划管理园区方式的新趋势。同时，不仅需要对游乐园中客流进行预测，对游乐园周边进行合理预测也是有必要的。有效的预测能带给游乐园更多的收益，尤其是对游乐园中的酒店中被预定房间数。

本文通过已知每个项目场地的排队人数以及正在进行的项目剩余时间，建立相关约束条件，为使游客游园达到最优，规划出最合理的客流疏导方案，为每一位在节点处的游客提出合理建议。针对本文第二个问题，利用数据具有周期性的特点，采用时间序列模型进行描述，并进一步做出预测。

在问题一中，对约束条件以及相关的条件进行简化，以游客用相对短的时间能体验完所有项目为目标，利用lingo进行整数规划。但计算中越被简化，精度加快减小。问题二中首先采用EViews确定其适用于该事件序列模型，并找出周期数。接着借助matlab利用位移平均法求出每个星期中不同天的季节指数。将数据除去星期因素的影响，用处理后的每周平均值进行拟合求出季节拟合Fourier函数。稍加修正后，将全部因素相乘最后得出模型最后的拟合函数。

模型一的思想是，联系所有节点同时进行疏导，使得每个节点的拥挤程度随不断的疏导进而趋向一个平衡状态，并使每个节点的拥挤程度不会有较大的差异。

模型二既能很好的描述一年中的淡旺季，还能体现出每周不同天数对酒店房间预订的影响。模型拥有很高的契合度，很较完美的描述数据以及对未来短时间进行预测。

模型一可以结合交通流为城市交通疏导提供参考依据，或者结合排队论为消费者提供选择建议富有实际效益。而模型而可以很有效的描述具有周期性的时间序列模型，尤其是带有季节因素影响的。

关键词：季节时间序列模型 自相关分析 lingo整数规划

**1问题的重述**

1.1引言

约上亲朋好友一起去游乐园度假，对休闲放松，增进感情，不失为一种很好的选择。加上游乐园的周边服务做的愈来愈出色的情况下，这种休闲娱乐方式已经被大多数生活在城市里的人们所接纳。即将开业的Youth游乐园将以拥有最多过山车的游乐园出现在更多人的视野中，丰富多彩的娱乐体验以及假日酒店的优秀服务将会吸引更多的人们向往。

1.2问题的提出

将会有数以万计的人们前往游乐园里享受美好时光，本文为了能使游客们更好地体验游乐园，现提出以下两个问题。

（1）、游乐园的客流情况必须得以预测以便及时引导游客游园分散客流。给出每场项目所能容纳游客数以及持续时间以及每场项目之间的距离，通过分析每个场地的拥挤程度，每隔一段时间为在场游客给出建议，即是否在此场地排队等候或者前往下一个场地观看项目。并建立一个公园模型加以验证。

（2）、园内的皇冠假日酒店可以为游客提供方便舒适的住宿条件，合理预测酒店房间预定数可以为酒店提前规划做好工作准备。现以2015年全年预定房间数据，找出相关影响因素，用合适的模型加以描述，并为2016年的一月至三月每天被预定房间总数做出合理预测。本文中，被预订房价总数为此日已被预订与新增被预定之和。

2 问题分析

2.1问题一

该问题是在游乐园内部整体规划人流的问题，目的是让每个游乐点的拥挤程度相同，不至于造成一个项目场地过于拥挤，而其他游乐点相对空闲的情况。问题需要分析每一个项目场地中，游客不同的选择所需要等待的时间。具体为每个场地的游客去相邻场地体验上项目的时间，以及留在该场地排队等待至体验完项目的时间。作为游客，肯定会选择花费时间较短的方式使得自己尽快地体验上项目。所以对一位游客来说，合理的规划要使得其在各项目上排队的时间之和要最短。

而作为疏导方案的实施方来看，实施方需要使得每时每刻任何一个场地不要与相邻场地的拥挤程度产生较大悬殊。否则就会产生局部拥堵，给场地项目产生额外压力。所以实施方要通过合理疏导，使得前往相邻场地体验上项目的时间不大于在该场地排队等候，所以可以促进游客选择下一目的地，而不会造成不必要的拥堵。

综上所述，该问题为线性规划问题。根据题目给出的条件，列式表达出各个选择需要的时间，规划出游客遍历所有场地所需要的最小排队时长总和。以此求解出单位时间内，每个游乐项目场地的往相邻各个方向疏导人数的建议。

2.2问题二

该题旨在根据已有数据分析未来一段时间的酒店房间每天预订数。在此问题中，应该考虑到季节,工作日/周末,法定假日,寒暑期等各因素对2015年每天房间预订总数的影响。

为了解决问题，应先对现有数据进行处理。已知2015年每一天新增的房间预定数，通过简单的统计，可以得到2015年每天房间预订总数，即每天晚上所有被“占用”的房间数。利用相关软件处理得到数据的特点（周期性，总趋势，变异点），根据模型的特点选择合适的数学模型，再利用模型进行预测接下来的三个月的房间预订量。最后对模型做出检验和评价。

3 模型的假设

1. 假设所有在园游客在相邻场地间的道路上的行走速度均为一致。为方便计算假设速度为每秒1米。
2. 假设出入口可容纳人数极少，而每场持续时间以及剩余时间无穷大。
3. 假设在每个场地的正在排队等候的人数，以及在该场地的项目还剩余时间是可测的，且在模型每次进行预测前，可以先被输入模型。
4. 假设每个场地的项目对游客的吸引程度差异不大，即游客仅以需要等待时间长短为评判标准选择体验项目。
5. 以随机生成的模拟园区游客数量为起始情形，开始利用模型进行疏导。假设此初始状态下的园区中，游客绝大多数在个项目场地。且在两相邻场地的道路上的游客数量在实施模型进行疏导前后的变化量为小变化量。
6. 假设每两场项目间无空余时间，即一场节目结束后马上开始第二场。
7. 假设每隔单位时间，疏导模型做出一次反应，每个节点的工作人员可以结合游客的意愿按照疏导模型给出的人数进行疏导，且绝大多数游客可以听从疏导指挥。
8. 假设影响酒店被预定房间数因素不与游客本身有关，例如游客的喜好等。不考虑酒店房间成功预定后被取消的情况。

4 符号的说明

|  |  |
| --- | --- |
| **符号** | **含义** |
| **t** | 时间轴 |
|  | 所有节点集合 |
|  | 与i节点相邻的所有节点的集合 |
|  | 在i点的本场项目剩余时间 |
|  | 处于i点的人留在i点排队体验上项目所需时间 |
|  | 处于i点的人走到j点并体验上项目所需时间 |
|  | 此时在i点排队等待体验上项目人数 |
|  | 每场项目的容纳量（已知） |
|  | 每场项目的持续时间（已知） |
|  | 单位时间内i点人数变化量 |
|  | 单位时间内建议从i点前往j点人数 |
|  | 从i走到j所需时间（与ij路程成正比，且=） |
|  | 内i点人数变化量，即 |

5模型的建立

5.1模型一

将游乐园地图节点间的联系看作无向图，以每一条路线作为一条边，以每个场地作为一个节点，将O,A,B,C……J分别标记为0,1,2……10，OA,OB,AB……为每条边，这样就可以构造出一个网络图（如下图所示）。图中共11个节点，16条边。

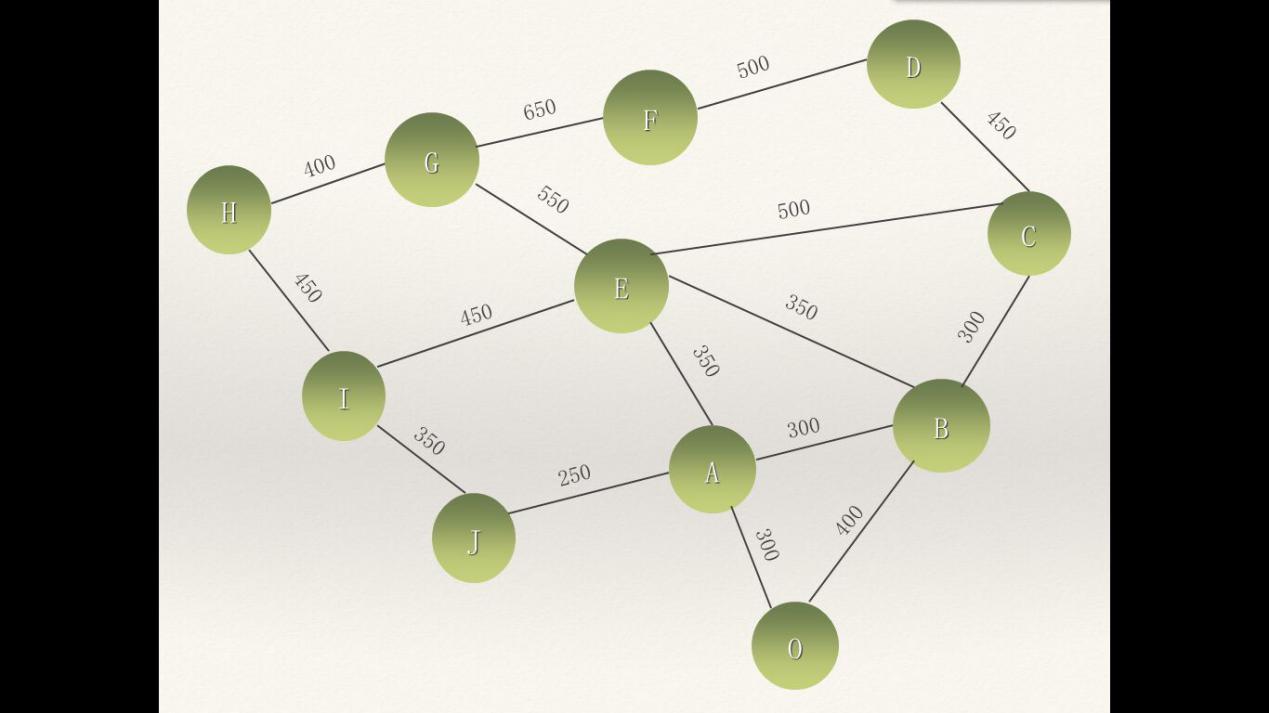


图5-1 游乐园节点无向图

对于每个节点，储存6个数据：

1）节点名称；

2）与节点连接的边；

3）节点每场容纳量；

4）节点每场持续时间；

5）节点在t时距离本场节目结束所需时间；

6）节点在t时排队的人数；

对于每条边，储存3个数据：

1）边指向的节点；

2）指向下一条边的指针；

3）边长的值；

对于t=0s抵达i节点的游客，其在i节点体验上项目所需要的时间为该游客需排队时间，而其去j节点体验上项目的时间不仅包括了该游客需要在j节点的排队时间以及从i前往j节点的路途时间。

假设第(+1)位游客到达i场地，即该游客前已有位游客正在排队。若其选择在此排队等候，则其需要花费时间可由下式求得：



其中“”为向下取整函数。如未特加说明，本文将向下取整函数简称取整函数。式中，表示该位游客待正在进行的项目结束后还需要等候的场次数。

同理若该位游客想前往j场地体验项目，则其需要花费的时间可由下式求得：



式中，的含义为当该位游客花费了从i到达j后，待那时正在进行的项目结束后还需要等待的场次数。表达了当游客抵达j点时，j点的人数变化。且，其中。

以I点为例，与I点相邻的点有J、E、H三个点。对于第（+1）位达到I点的游客现有4种选择，体验上项目的所需时间分别为：









为了尽量使游客选择后三种方案，所以、、和满足以下关系：



此不等式组即表明游客选择后三种方案将会更快体验上节目。

而对所有来说，其可以用32个，且线性表示。即得以下线性方程组:



其中，为32阶的系数矩阵（见附录三）。

以上说明了此模型中所有数据都是相互关联的。由于我们只能在每个节点的路口进行疏导，所以当游客踏上去下一个项目场地的路上时，实施方疏导的目的就达到了。而对于游客来说，能排上相对短的队伍而体验上更多的项目是游客所想要的，即使得达到最小值。

综上分析，可以建立以下模型：

，

且。

5.2模型二

首先，将数据导入至matlab，另存为数据备用。用脚本编写代码计算出每一天被预定的房间总数，并计算每月房间预定总数日均值，如表5-2。

表5-2 每月份房间预订总数的日均值

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 月份 | 2015-1 | 2015-2 | 2015-3 | 2015-4 | 2015-5 | 2015-6 | 2015-7 |
| 房间预订总数日均值 | 0.56 | 0.32 | 6.16 | 35.53 | 44.77 | 45.5 | 44.06 |
| 月份 | 2015-8 | 2015-9 | 2015-10 | 2015-11 | 2015-12 | 2016-1 | 2016-2 |
| 房间预订总数日均值 | 39.77 | 40.5 | 47.61 | 45.6 | 42.97 | 8.81 | 2.00 |

经过计算，这十四个月中391天的房间预订总数的日均值为31.45，方差为458.8176。

继续对数据进行简单的统计处理，我们可以发现约有84%的顾客会提前预订房间。在这些顾客中，有一成的顾客会提前一个月预定，甚至有一位顾客提前了103天预定房间。所以，2015年前几个月的（1月、2月、3月）每天房间预定总数，没有包含2014年底的几个月就已经预定的。所以导致了2015年前几个月每月房间预订总数日均量极低。

为了处理这个问题，减小误差的同时避免损失过多的数据。所以我们直接把最受影响的2015年一月份与2016年一月份的数据以相同星期数且日期最接近的方式相加，得到修正数据，这样避免了剔除四个月数据导致数据缺失过多。

我们利用EViews软件对修正后的数据进行分析。导入数据，进行自相关性与偏自相关分析。得到以下图：

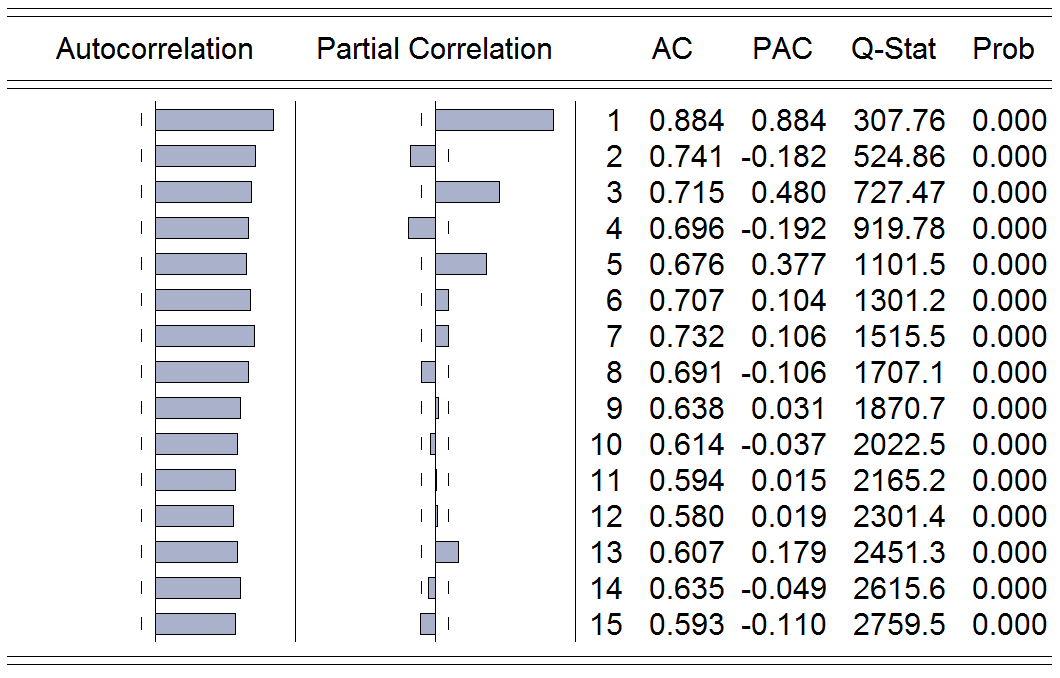


图 5-1 修正数据的自相关与偏相关数分析

由于数据的自相关系数没有很快的收敛到零，所以数据是不平稳的[1]。在对数据进行一阶差分的处理下，再进行自相关分析，得结果如图。发现在趋势已基本消除的情况下，k=7与k=14时，数据的自相关系数分别为0.292与0.311，大大超过了随机范围的区间，与0有显著差异，表明数据有周期为7天的数据波动。

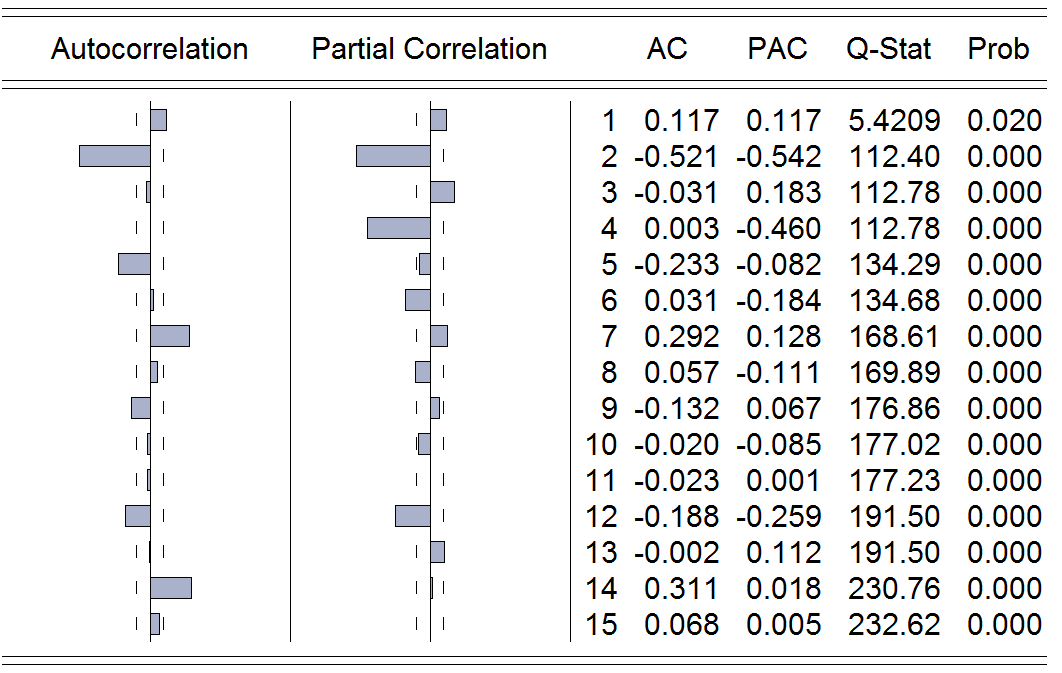


图 5-2 一阶差分后数据的自相关与偏相关数分析

综上，我们选择季节时间序列模型(seasonal ARIMA model)。

6 模型的解决

6.1模型一的解决

先用附录一种的C++代码随机生成一个游乐园初始状态数据，见表6-1。将所有条件与约束条件与模型输入lingo，利用lingo解决。然而完整的模型，lingo跳出“out of memory”的问题。

表6-1 游乐园初始状态数据

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 节点 | A | B | C | D | E |
| 初始人数 | 4639 | 3721 | 2163 | 1575 | 2861 |
| 节点 | F | G | H | I | J |
| 初始人数 | 700 | 1973 | 3852 | 3031 | 1610 |

这是由于存在取整函数，所以这使得原本就是整数规划的问题更加复杂化，这就迫使我们寻求进一步的简化计算。

考虑到取整函数中的可以为32个所描述。所以可以先对进行整数规划，待被解出后，再利用解线性方程组解出剩下的。相关的lingo程序语言见附录二。

虽然还被提醒有可能不是最优解，如图6-1。

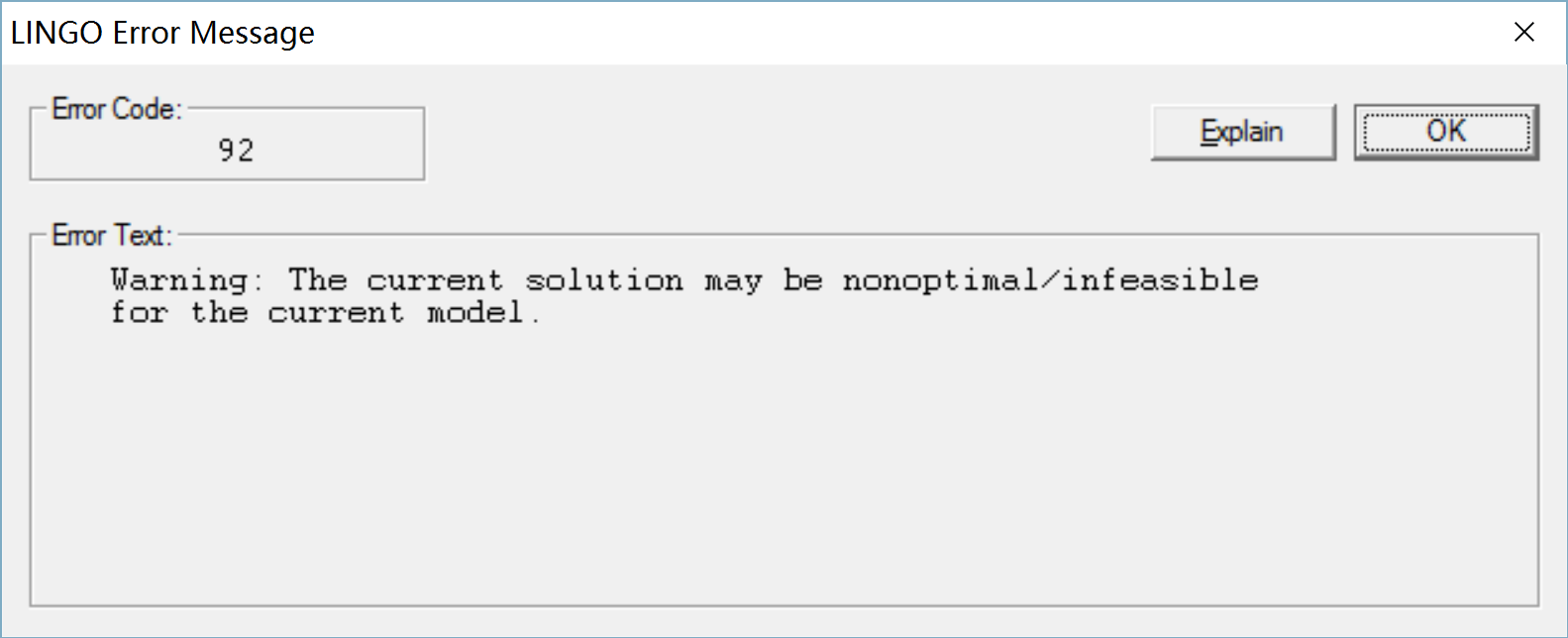


图6-1 来自lingo的警告

再通过matlab借助qr分解解出32个的值，即为此刻疏导系统对游乐园每节点的游客进行建议。所以此模型是可以对任何时刻的游乐园情况进行疏导。为了提高疏导效率，疏导系统应取每场项目持续时间的最大公约数，即15s。

6.2模型二的解决

根据季节时间序列模型的特点[2]本文将分为以下四种因素。

(1)、趋势因素（T）

(2)、星期因素（D）

(3)、季节因素（S）

(4)、不规则因素（I）

并选择乘法模型，即。因为存在星期因素，为方便计算，我们取从2015年1月11日周日至2016年1月23日周六区间的数据进行计算。

6.2.1趋势因素的求解

由于数据只有13个月（一年多一个月），所以不足以反应数据的长期趋势。所以忽略长期趋势因素，即认为趋势因素等于1。

6.2.2星期因素的求解

由于十一月份既不受数据误差的影响，且离十一假期与元旦假日时间间隔较长，所以十一月份的连续四周是相对很“正常”的。以十一月份为原始数据使用matlab编写语言求出七项移动平均值，用原数据初移动平均值求出季节比率（代码见附录四），求平均值计算出每周不同天数的季节指数。如下表6-2：

表6-2 每周不同天数的季节指数

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 时间 | 星期日 | 星期一 | 星期二 | 星期三 | 星期四 | 星期五 | 星期六 |
| 季节  指数 | 0.8056 | 0.8673 | 1.3617 | 1.2599 | 0.7976 | 0.9029 | 0.9989 |

6.2.3季节因素的求解

我们将区间内数据按照星期数除以相应的季节指数，得到数据。

利用matlab对数据进行拟合。利用matlab进行Fourier拟合，一级Fourier拟合的效果较差，而三级Fourier拟合不能很好的描述数据周期性。所以选择二级Fourier拟合，得到拟合方程：



6.2.4不规则因素分析

由于各个假期的分布毫无规律，且2016年一至三月份只有元旦假期(寒假假期已包含在了季节因素中)。所以只需对2016年1月1日至1月3日的数据加以调整。计算元旦假期因素。

6.2.5模型的解出

将上述的所有因素相乘，得到X的拟合函数。将拟合结果值为负的调整至零。画出拟合图如下。

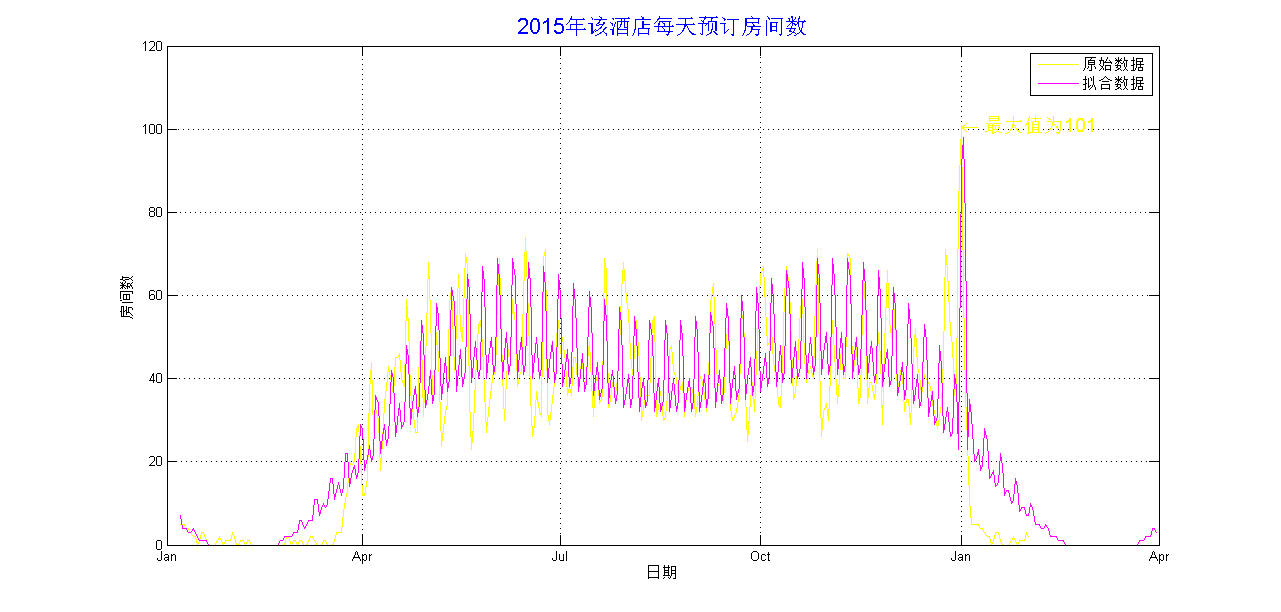


图6-2 原始图形与拟合图形

6.2.6模型的预测

根据模型，对2016年1月至3月的被预订房间数进行预测，得下表6-3。

表6-3 2016年1月至3月每天房间预订总数预测表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **日期** | **房间预定总数** | **日期** | **房间预定总数** | **日期** | **房间预定总数** |
| **2016-1-1** | 91 | **2016-1-31** | 7 | **2016-3-1** | 0 |
| **2016-1-2** | 98 | **2016-2-1** | 7 | **2016-3-2** | 0 |
| **2016-1-3** | 77 | **2016-2-2** | 10 | **2016-3-3** | 0 |
| **2016-1-4** | 23 | **2016-2-3** | 9 | **2016-3-4** | 0 |
| **2016-1-5** | 35 | **2016-2-4** | 5 | **2016-3-5** | 0 |
| **2016-1-6** | 31 | **2016-2-5** | 5 | **2016-3-6** | 0 |
| **2016-1-7** | 20 | **2016-2-6** | 5 | **2016-3-7** | 0 |
| **2016-1-8** | 21 | **2016-2-7** | 4 | **2016-3-8** | 0 |
| **2016-1-9** | 23 | **2016-2-8** | 4 | **2016-3-9** | 0 |
| **2016-1-10** | 18 | **2016-2-9** | 5 | **2016-3-10** | 0 |
| **2016-1-11** | 19 | **2016-2-10** | 4 | **2016-3-11** | 0 |
| **2016-1-12** | 28 | **2016-2-11** | 2 | **2016-3-12** | 0 |
| **2016-1-13** | 25 | **2016-2-12** | 2 | **2016-3-13** | 0 |
| **2016-1-14** | 16 | **2016-2-13** | 2 | **2016-3-14** | 0 |
| **2016-1-15** | 17 | **2016-2-14** | 2 | **2016-3-15** | 0 |
| **2016-1-16** | 18 | **2016-2-15** | 1 | **2016-3-16** | 0 |
| **2016-1-17** | 14 | **2016-2-16** | 1 | **2016-3-17** | 0 |
| **2016-1-18** | 15 | **2016-2-17** | 1 | **2016-3-18** | 0 |
| **2016-1-19** | 22 | **2016-2-18** | 0 | **2016-3-19** | 0 |
| **2016-1-20** | 19 | **2016-2-19** | 0 | **2016-3-20** | 0 |
| **2016-1-21** | 12 | **2016-2-20** | 0 | **2016-3-21** | 0 |
| **2016-1-22** | 13 | **2016-2-21** | 0 | **2016-3-22** | 0 |
| **2016-1-23** | 13 | **2016-2-22** | 0 | **2016-3-23** | 1 |
| **2016-1-24** | 10 | **2016-2-23** | 0 | **2016-3-24** | 1 |
| **2016-1-25** | 11 | **2016-2-24** | 0 | **2016-3-25** | 1 |
| **2016-1-26** | 16 | **2016-2-25** | 0 | **2016-3-26** | 2 |
| **2016-1-27** | 14 | **2016-2-26** | 0 | **2016-3-27** | 2 |
| **2016-1-28** | 8 | **2016-2-27** | 0 | **2016-3-28** | 2 |
| **2016-1-29** | 9 | **2016-2-28** | 0 | **2016-3-29** | 4 |
| **2016-1-30** | 9 | **2016-2-29** | 0 | **2016-3-30** | 4 |
|  |  |  |  | **2016-3-31** | 3 |

7 模型的检验

7.1模型一的检验

易证是收敛的，所以当考虑园内游客总数不变的情况下，经过若干次的疏导，园内每一个节点的拥挤程度将会收敛到一个稳定的状况，所以该模型是可行的。

7.2模型二的检验

在进行季节因素的Fourier拟合中，在第一次拟合得到以下数据：

Coefficients (with 95% confidence bounds):



括号内为置信水平为95%的置信区间。显然置信区间过大，所以固定以及，对剩下的，得出



同样为95%的置信水平，第二次拟合的置信区间就要小的多。且此时模型的R-square已达到0.8354，很接近1了，所以该模型是合理的。

8 模型评价

第一个模型做出了许多的假设，且在模型的解决中被多次简化计算，模型计算误差相对也就增大。在通过C++模拟出一个初始状态后，带入已知数据加以验证可发现，当对模型精度要求较高时，lingo将会报错得不到合适的解，所以仍需对模型进行简化并且不以减少模型精度为代价。

第二个模型由于只由一年多一个月为基础建立，所以该模型不能很好的预计长期趋势。然而长期趋势往往可以决定每一年的总体情况，若有多几年的数据，则对2016年的数据预测会有更大的帮助。且在计算季节因素时也应该才用平均移动法计算平均值，这样可以有效的减小拟合和方差，即SSE值。

9模型推广

在日益拥堵的城市交通网络中，拥挤状况随时可见。通过监控每个路口节点拥堵状况给出整个路网分析，能为司机们提供更加有效的路径选择以避免拥堵。尤其是当城市中举办有大型活动如展览会等，合理有效的疏导分流可以尽可能减少拥挤的情况。

通过时间序列模型能描述的对象不胜枚举。例如，逐年的粮食产量可以用时间序列模型进行预测，精准的预测可以使得粮食种植户能够及时做好准备，找到会限制产量的因素解决之并提高产量。

10 参考文献

[1]易丹辉《数据分析与EVIEWS应用》 中国人民大学出版社 10/2008

[2]王振龙《时间序列分析》 中国统计出版社 02/2000

11 附录

附录一 建立无向图表

#include<iostream>

#include<cstdio>

#include<cstdlib>

#include<vector>

#include<ctime>

#include <fstream>

using namespace std;

#define MaxVertexNum 50

typedef struct node \*EdgeNode;

typedef struct vnode \*VertexNode;

typedef struct graph \*ALGraph;

struct node

{ //边表节点

int adjvex; //邻接点域

int length;

EdgeNode next; //链域

};

struct vnode

{ //顶点表节点

int vertex; //顶点域

int now\_people;

int cost\_time;

int perR\_people;

int rest\_time;

EdgeNode firstedge; //边表头指针

};

struct graph

{

struct vnode \*adjlist;

int n; //图中当前顶点数

int e; //图中当前边数

};

void CreatSides(ALGraph G,int i,int j,int length)

{

EdgeNode s;

s = new struct node; //生成边表节点

s->length=length;

s->adjvex = j;

s->next = G->adjlist[i].firstedge;

G->adjlist[i].firstedge = s; //将新节点\*s插入顶点Vi的边表头部

s = new struct node;

s->length=length;

s->adjvex = i;

s->next = G->adjlist[j].firstedge;

G->adjlist[j].firstedge = s;

}

void CreatNode(ALGraph G,int i,int a,int b)

{

G->adjlist[i].perR\_people=a;

G->adjlist[i].cost\_time=b;

G->adjlist[i].now\_people=rand()%(1500-200+1)+200; //生成[a,b]之间的随机整数的方法

}

void CreatALGraph(ALGraph G)

{

int i;

G->n = 11; //顶点数和边数

G->e = 16;

for (i = 0; i < G->n; i++) //建立顶点表

{

G->adjlist[i].vertex = i; //读入顶点信息 //顶点编号

G->adjlist[i].firstedge = NULL; //边表头指针置为空

}

CreatNode(G,0,1,1000000);

CreatNode(G,1,400,33\*60);

CreatNode(G,2,30,75);

CreatNode(G,3,50,150);

CreatNode(G,4,30,150);

CreatNode(G,5,100,300);

CreatNode(G,6,50,150);

CreatNode(G,7,30,120);

CreatNode(G,8,30,90);

CreatNode(G,9,20,90);

CreatNode(G,10,50,120);

CreatSides(G,0,1,300);

CreatSides(G,0,2,400);

CreatSides(G,1,2,300);

CreatSides(G,1,5,350);

CreatSides(G,1,10,250);

CreatSides(G,2,3,300);

CreatSides(G,2,5,350);

CreatSides(G,3,4,450);

CreatSides(G,3,5,500);

CreatSides(G,4,6,500);

CreatSides(G,5,7,550);

CreatSides(G,5,9,450);

CreatSides(G,6,7,650);

CreatSides(G,7,8,400);

CreatSides(G,8,9,450);

CreatSides(G,9,10,350);

G->adjlist[0].rest\_time=0;

for (i = 1; i < G->n; i++) //建立顶点表

{

G->adjlist[i].rest\_time = rand()%G->adjlist[i].cost\_time+1;

}

}

附录二 经过简化的lingo规划算法语言

model:

data:

na=1980;nb=75;nc=150;nd=150;ne=300;nf=150;ng=120;nh=90;ni=90;nj=120;no=10000;

ta=500;tb=54;tc=49;td=30;te=132;tf=136;tg=78;th=53;ti=43;tj=30;to=10000;

enddata

min=taa+tbb+tcc+tdd+tee+tff+tgg+thh+tii+tjj;

taa=xaa\*na+ta;

tbb=xbb\*nb+tb;

tcc=xcc\*nc+tc;

tdd=xdd\*nd+td;

tee=xee\*ne+te;

tff=xff\*nf+tf;

tgg=xgg\*ng+tg;

thh=xhh\*nh+th;

tii=xii\*ni+ti;

tjj=xjj\*nj+tj;

xii=(xi)/mi;

xjj=(xj)/mj;

xaa=(xa)/ma;

xbb=(xb)/mb;

xcc=(xc)/mc;

xdd=(xd)/md;

xee=(xe)/me;

xff=(xf)/mf;

xgg=(xg)/mg;

xhh=(xh)/mh;

xoo=(xo)/m0;

@floor(xii)\*ni+ti-(xih)\*nh-th>0;

@floor(xii)\*ni+ti-(xie)\*ne-te>0;

@floor(xii)\*ni+ti-(xij)\*nj-tj>0;

@floor(xjj)\*nj+tj-(xji)\*ni-ti>0;

@floor(xjj)\*nj+tj-(xja)\*na-ta>0;

@floor(xaa)\*na+ta-(xao)\*no-to>0;

@floor(xaa)\*na+ta-(xab)\*nb-tb>0;

@floor(xaa)\*na+ta-(xae)\*ne-te>0;

@floor(xaa)\*na+ta-(xaj)\*nj-tj>0;

@floor(xoo)\*no+to-(xoa)\*na-ta>0;

@floor(xoo)\*no+to-(xob)\*nb-tb>0;

@floor(xbb)\*nb+tb-(xba)\*na-ta>0;

@floor(xbb)\*nb+tb-(xbo)\*no-to>0;

@floor(xbb)\*nb+tb-(xbe)\*ne-te>0;

@floor(xbb)\*nb+tb-(xbc)\*nc-tc>0;

@floor(xcc)\*nc+tc-(xcb)\*nb-tb>0;

@floor(xcc)\*nc+tc-(xce)\*ne-te>0;

@floor(xcc)\*nc+tc-(xcd)\*nd-td>0;

@floor(xdd)\*nd+td-(xdc)\*nc-tc>0;

@floor(xdd)\*nd+td-(xdf)\*nf-tf>0;

@floor(xee)\*ne+te-(xea)\*na-ta>0;

@floor(xee)\*ne+te-(xeb)\*nb-tb>0;

@floor(xee)\*ne+te-(xec)\*nc-tc>0;

@floor(xee)\*ne+te-(xeg)\*ng-tg>0;

@floor(xee)\*ne+te-(xei)\*ni-ti>0;

@floor(xff)\*nf+tf-(xfd)\*nd-td>0;

@floor(xff)\*nf+tf-(xfg)\*ng-tg>0;

@floor(xgg)\*ng+tg-(xgf)\*nf-tf>0;

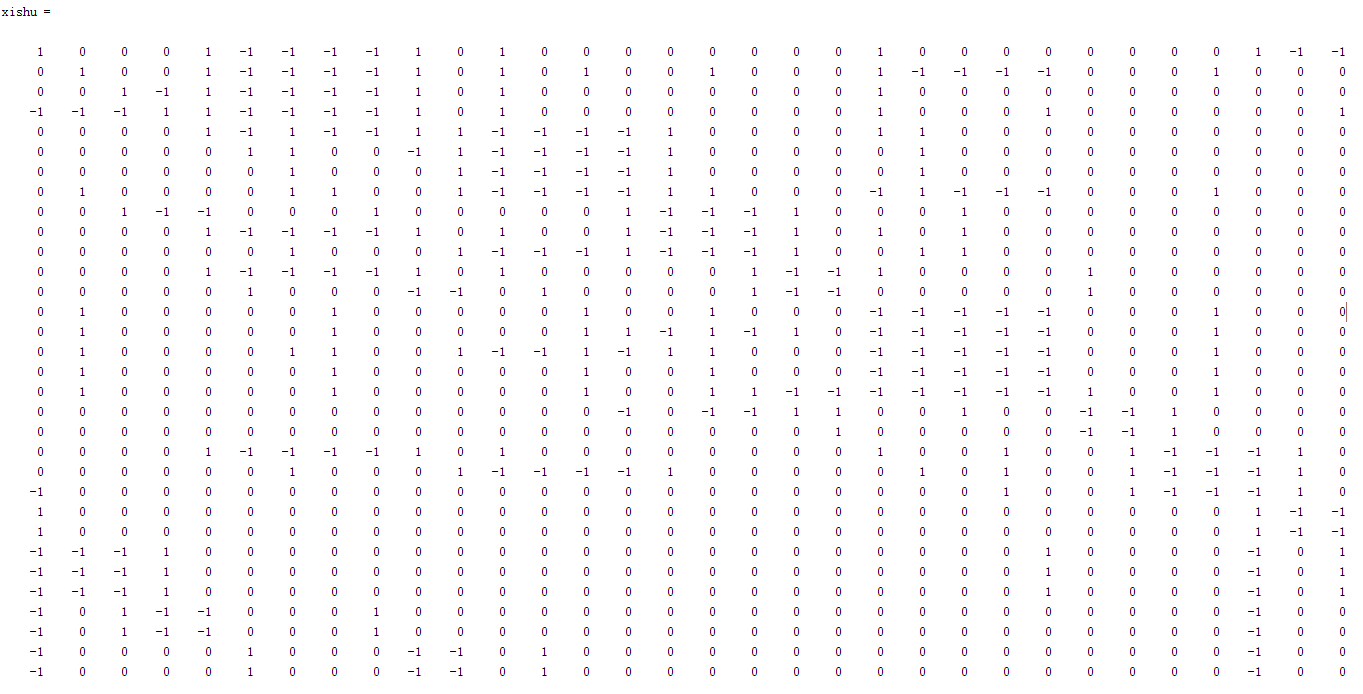
@floor(xgg)\*ng+tg-(xge)\*ne-te>0;

@floor(xgg)\*ng+tg-(xgh)\*nh-th>0;

@floor(xhh)\*nh+th-(xhg)\*ng-tg>0;

@floor(xhh)\*nh+th-(xhi)\*ni-ti>0;

附录三 系数矩阵



附录四 模型二的matlab代码

clc,clear

load lidian; load daodian; load fangjianshu;load fangjia;load xiuzheng;

x=length(lidian);

n=max(lidian)-min(daodian);

t=zeros(n,2);

t(:,1)=[0:n-1]'+ones(n,1)\*min(daodian);

%datestr(t(1:n,1),'yyyy-mm-dd ddd')

total=lidian-daodian;

for i=1:x

for j=1:total(x)

t(daodian(i)-min(daodian)+j,2)=t(daodian(i)-min(daodian)+j,2)+fangjianshu(x);

end

end

%xlswrite('result1.xls',t)

t(:,2)=xiuzheng;

[nummax,xmax]=max(t(:,2));

figure(1)

plot(t(:,1),t(:,2),'y')

hold on,grid on

xlabel('日期')

ylabel('房间数')

title('2015年该酒店每天预订房间数','FontSize',16,'color','b')

text(t(xmax,1),nummax,'\leftarrow 最大值为101','color','y','FontSize',14)

b=736059-56-14-14;o=736324+14+14;

b1=736059+196+14;o1=736324-28;

set(gca,'XTick',b:28:o);set(gca,'XTickLabel',{'Sun'});datetick('x','mmm')

begin1=b1-t(1,1)+1;over1=o1-t(1,1)+1;l91=over1-begin1+1;

begin=b-t(1,1)+1;over=o-t(1,1)+1;l9=over-begin+1;%全部时间取整周

datestr(b,'yyyy-mm-dd ddd')

datestr(o,'yyyy-mm-dd ddd')

datestr(b1,'yyyy-mm-dd ddd');

datestr(o1,'yyyy-mm-dd ddd');

t9(:,1)=t(begin1:over1,1);

t9(:,2)=t(begin1:over1,2);

t9x=t9(:,1);

t9y=t9(:,2);

%plot(t9x,t9y,'m')

mavg=zeros(l91-6,1);%平均移动数

for h1=1:l91-6

mavg(h1)=(t9(h1,2)+t9(h1+1,2)+t9(h1+2,2)+t9(h1+3,2)+t9(h1+4,2)+t9(h1+5,2)+t9(h1+6,2))/7;

end

cmavg=zeros(l91-7,1);%中心化的平均移动数,不用

for h2=1:l91-7

cmavg(h2)=(mavg(h2)+mavg(h2+1))/2;

end

season=zeros(l91-7,1);%季节性不规则数

season=t9(4:l91-7+3,2)./mavg(1:l91-7);

jjindex=zeros(7,1); %季节指数

i10=4;%season第一个值是周三的

for i9=1:l91-7

jjindex(i10)=jjindex(i10)+season(i9);

i10=i10+1;

if i10>7

i10=1;

end

end

jjindex=jjindex./(l91/7-1)

t8=t;

for i8=1:n

xingqi=mod(t8(i8,1),7);

datestr(t8(i8,1),'yyyy-mm-dd ddd');

if xingqi<2

t8(i8,2)=t8(i8,2)/jjindex(xingqi+6);

else

t8(i8,2)=t8(i8,2)/jjindex(xingqi-1);

end

end

t8x=t8(begin:over,1);t8y=t8(begin:over,2);

%plot(t8(:,1),t8(:,2),'g')

tvalid(:,1)=t8(begin:over,1);tvalidx=tvalid(:,1);

tvalid(:,2)=t8(begin:over,2);tvalidy=tvalid(:,2);

p1=polyfit(tvalid(:,1),tvalid(:,2),1); %长期趋势T(整年的无法预估？)

x331=t(1,1):datenum('2016-3-31');

y331=polyval(p1,x331); %延长至3-311

weekavg=zeros(l91/7,1);

for he=1:l9/7

weekavg(he)=tvalid((he-1)\*7+1,2)+tvalid((he-1)\*7+2,2)+tvalid((he-1)\*7+3,2)+tvalid((he-1)\*7+4,2)+tvalid((he-1)\*7+5,2)+tvalid((he-1)\*7+6,2)+tvalid((he-1)\*7+7,2);

end

lw=length(weekavg);

weekavg=weekavg/7;

week=([1:l9/7]-1).\*7+3+tvalid(1,1);la=length(week);

%plot(week,weekavg);

a0 = 32.35;

a1 =-11 ;

a2 = 6.582 ;

b1 = 17.8 ;

b2 = 12.47 ;

w = 0.01592 ;

x=x331;

f = a0 + a1\*cos(x\*w) + b1\*sin(x\*w) + a2\*cos(2\*x\*w) + b2\*sin(2\*x\*w);

t6(:,1)=x;t6(:,2)=f;

%plot(x,ceil(t6(:,2)),'b');

for i6=1:length(x)

xingqi=mod(t6(i6,1),7);

datestr(t6(i6,1),'yyyy-mm-dd ddd');

if xingqi<2

t6(i6,2)=t6(i6,2)\*jjindex(xingqi+6);

else

t6(i6,2)=t6(i6,2)\*jjindex(xingqi-1);

end

end

shujia0=datenum('2015-12-31');shujia1=datenum('2015-12-31');lshujia=shujia1-shujia0;

ave0=0;

for sj=1:n

if (t(sj,1)>=shujia0)&&(t(sj,1)<=shujia1)

ave0=t(sj,2)+ave0;

end

end

ave1=0;

for sj=1:n

if (t6(sj,1)>=shujia0)&&(t6(sj,1)<=shujia1)

ave1=t6(sj,2)+ave1;

end

end

sjyx=ave0/ave1

hanjia0=datenum('2016-1-1')-t6(1,1)+1;hanjia1=datenum('2016-1-3')-t6(1,1)+1;

t6(hanjia0:hanjia1,2)=t6(hanjia0:hanjia1,2)\*sjyx;

l6=length(t6);

for i6=1:l6

if t6(i6,2)<0

t6(i6,2)=0;

end

end

plot(x,ceil(t6(:,2)),'m');

legend('原始数据','拟合数据')

result0=datenum('2016-1-1')-t6(1,1)+1;result1=datenum('2016-3-31')-t6(1,1)+1;

%xlswrite('result.xls',ceil(t6(result0:result1,:)))