

“拍照赚钱”的定价分析

摘要

随着互联网技术的发展，“拍照赚钱”已经成为时下一种热门的互联网自助模式。本文通过建立合理的数学模型，对拍照任务定价问题进行了分析。

针对问题一，我们通过分析发现对所有的任务点用 ISODATA 算法进行聚类分析，可以将任务点分为四类，将这些点以及得到的聚类中心通过开源的地图网站反应到实际地图中，发现这四个聚类中心与“广州、佛山、东莞、深圳”这四个城市中心对应，并且这四类大致与城市相符合。通过对价格与任务点到中心的距离分析发现离市中心越近的任务点，其价格也就越低。为了寻求这些任务的定价规律，我们利用回归分析的方法得到定价与其坐标的关系为： $s = 13.9373222273683 * d_0 + 66.7401303959352$ 。采用前 417 作为训练样本，后 417 个点来检验，我们发现此方程拟合情况基本较好，但是对于定价超过 80 元的任务有较大误差。我们取出这些任务点进行分析，得出这些任务点定价更高是因为特定城市特定任务的要求和难度可能存在较大的特殊性。在后续的过程中，我们将针对具体点的要求难度等因素通过改变加权来进行修正。关于任务的完成情况，通过分析任务定价与完成率之间的关系，我们得出定价越高的任务，完成率越高的结论。

针对问题二，我们综合考虑任务情况与会员分布的相互影响，即任务对于周围的会员存在着吸引力 $c_i = f(s, d)$ ，它与任务价格正相关，与距离任务距离负相关。将会员的分布与会员等级通过位势函数法抽象为空间内分布的用户活跃度 $a_i = g(x, y)$ ，在任务点的邻域内对二者的乘机积分 $\iint c_i a_i dx dy$ 即可得到任务被完成的概率 p_i ，从而找到定价价格与概率 p_i 之间的函数关系。进而得到公司的收益期望与定价的关系 $E(R_i) = p(L - s)$ ，是公司收益期望最大的定价即是我们重新确定的定价。

针对问题三，我们引入了行动力这一概念，会员产生行动力，完成任务以及移动到任务点均需消耗行动力，并将模型的主要着眼点聚焦于合理组织分配行动力。我们把行

动力效率 $\omega = \frac{\text{做任务行动力}}{\text{移动行动力} + \text{做任务行动力}}$ 作为标准，通过分析任务被不同用户 j 完成的概率 P_j ，

控制移动的行动力消耗，从而控制 ω 。据此我们分析得到，应区别对待不同位置的任务。临近市中心、居住区的任务，应少打包或不打包，反之，应较打大的包，依据这个我们建立了打包算法。在四个城市中心分类的基础上，依据每一个点到中心的距离将每一类分带，分别设置包内任务数。然后包内采用最近邻法则，不断合并剩余点中最近两点，直到完成聚类。

针对问题四，我们通过比较最新系项目任务图和用户分布图行动力情况图，发现东莞为的行动力饱和区，广州为行动力消耗饱和区。我们综合考虑各用户、各任务的相互影响，计算并分析各任务和各用户间匹配性的矩阵，发现一些行动力消耗饱和区的任务更应由行动力饱和城市的用户完成而不是本城市用户。我们将这些任务视为行动力饱和城市的市郊任务计算，并据此利用问题二和问题三建立的模型得出各包的分类与定价情况。

关键词： 聚类分析 回归分析 位势函数法 最近邻法则

目录

一、问题重述	3
1.1 背景资料	3
1.2 需要解决的问题	3
二、问题分析	3
2.1 问题一的分析	3
2.2 问题二的分析	4
2.3 问题三的分析	4
2.4 问题四的分析	4
三、模型假设	4
四、符号说明	4
五、模型的建立与求解	5
5.1 建立项目定价规律模型	5
5.1.1 项目定价规律的可视化分析	5
5.1.2 任务未完成的原因分析	8
5.2 考虑会员等级带来影响的定价模型	10
5.3 将任务打包发布的定价模型	13
5.3.1 引入行动力建立打包数的函数关系	13
5.3.2 打包算法的实现及结果	15
5.3.3 打包算法对定价模型的影响	17
5.4 对于新项目的定价方案	19
六、模型的评价	22
6.1、模型的优点	22
6.2、模型的缺点	23
七、参考文献	23
附录	23

一、问题重述

1.1 背景资料

近年来，随着互联网技术的高速发展，网络已经融入了我们生活的方方面面。

“拍照赚钱”是移动互联网下的一种自助式服务模式，有多家公司依托移动互联网建立了服务平台，如“拍拍赚”、“蚂蚁威客”、“美团拍客”等等。用户下载 APP，注册成为 APP 的会员，然后从 APP 上领取需要拍照的任务（比如上超市去检查某种商品的上架情况），赚取 APP 对任务所标定的酬金。这种基于移动互联网的自助式劳务众包平台，为企业提供各种商业检查和信息搜集，相比传统的市场调查方式可以大大节省调查成本，而且有效地保证了调查数据真实性，缩短了调查的周期。因此 APP 成为该平台运行的核心，而 APP 中的任务定价又是其核心要素。如果定价不合理，有的任务就会无人问津，而导致商品检查的失败。

1.2 需要解决的问题

我们通过分析相关数据，运用数学思想，建立数学模型来研究拍照软件定价中的下列问题：

- （1）研究项目的定价规律，并分析任务未完成的原因；
- （2）为项目设计新的定价方案，并与原方案进行比较，分析优劣之处；
- （3）在实际情况下，多个任务可能因为位置比较集中，导致用户会争相选择，我们考虑将这些任务联合起来打包发布并修改前面的定价模型，分析最终的任务完成情况；
- （4）对一些新项目给出我们的定价方案，并评价该方案的实施效果。

二、问题分析

2.1 问题一的分析

问题一要求我们根据已有提供的数据，研究得出项目的定价规律，并分析得出任务未完成的原因。

在分析定价规律时，考虑到题目中给出了每个项目的经纬度坐标，我们在地图中用热力图标识出这些点的分布概况，并按照大体的价格分布以及任务的完成情况进行分析从而得出了项目的定价规律。

2.2 问题二的分析

在问题二改进定价方案时，我们考虑到任务是否完成主要受到两个方面因素影响，一是任务本身对于会员的吸引力，主要由任务位置以及任务定价决定，二是任务周围会员的数量以及活跃度，我们综合分析考虑这些因素，得出定价与公司收益之间的关系，以确定每个任务的合理定价。

2.3 问题三的分析

在考虑对任务进行打包时，我们引入了会员行动力这一概念，综合分析这一概念与打包函数之间的关系，然后建立了打包函数求解的算法，在得到如何打包的分类后，我们可以建立新的任务定价与收益之间的关系，以此确立打包对函数模型带来的影响。

2.4 问题四的分析

我们分析发现，对于新项目而言，完成项目需要的行动力和当前用户的行动力分布是不均匀的。因此我们通过分析任务的吸引力与用户的完成能力，确定了对于不同任务完成能力最大的那个用户所在的城市，以此为依据，利用二、三问的模型进行求解即可。

三、模型假设

- (1) 用户的情况不同。有的用户比较忙，在工作生活之余做点任务，是零散行动力；有的用户能抽出大量时间做任务，是整块的行动力。
- (2) 短时间内，用户的数量和成分不变，即可以使用的行动力不变。
- (3) 短时间内，任务的价格和位置不会进行调整。
- (4) 每个人的精力都是有限的，不可能一直在完成任务。
- (5) 假设任务完成的难度基本一致，即会员完成一个任务所耗时间相同。

四、符号说明

符号	说明
d	两坐标点之间的距离
x	坐标点经度坐标
y	坐标点纬度坐标
s	任务点定价
c	吸引力指标函数
m	系数
a	地区活跃度指标
k	会员活跃度指标

b	会员信誉积分
r	划分区域的半径
G	门限
P	任务被完成的概率
R	公司获得的收益
v	行动力
Ω	行动力效率
Acm	移动消耗的行动力
Acw	工作消耗的行动力
n	打包后的总包含有原先小包的个数
e	打包后包内每个任务的平均定价
f	会员完成任务的能力

五、模型的建立与求解

5.1 建立项目定价规律模型

5.1.1 项目定价规律的可视化分析

本文考虑到，项目的定价规律主要受到项目的地理位置、当地人的收入预期、当地社会的社会人文特征的影响。我们首先利用了地图无忧软件按照价格区间进行分类，绘制了如下任务点的空间分布图，其中红色表示价格区间为 65-67 元，绿色表示价格区间为 67-70 元，灰色表示价格区间为 70-75 元，紫色表示价格区间为 80-85 元，在地图上标识出这些点后，我们通过观察可以看出这些点围绕在广州、佛山、东莞、深圳四市市中心周围，并且靠近市中心价格较低，远离市中心价格较高。

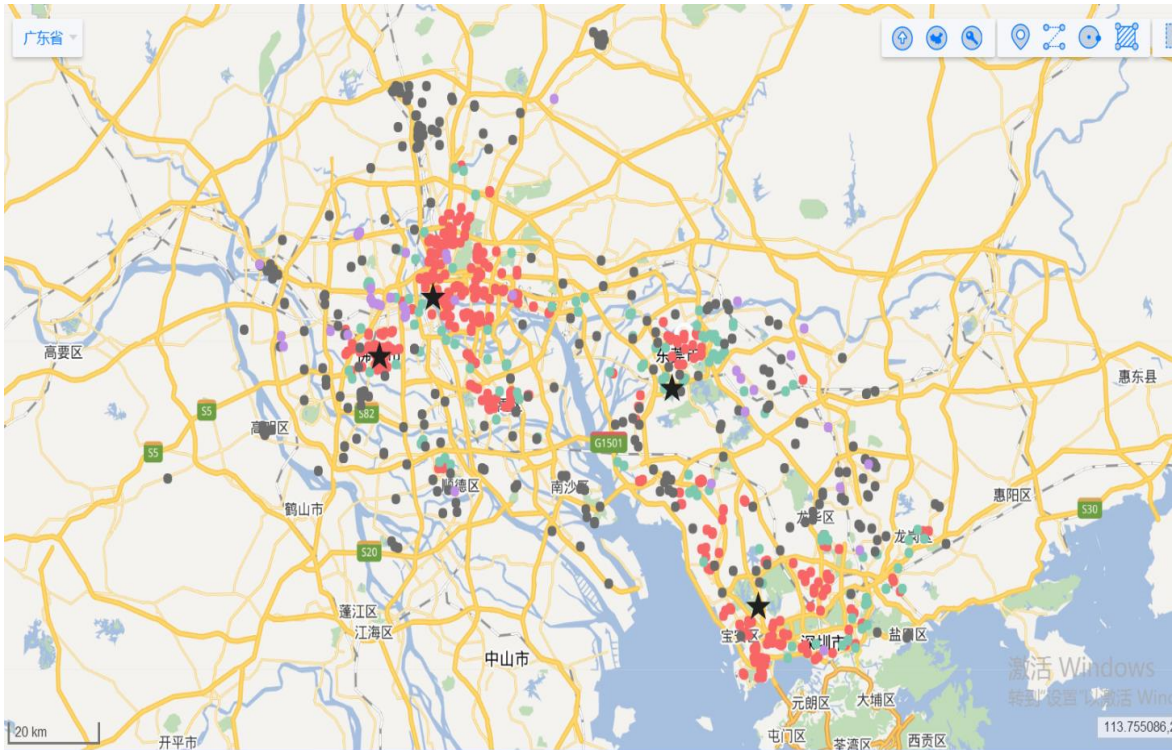


图 1 任务点按照价格区间分类的空间分布图

因此我们利用 ISODATA 算法对于这些任务点数据进行聚类分析，我们在每轮迭代过程中，样本进行重新调整类别之后计算类内及类间有关参数，并和设定的门限比较，确定是两类合并为一类还是一类分裂为两类，不断地“自组织”，已达到在各参数满足设计要求条件下，使各模式到其类心的距离平方和最小。利用 MATLAB 对此算法进行求解我们得到了四类的聚类中心为：

表 1 四类聚类中心的坐标

第 1 类聚类中心为	22.6107	113.927
第 2 类聚类中心为	23.124	113.2424
第 3 类聚类中心为	22.9732	113.7467
第 4 类聚类中心为	23.02717	113.1287

我们将求出的上面四点在图一中用五角星标出，发现聚类情况与我们预先的估计较为吻合，因此我们可以得出以下结论：任务的定价与其所处的地理位置有关，其大小与其到市中心的距离大体符合负相关关系，因为真实情况下存在着一定的随机性，所以我们使用线性回归分析来简化模型，得到价格与位置的关系，据此我们建立了我们的任务定价模型：

首先我们需要求出任务点分别到四个聚类中心的距离：

$$d_i = \sqrt{(x_0 - x_i)^2 + (y_0 - y_i)^2}$$

然后即将四个距离值 d_1, d_2, d_3, d_4 进行比较，取出四个中的最小值：

$$d_0 = \min\{d_1, d_2, d_3, d_4\}$$

然后我们利用回归分析以前面 417 组数据作为训练样本，求出任务点定价 s 与坐标之间的关系：

$$s = 13.9373222273683 * d_0 + 66.7401303959352$$

并利用后面 417 组数据作为检验，截取部分数据如下表：

表 2 线性回归模型的检验结果

任务号码	任务 gps 纬度	任务 gps 经度	任务标价	试验标价	误差
A0509	22.81963599	113.8037291	65	67.22613	-2.22613
A0538	22.84339687	113.2557574	66	68.07259	-2.07259
A0464	22.52712849	114.0539088	65.5	67.05758	-1.55758
A0463	23.09785202	113.3057915	65	65.87237	-0.87237
A0703	22.78108249	114.0574014	69	67.94743	1.052565
A0418	23.54196985	113.5979552	75	72.69763	2.302368
A0687	23.10425851	113.8644239	72	67.40177	4.598233
A0711	22.84158065	113.9837534	75	68.27716	6.722842
A0829	23.17903001	112.8761925	80	69.08648	10.91352
A0450	22.57872798	114.4936096	85	72.96441	12.03559

发现拟合情况大部分较好，误差在 2.5 元以内，但是对于定价超过 80 元的一些情况，我们发现误差达到了十元以上，拟合情况在这种状况下不尽如人意，我们分析认为对于这种情况除了存在距离市中心距离长短的影响外，必然还存在着其他因素的作用。

因此我们将这些定价超过 80 元的点利用地图找到了他们各自的具体位置，情况如下表：

表 3 定价超过 80 元的任务所在地

任务号码	地址
A0450	广东省深圳市龙岗区南湾街道南新社区东北方向约 1.13 公里
A0460	广东省深圳市龙岗区布吉街道凤凰社区东北方向
A0739	中国广东省佛山市南海区
A0742	广东省东莞市寮步镇寮步社区东南方向约 1.19 公里
A0749	广东省东莞市大朗镇大朗社区西北方向
A0750	广东省东莞市横沥镇村头村西南方向
A0751	广东省东莞市石龙镇王屋洲村东北方向约 1.15 公里
A0760	广东省东莞市塘厦镇石潭埔社区西南方向约 1.02 公里
A0765	广东省东莞市寮步镇良边村西北方向
A0773	广东省东莞市企石镇铁炉坑村东北方向
A0775	广东省佛山市南海区大沥镇沥北社区西北方向
A0777	广东省佛山市南海区丹灶镇东升社区东北方向
A0788	广东省佛山市南海区里水镇草场社区西北方向约 1.13 公里
A0789	广东省佛山市南海区河东中心路
A0790	广东省佛山市南海区永安大道东 37 号
A0791	广东省佛山市南海区桂丹路辅路
A0793	广东省佛山市南海区广佛路 125 号
A0794	广东省佛山市南海区永安大道南三巷
A0796	广东省佛山市南海区兴隆路，528231
A0797	广东省佛山市顺德区民安路
A0798	广东省佛山市南海区荷桂路，528216
A0800	广东省佛山市南海区荔新一路南四街，528231
A0808	广东省东莞市上元路 20 号
A0809	广东省佛山市南海区兴隆路，528231
A0830	广东省佛山市南海区涌边大街，528244
A0833	广东省东莞市德政东路 7 号，511700
A0835	广东省佛山市南海区南湾南路，528231

项目的定价会根据项目内容完成的难易程度存在着差异，我们知道当项目的位置处于相对偏远的地区或者交通不易到达的地方时，由于存在调查的困难性，需要定更高的价格以提高这些任务的完成率。

通过此表，我们发现这些定价超过 80 元的任务大多集中在东莞市与佛山市的偏僻郊区，我们结合现实因素加以考虑，认为出现这样的情况主要有两方面的原因，一是这些任务所在地较偏僻，远离市中心，完成难度较大，需要更高的价格刺激会员去完成任务；二是可能由于任务的发布者迫切需要了解东莞、佛山两市的相关情况，以提高价格的手段来促使任务完成的周期缩短。

5.1.2 任务未完成的原因分析

我们利用百度地图开放平台将任务点按照价格区间以及完成的情况进行了地图的绘制，采用了密度图的形式进行了反应，通过对于下图的分析

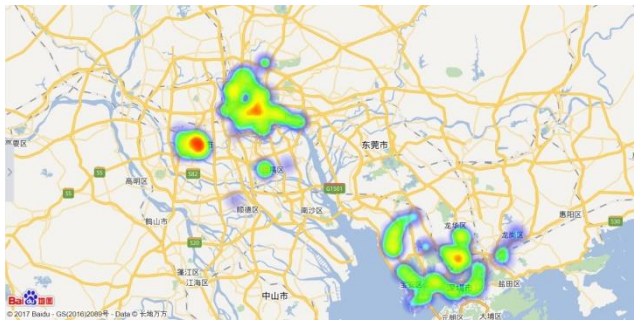


图 2 定价 67 元以下未完成的密度图

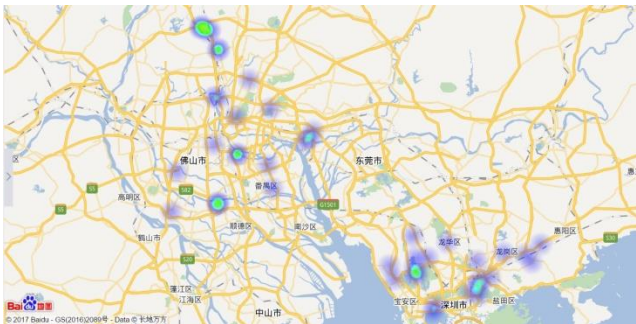


图 3 定价 67-70 元未完成的密度图

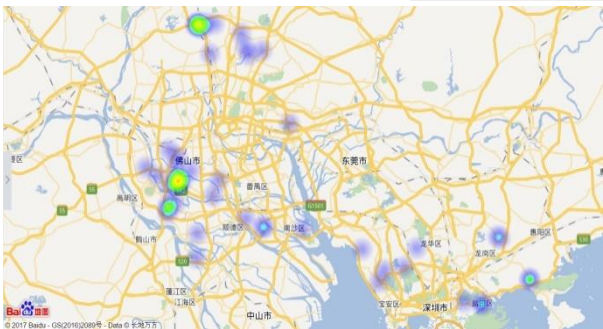


图 4 定价 70-76 元未完成的密度图

我们可以发现未完成的任務主要集中在廣州市以及深圳市，而東莞市的任務則大多都被很好地完成了。我們通過查閱三處城市的社会经济特征，发现東莞市的经济发展状况比廣州市和深圳市的差，并且人口以年轻人以及女性偏多。年轻人接受新鲜事物快，并且一些在校学生以及全职太太更倾向于通过完成任务的方式获得一些报酬，所以任务完成率更高。



图 5 定价 67 元以下完成的密度图

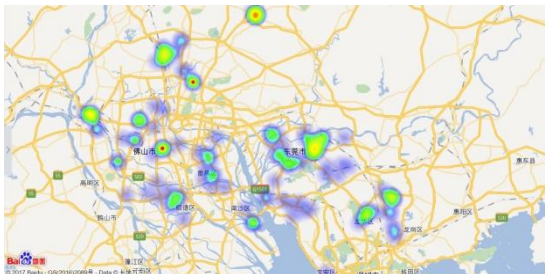


图 6 定价 67-70 元完成的密度图

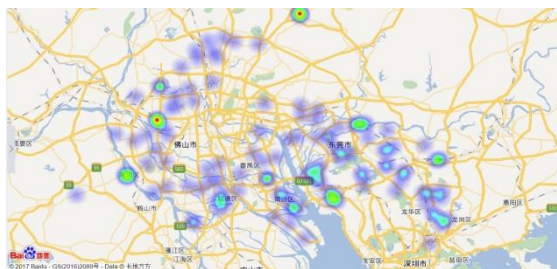


图 7 定价 70-76 元完成的密度图

通过上图我们发现定价高的任务完成的情况较好，在城市中心的任务完成情况也较好。在市中心是因为交通便利，会员容易到达，也因为市中心的人员更加密集，周围分布的会员多，任务更易被完成。而针对任务定价带来的影响，我们进行了如下进一步分析：

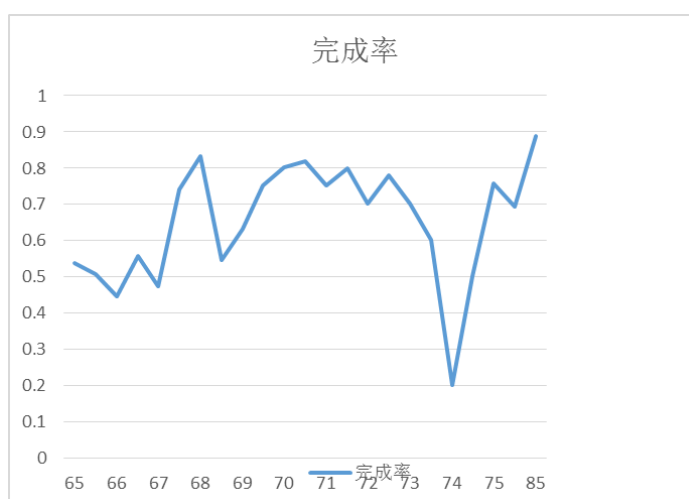


图 8 任务每种价格的总数与完成的个数

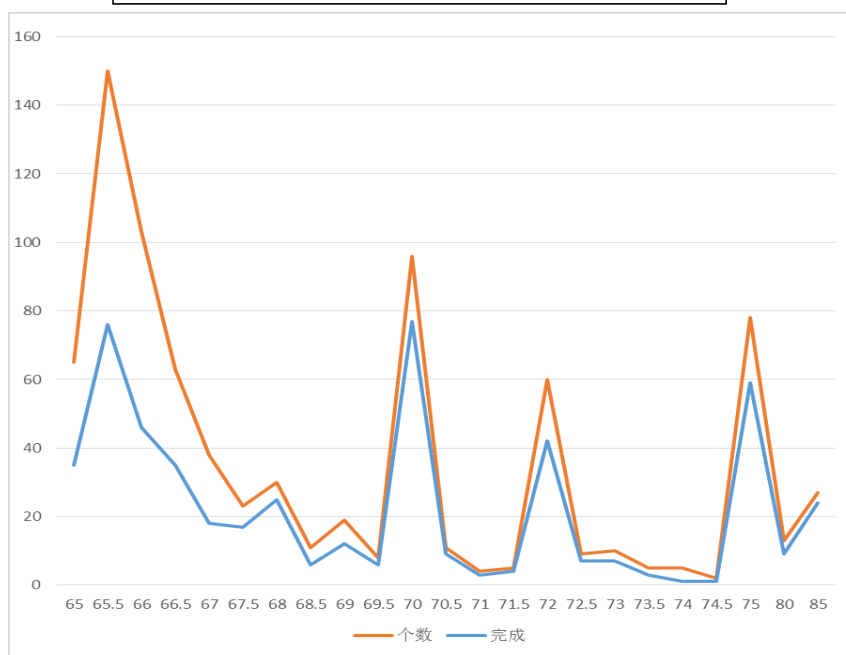


图 9 任务每种价格完成率

我们结合上面的两张图综合分析可以看出，在价格为 74 元时完成率很低是由于价格在 74 元的任务总数较少，导致偶然性带来的波动较大，可以舍弃这一组数据。总的来看，我们发现在任务价格较低时，任务的完成率也会随之降低，而在任务价格较高时，任务完成情况较好。这是因为人们在有所选择的情况下更加倾向于去完成获得报酬高的任务来给自己带来更大的收益。

5.2 考虑会员等级带来影响的定价模型

我们分析得到，会员对于一个发布的任务存在着一个完成任务的意愿，我们用任务对会员的吸引力指标函数 c 来表示这个意愿，分析实际情况我们容易得知当一个任务的定价 s 越高并且任务的距离 d 距会员越近时，会员完成这个任务意愿就会越强，即吸引力指标函数 c 与 s 正相关，与 d 负相关且根据实际我们认为意愿应呈指数形式，得到：

$$c_i = f(s, d_i) = m_1 e^{-\frac{m_2 d_i^2}{s}}$$

我们在建立模型时，考虑到会员不是静止不变的，他们总是处在不断的运动之中，而这时一些道路路况好，容易到达以及繁华的地方由于人流量大，便会有更大的几率有更多的会员从这里经过，相应地这些地方的任务更容易被完成，因此我们用 a 来定义一个坐标处的活跃度函数，而它的活跃度大小受到其周围区域内会员的数量以及会员等级的影响，会员等级越高代表其完成任务更加积极主动，会给其周围区域带来更高活跃度。因此我们用 k 来表示会员自身的活跃度，它由会员信誉积分 b 决定，公式为：

$$k = \ln b$$

由此我们可以得到一个坐标的活跃度为：

$$a = g(x, y) = \sum_{i=1}^n k_i \exp\{m_3[(x - x_i)^2 - (y - y_i)^2]\}$$

其中 (x, y) 为地区的经纬度坐标； (x_i, y_i) 为第 i 个会员的经纬度坐标。

考虑到会员的分布很集中，全部考虑的话模型过于复杂且真实情况下存在偶然性，我们对模型进行简化，将单位区间内密集的会员进行合并，产生的新的经度 x_i' ，纬度 y_i' 以及会员活跃度 k_i' 定义为：

$$x_i' = \frac{\sum_{i=1}^j k_i x_i}{\sum_{i=1}^j k_i}$$

$$y_i' = \frac{\sum_{i=1}^j k_i y_i}{\sum_{i=1}^j k_i}$$

所以我们可以得到改变后的地区的活跃度为：

$$a = g(x, y) = \sum_{i=1}^n k_i' \exp\{m_3[(x - x_i')^2 - (y - y_i')^2]\}$$

利用 MATLAB 得出各个地区的活跃度如下图：

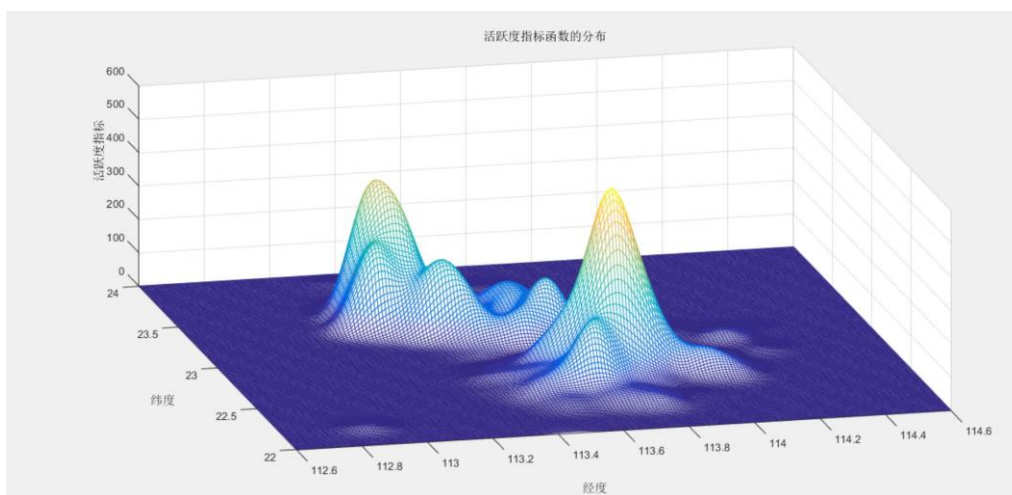


图 10 各地区活跃度指标函数分布图

所以任务被完成的可能性大小取决于任务吸引力 c 和任务所在地区活跃度 a ，据此我们得到任务被完成的可能性 p_i ：

$$p_i = \iint c_i a_i dx dy$$

对 p 进行运算，为了方便求解，将其划分为 13 块进行运算，划分如下：

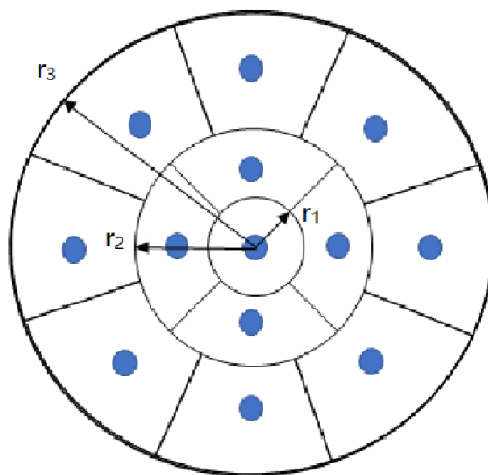


图 11 积分划分情况图

根据积分中值定理，我们可以得到：

$$\iint_{\Omega_i} f(s, d) g(x, y) dx dy = A_i \iint_{\Omega_i} f(s, d) dx dy (i = 1, 2 \dots 13)$$

并令

$$\tau_i = \iint_{\Omega_i} f(s, d) dx dy$$

所以 p 为：

$$p = \sum_{i=1}^{13} A_i \iint_{\Omega_i} f(s, d) dx dy = \sum_{i=1}^{13} A_i \tau_i$$

因为 A_i 符合正态分布

$$A_i \sim N(g(x_i, y_i), \sigma_i)$$

我们得到：

$$S^2(P) = \sum_{j=1}^{13} \tau_i^2 \sigma_i^2$$

因此我们得出当

$$\tau_1^2 = \tau_2^2 = \cdots = \tau_{13}^2$$

此时有

$$S^2(P) \rightarrow \min$$

设门限 G , $f(x)$ 小于 G 的省略, 得到 :

最大半径 $r_3 : f(r_3) = G$

由

$$m_1 e^{-\frac{m_2 r_3^2}{s}} = G$$

得到

$$r_3 = -\frac{s}{m_2} \ln \frac{G}{m_1}$$

又有

$$\tau_1 = \iint_{\Omega_1} f(x) dx dy = \int_0^{r_1} \int_{0 \leq \theta \leq 2\pi} m_1 e^{-\frac{m_2 r_1^2}{s}} r dr d\theta = 2\pi \frac{m_1 s}{m_2} (1 - \exp(-\frac{m_2 r_1^2}{s}))$$

$$\tau_2 = \iint_{\Omega_2} f(x) dx dy = \frac{\pi m_1 m_2}{2 s} \left(\exp\left(-\frac{m_2 r_1^2}{s}\right) - \exp\left(-\frac{m_2 r_2^2}{s}\right) \right) = \tau_3 = \tau_4 = \tau_5$$

$$\tau_6 = \iint_{\Omega_6} f(x) dx dy = \frac{\pi m_1 m_2}{4 s} \left(\exp\left(-\frac{m_2 r_2^2}{s}\right) - \exp\left(-\frac{m_2 r_3^2}{s}\right) \right) = \tau_7 = \tau_8 = \cdots = \tau_{13}$$

联立方程：

$$\tau_1 = \tau_2 = \tau_6$$

即可解出 r_1, r_2, r_3 。

即可得到 p 的值。

我们联系实际得知，软件公司给任务设立定价，当任务被完成时软件公司即可获得收益，因此我们可以得到该点任务对于公司的收益期望为：

$$E(R_i) = P(L - s) + (1 - p) \cdot 0 = p(L - s)$$

算出令 $E(R_i)$ 最大的任务定价 s 即是我们修改的更加有利的定价方案。

任取一点（23.00,113.80）为例，我们可以得到其任务价格与被完成概率的关系以及任务价格与收益期望的关系：

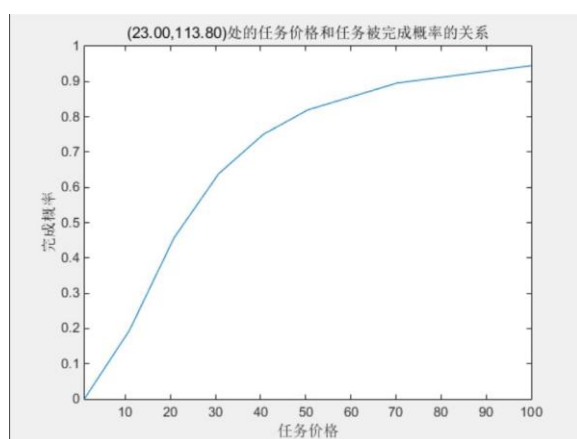


图 12 任务价格与被完成概率的关系

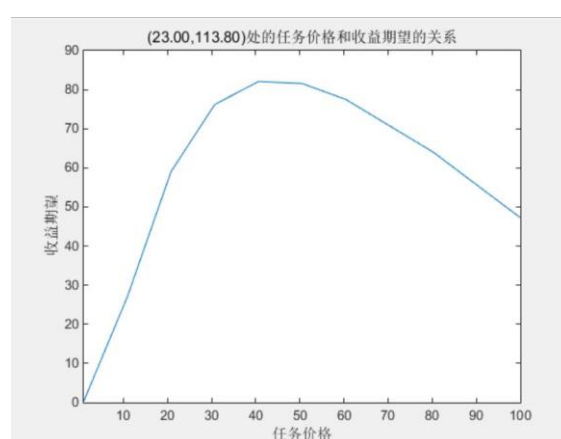


图 13 任务价格与收益期望的关系

据此，我们将（23.00,113.80）处的任务价格定为 45 元，可以使得公司获得更大的收益。

我们确定的新的定价相比原先的定价方案多考虑了会员的分布以及会员的等级所带来的影响，相比原先的方案此方案的价格定价更加的合理，可以使公司获得更多的收益。

5.3 将任务打包发布的定价模型

5.3.1 引入行动力建立打包数的函数关系

我们分析认为一个人的精力并不是无穷的，他在工作、运动时都会消耗他的精力，当他的精力到达临界点时，他就会休息，此时他是不会去完成任务的。据此我们定义一个人的行动力为 v ，他在一次活动中消耗的行动力应该是行动到任务点的行动力和做任务的行动力之和，即

$$v = A_{cm} + A_{cw}$$

并求出行动力的效率：

$$\omega = \frac{Acw}{Acm + Acw}$$

当一个人的行动力效率即 ω 越高时，说明他做任务所占的时间更多，即他所能完成的任务也更多。我们认为将任务进行打包后发布，由一个人去任务点完成任务，它的效率比多个人分别去完成任务的行动率更高。并且打包可能导致零散的行动力被拒绝。

我们定义一个参数 u 来表示任务与会员之间吸引力的相关系数 ($0 < u < 1$)，它与会员的信誉积分 b 有关：

$$u = \frac{m_3}{1 + \frac{m_3 - u_0}{u_0}(b + 1)^{-2}}$$

由此我们得到第 i 个任务对第 j 个会员的吸引力为：

$$c_{ij} = \frac{1}{1 + \frac{m_4 d_{ij}^2}{u}}$$

我们分析认为当一个任务对一个会员的吸引力越高时，它就越会被这个会员完成，因此我们可以得到若任务完成，任务被第 i 个会员完成的概率为：

$$P(A_i|A) = c_{ij} / \sum_{i=1}^n c_{ij}$$

因此我们可以得到第 j 个任务被完成的行动力消耗的期望：

$$E(Acm_j) = \sum_{i=1}^n P(A_i|A) Acm_{ij}$$

在将任务进行打包时，我们不应只由 **app** 盈利角度出发，而应着眼于如何将互不相识，零散、完整程度不一的行动力组织起来，充分物尽其用，这才是问题的本源。因此我们就要使得我们打包后的效率满足：

$$w = \frac{Acw}{Acm + Acw} \geq Gate$$

即

$$Acw_{\text{总}} = n \cdot Acw \geq \frac{Gate \cdot Acm_{\text{总}}}{1 - Gate}$$

又有

$$Acm_{\text{总}} = \sum_{j=1}^n E(Acm_j)$$

所以我们推得

$$n > \frac{Gate \cdot \sum_{j=1}^n E(Acm_j)}{Acw(1 - Gate)}$$

在得到此式后，我们再通过带入数据技术计算出各个任务被完成的行动力消耗的期望 $E(Acm_j)$

将 $E(Acm_j)$ 在地图上表现出来，得到：

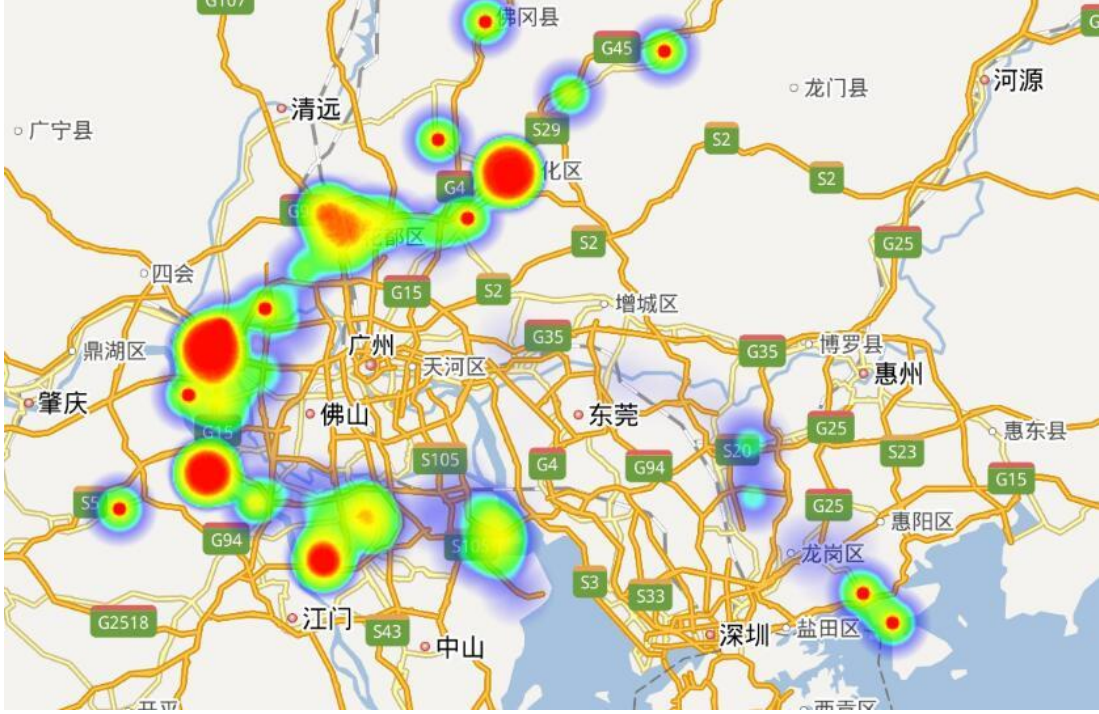


图 14 各个任务点处的行动力消耗

可以看到，距离市中心越近的任务， $E(Acm_j)$ 越大，按照公式

$$n > \frac{Gate \cdot \sum_{j=1}^n E(Acm_j)}{Acw(1 - Gate)}$$

可得结论：距离市中心越远，一包内应打越多的任务，这与前文的分析相符。

通过分析 $E(Acm_j)$ 与任务 j 距离市中心的距离 d 的关系，我们得到拟合函数：

$$E(Acm_j) = 20.5303 * (0.65477 + d)^2 - 8.88078$$

取 $Gate = 0.3$ ， $Acw = 1$ 我们得到 $n = 8.7987 * (0.65477 + d)^2 - 3.7722$

根据这个函数，我们将任务按照距离市中心的距离分为不同的带，不同带内的任务打包的目标任务数不同。

5.3.2 打包算法的实现及结果

同时，为了能同时照顾整块行动力和零散行动力，我们应区别对待不同位置的任务。

对于临近市中心、居住区的任务，应少打包，或不打包，以发挥其充分利用零散行动力的优点。对于市郊的任务，应打包，一包打多个，以保证整块行动力有足够的高的效率，完成更多的任务。据此，我们设计了算法流程图设计如下：

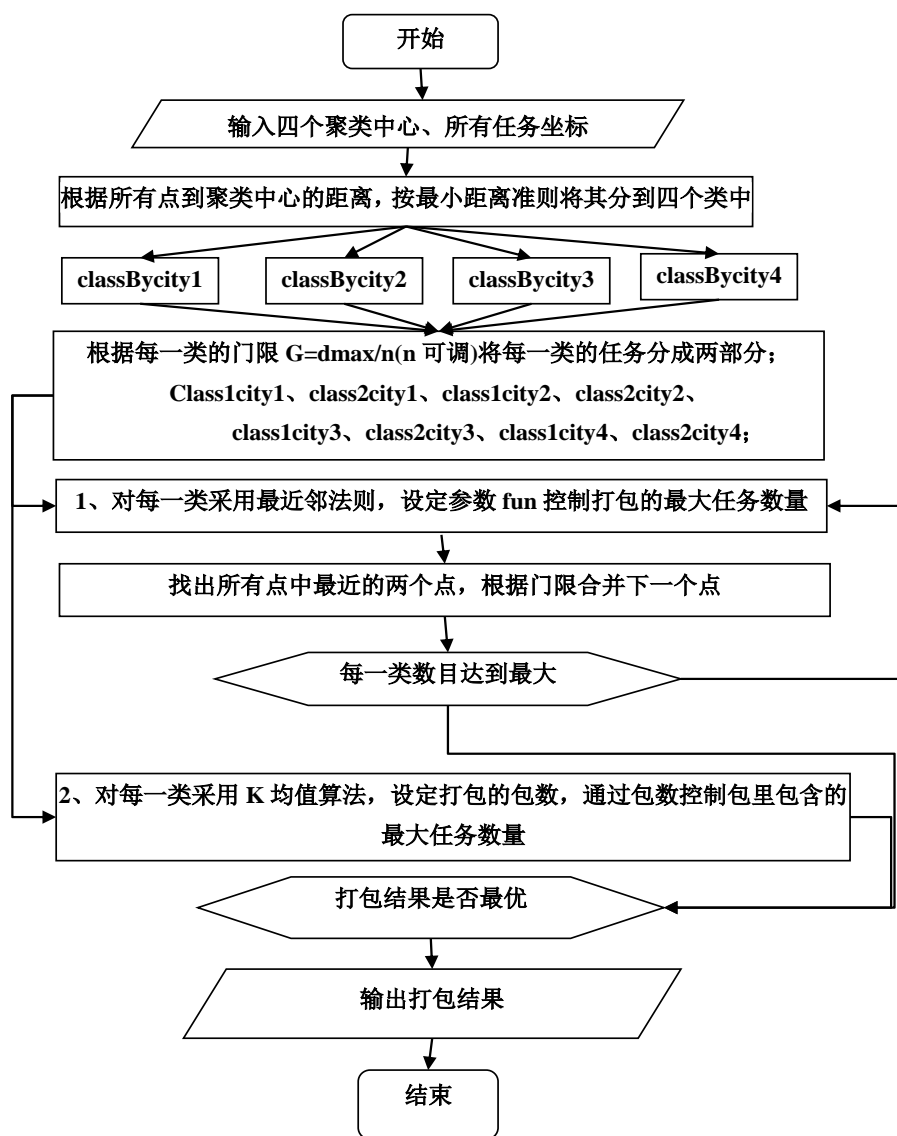


图 15 打包算法流程图

运用 MATLAB 进行运算求解可以得到各类的打包情况：
以第一大类远心部分的打包结果为例：

表 4 第一大类远心部分的打包结果

打包	序号							
1	43	19						
2	131	18						
3	77	66	107	50	57			
4	82	55	21					

5	138	128						
6	137	134	135	122	121	149	125	120
7	44	37						
8	91	40	95	32				
9	111	47	63	26	85			
10	101	29						
11	48	41	33	22				
12	73	68						
13	88	79	74	35	20	71	80	
14	109	90						
15	6	3						
16	81	76	72	93	75			
17	129	124						
18	106	59						
19	53	49						
20	144	89	56	24	45			
21	126	123						
22	108	27						
23	61	42	25					
24	114	102	98					
25	17	15						
26	10	9	7	8	5	4	13	
27	143	141						
28	142	133						
29	2	1						
30	36	23						
31	94	11	104					
32	86	64						
33	115	16	14					
34	139	130						
35	113	52						
36	112	38						

各序号对应的任务点的坐标以及其余各类的打包情况均参见支撑材料。

5.3.3 打包算法对定价模型的影响

我们的打包算法是区别对待不同位置的任务，在已算得各分类区域的打包情况后，我们利用 5.2 中的模型求出公司的收益期望与打包后的包内平均每个任务的定价价格之间的关系：

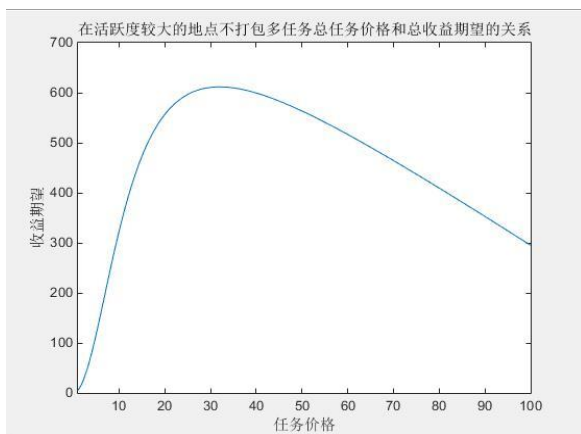


图 16 活跃度较大的地点不打包每个任务的平均定价与总收益期望的关系

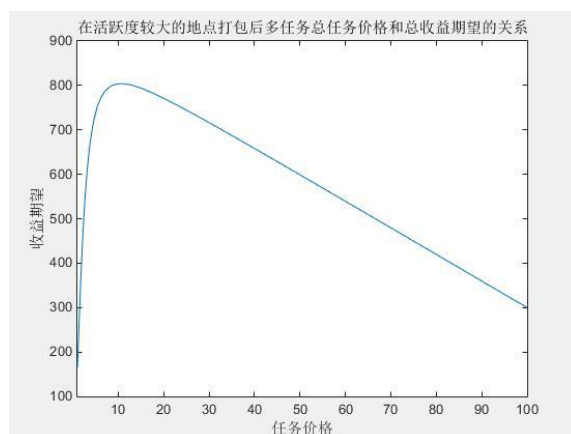


图 17 活跃度较大的地点包内每个任务的平均定价与总收益期望的关系

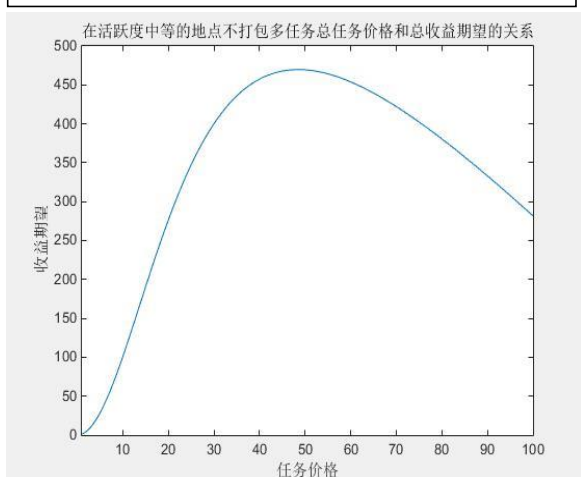


图 18 活跃度中等的地点不打包每个任务的平均定价与总收益期望的关系

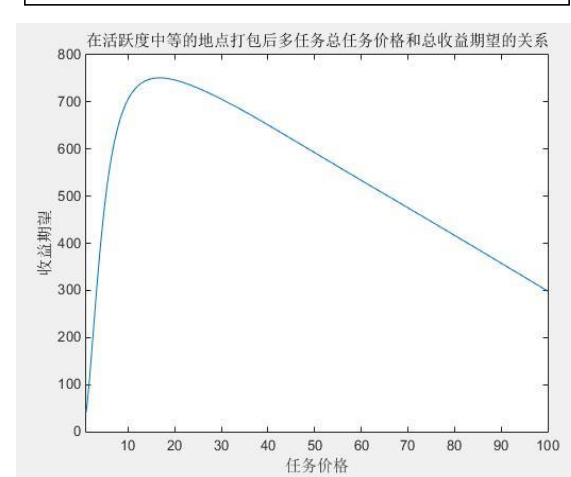


图 19 活跃度中等的地点包内每个任务的平均定价与总收益期望的关系

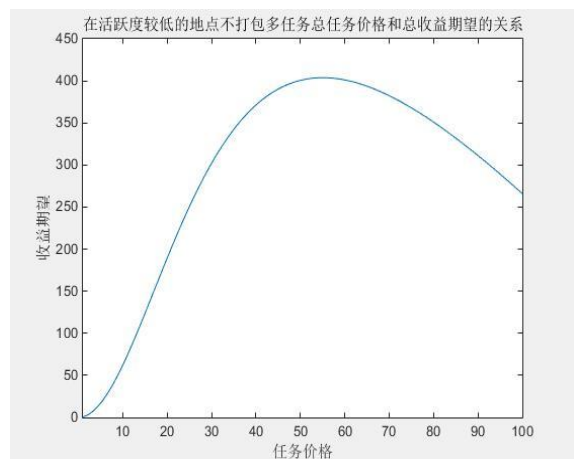


图 20 活跃度较低的地点不打包每个任务的平均定价与总收益期望的关系

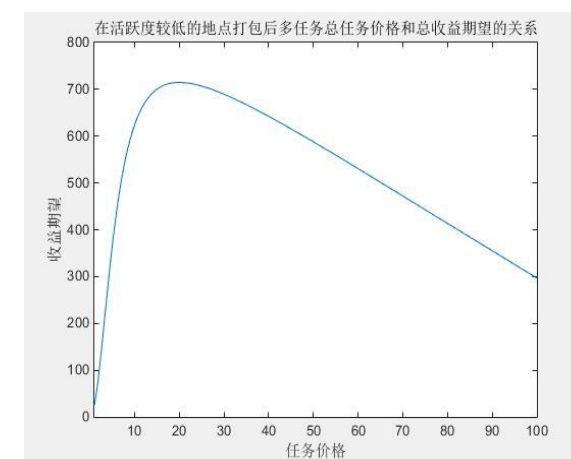


图 21 活跃度较低的地点包内每个任务的平均定价与总收益期望的关系

从上面的图中，横向分析发现打包后公司为获得最大收益期望所需确定的包内每个任务的平均价格相比于不打包时公司确定的每个任务的平均价格大大价低，而且公司能够获得的收益期望也有显著提高，也就是说采用打包策略后，公司可以在付出更少成本的情况下获得更多的收益，实现公司的利润最大化。

而通过纵向的对比分析，我们分析下面两个参数的变化情况：

$$\text{平均任务价格降低百分比} = \frac{|e - s|}{s}$$

$$\text{总收益期望提高百分比} = \frac{|E(R_{\text{后}}) - E(R_{\text{先}})|}{E(R_{\text{先}})}$$

得出结果：

表 5 不同地区打包后的参数变化情况

地区分类	平均任务价格降低百分比	总收益期望提高百分比
活跃度较大的地点	0.586206897	0.31147541
活跃度中等的地点	0.645833333	0.617021277
活跃度较低的地点	0.672727273	0.731707317

从上表可以发现，在活跃度较低的地区将任务进行打包可以获得更高的收益，实现利润的最大化。这是因为将活跃度较低的地区任务打包发布会使得会员完成一次任务可以获得更多的收益，这就大大增加了低等级会员去完成任务的积极性。

因此在打包算法模型下，我们对于临近市中心、居住区的任务，应少打包，或不打包，以发挥其充分利用零散行动力的优点，从而也使得任务的完成情况更好并使公司获得更多的利益。

5.4 对于新项目的定价方案

通过比较最新系项目任务图和用户分布图，我们发现，对于新项目而言，需要完成项目的行动力和当前用户的行动力分布是不均匀的。

原任务的行动力以及新项目的行动力分布如下：

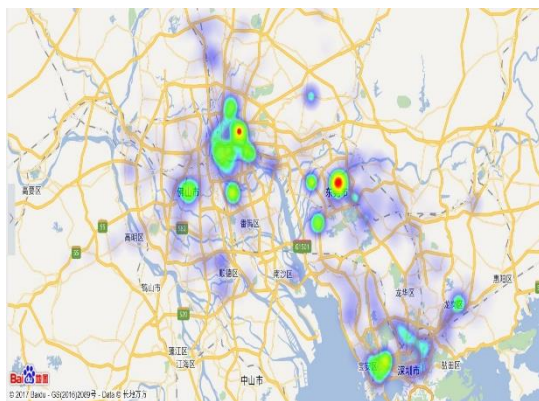


图 22 原项目需要的行动力分布

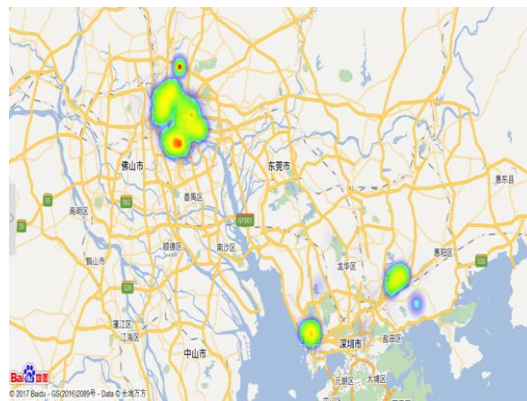


图 23 新项目需要的行动力分布

如图所示，我们发现东莞为的行动力饱和区，广州为行动力消耗饱和区。若仍由之前的模型，只考虑任务由最近城市的用户来完成，是不合理的，必须考虑行动力的流通，即行动力饱和区的用户来完成行动力消耗饱和区的任务。

我们为了区分什么样的任务应面向什么样的用户，引入了以下描述任务和用户匹配程度的变量：

1. 对于一个任务而言，不同用户的完成能力。

我们知道距离越近的用户，完成能力越强。按照经验区分，信誉积分越高的用户，完成能力越强。

2. 对于一个用户而言，不同任务的吸引力。

距离越近的用户，吸引力越强。信誉积分越高的用户，越是能抽出整块时间做任务，任务距离对于他的吸引力的影响越不显著

此外，完成能力和吸引力之间、不同任务、用户之间又有相互影响：

1、对于这个任务的完成能力，用 f 表示，相比较于完成其他任务越高，则这个任务对于用户的吸引力越强，或者说，用户的能力越是狭隘的局限于这个或几个任务，这个任务对用户的吸引力越大。

2、受其他任务吸引越少的用户，完成能力越强。因为用户面临的选择越少，他们就越可能去完成这个任务。

根据上述性质与关系，我们可以得到两个变量量化后的如下关系：

$$f_{ij} = \frac{c_{ij}}{\sum_{i=0}^n c_{ij}} \cdot \frac{1}{1 + 10ud_{ij}^2}$$

$$c_{ij} = \frac{f_{ij}}{\sum_{i=0}^n f_{ij}} \cdot \frac{1}{1 + 10ud_{ij}^2}$$

我们利用 **MATLAB** 将用户按城市聚类，则根据结果，我们可以画出各个任务完成能力最大的用户分别从属于哪个城市：

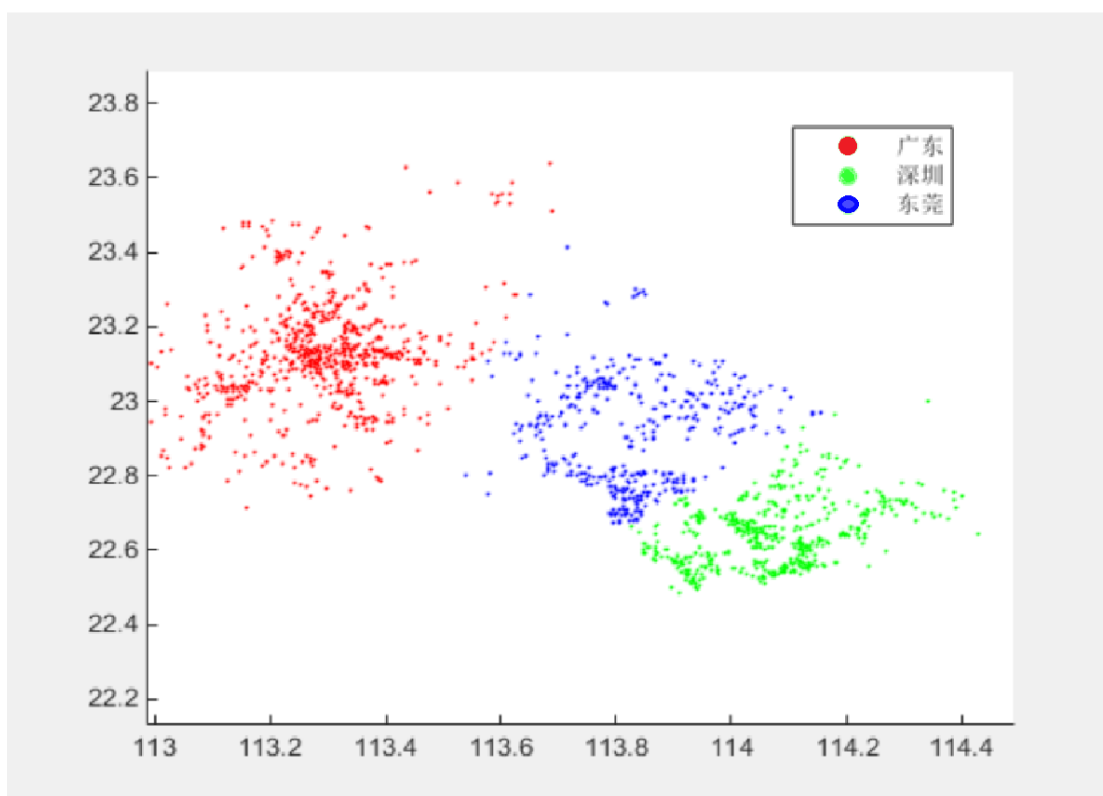


图 24 用户所属城市区分图

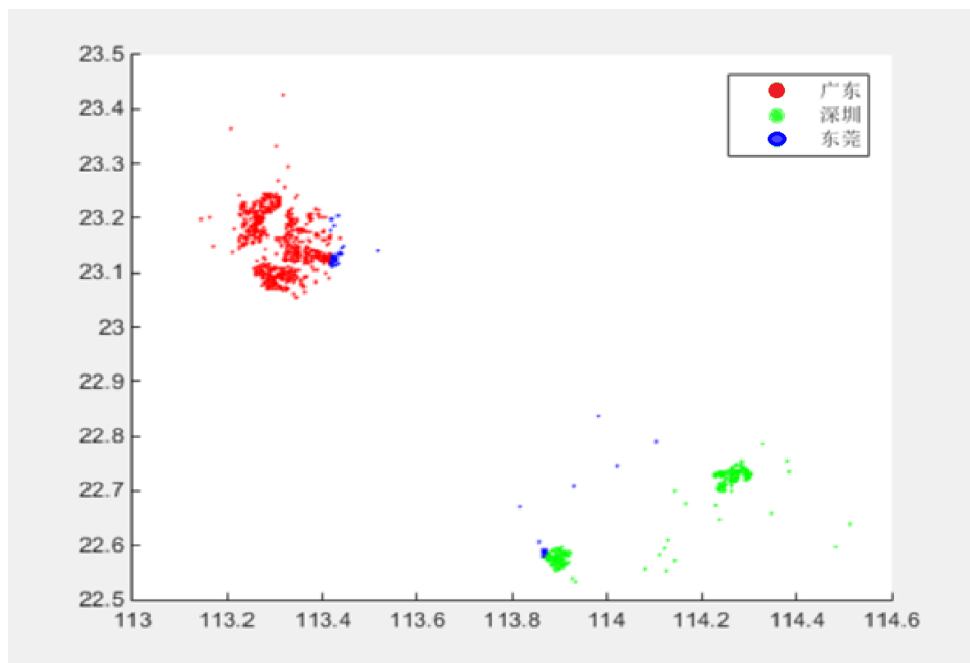


图 25 新项目各个任务完成能力最大用户所属城市

从图中，我们发现，由于广州、深圳行动力消耗饱和，东莞行动力饱和，导致广州、深圳部分靠近东莞的任务更可能被东莞用户完成，而不是当地用户。根据这一结论，为了更好的组织行动力，提高效率，在行动力饱和的广州、深圳，应适当考虑面向远道而来的东莞用户。

我们具体方法是，将东莞用户完成能力更高的任务视为需要消耗大量移动行动力的任务，尽管它们可能距离广州、深圳很近。通过这种办法，提高东莞用户来做任务的效率，吸引东莞用户前来，提高完成率。

我们用 MATLAB 依照 5.3 中的模型进行打包，部分结果如下，全部分类结果及序号对应的任务点，见支撑文件：

表 6 新项目的部分打包情况

编号	序号												
1	3	2	40										
2	516	9	12	10	515	14	517						
3	865	794	795	799									
4	994	478	482	989	481	889	214	892	213	215	321	248	757
5	613	60	61	603									
6	798	774	796	282	278	285	286	351	801				
7	615	81	71										
8	193	107											
9	38	26											
10	450	369	464	209									
11	934	911	470	762									
12	803	793											
13	546	545	522	592	20								

14	584	41											
15	691	681	181										

然后我们根据 5.2 的模型确定了这些包的最佳定价，部分打的包的定价如下，第一类定价见附录 4.3，其他分类中包的价格见支撑材料：

表 6 新项目的第一类部分任务的定价

打包编号	定价	未打包编号	定价
包 1	54.79007	6	54.79007
包 2	49.08227	7	49.08227
包 3	48.69057	8	48.69057
包 4	52.09483	11	52.09483
包 5	49.23526	15	49.23526
包 6	54.34172	17	54.34172
包 7	51.58704	22	51.58704
包 8	49.69754	23	49.69754
包 9	55.06218	33	55.06218
包 10	56.50805	34	56.50805
包 11	43.32998	35	43.32998
包 12	45.4059	43	45.4059
包 13	53.5996	45	53.5996
包 14	54.59533	48	54.59533
包 15	55.94745	49	55.94745

六、模型的评价

6.1、模型的优点

（1）我们建立了公司收益期望与任务定价价格之间的函数关系，并用图表的形式表现了出来，很直观，能准确地确定最佳的任务定价价格

（2）我们引入了任务对人的吸引力以及任务地点具有的活跃度两个概念，创新性地用这两个参数来确定该点任务被完成的概率，新颖且合理

（3）我们从会员完成任务的消耗引入了行动力这一概念，从而很好地量化了会员完成任务的实际情况，较为合理地给出了打包模型

（4）采用聚类的分析方法找到了繁杂数据中存在的规律，发现结果与现实的社会特征拟合的较好，反映了实际情况

6.2、模型的缺点

(1) 模型在求任务定价价格与任务完成率的关系时，运算过程极为的复杂，增加了实际操作的难度

(2) 实际问题中，对于任务点位置的分析还应该考虑位置点到主干道的距离，即考虑交通带来的影响

(3) 由于缺少数据，模型没有对会员的年龄、性别、收入等因素对完成任务的影响进行考虑。

七、参考文献

- [1] 姜启源 谢金星 叶俊，《数学模型（第三版）》，高等教育出版社，2003.8.
- [2] “互联网+”时代的出租车资源配置，2015 年数模国赛优秀论文.
- [3]CSDN 博客 ACdreamers ISODATA 算法，
<http://blog.csdn.net/acdreamers/article/details/44663975> .
- [4]陈名娇.基于微博数据的深圳市居民生活空间研究[D].深圳大学,2017
- [5]刘云刚，苏海宇（中山大学）.基于社会地图的东莞市社会空间研究[J].地理学报,2016,第 71 卷(8): 1283-1301
- [6] 蒋小荣，汪胜兰，杨永春（兰州大学）.中国城市人口流动网络研究——基于百度 LBS 大数据分析[J].人口与发展,2017,第 23 卷(1): 13-23

附录

附录 1 问题（1）主要代码

附录 1.1 回归分析求方程系数

```
clear; close all; clc;
x=csvread('beforehalf.txt');
datanum=size(x,1);
D=[22.6107,    113.927;
23.124, 113.2424;
22.9732,    113.7467;
22.7238,    114.1524;
];% 四个聚类中心
A1=zeros(4,1);
p=csvread('beforehalfprice.txt'); % 价格矩阵
```

```

d=zeros([datanum,1]);% 距离矩阵
pcount=zeros(datanum,1);
for i=1:datanum

    for n=1:4
        A1(n,1)=sqrt((x(i,1)-D(n,1))^2+(x(i,2)-D(n,2))^2);
    end
    d(i,1)=min(A1);

end
d1=[ones(datanum,1),d];
[b,bint,r,rint,stats]=regress(p,d1,0.05);
for i=1:datanum
    pcount(i,1)=d(i,1)*b(2,1)+b(1,1);
end

```

附录 1.3 画密度图的脚本

```

<!DOCTYPE html>
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <meta name="viewport" content="initial-scale=1.0, user-scalable=no" />
    <script type="text/javascript" src="http://api.map.baidu.com/api?v=2.0&ak=您的密
钥"></script>
    <script type="text/javascript"
src="http://api.map.baidu.com/library/Heatmap/2.0/src/Heatmap_min.js"></script>
    <title>热力图功能示例</title>
    <style type="text/css">
        ul,li{ list-style: none;margin:0;padding:0;float:left;}
        html{ height:100% }
        body{ height:100%;margin:0px;padding:0px;font-family:"微软雅黑";}
        #container{ height:500px;width:100%;}
        #r-result{ width:100%;}
    </style>
</head>
<body>
    <div id="container"></div>
    <div id="r-result">
        <input type="button" onclick="openHeatmap();" value="显示热力图"/><input
type="button" onclick="closeHeatmap();" value="关闭热力图"/>
    </div>
</body>
</html>
<script type="text/javascript">

```

```

var map = new BMap.Map("container");           // 创建地图实例

var point = new BMap.Point( 113.679983, 22.947097);
map.centerAndZoom(point, 11);                  // 初始化地图，设置中心点坐标
和地图级别
map.enableScrollWheelZoom(); // 允许滚轮缩放

var points =[
{"lng": 经度,"lat":纬度 ,"count":序号},
,
];

if(!isSupportCanvas()){
    alert('热力图目前只支持有 canvas 支持的浏览器,您所使用的浏览器不能使用热力图功能~')
}
// 详细的参数，可以查看 heatmap.js 的文档
https://github.com/pa7/heatmap.js/blob/master/README.md
//参数说明如下:
/* visible 热力图是否显示,默认为 true
 * opacity 热力的透明度,1-100
 * radius 热力图的每个点的半径大小
 * gradient {JSON} 热力图的渐变区间 . gradient 如下所示
 * {
    .2:'rgb(0, 255, 255)',
    .5:'rgb(0, 110, 255)',
    .8:'rgb(100, 0, 255)'
  }
  其中 key 表示插值的位置, 0~1.
  value 为颜色值.
 */

heatmapOverlay = new BMapLib.HeatmapOverlay({"radius":20});
map.addOverlay(heatmapOverlay);
heatmapOverlay.setDataSet({ data:points,max:200});
//是否显示热力图
function openHeatmap(){
    heatmapOverlay.show();
}
function closeHeatmap(){
    heatmapOverlay.hide();
}
closeHeatmap();
function setGradient(){
    /*格式如下所示:

```

```

        {
            0:'rgb(102, 255, 0)',
            .5:'rgb(255, 170, 0)',
            1:'rgb(255, 0, 0)'
        }*/
var gradient = { };
var colors = document.querySelectorAll("input[type='color']");
colors = [].slice.call(colors,0);
colors.forEach(function(ele){
    gradient[ele.getAttribute("data-key")] = ele.value;
});
heatmapOverlay.setOptions({"gradient":gradient});
}
//判断浏览区是否支持 canvas
function isSupportCanvas(){
    var elem = document.createElement('canvas');
    return !(elem.getContext && elem.getContext('2d'));
}
</script>

```

附录 2 问题二主要代码

附录 2.1 各地区活跃度指标函数

```

m=0;
gate=0.0001;
flag=0;
for i=1:1876
    for j=1:m
        if ((user_infor(i,1)-k(j,2))^2+(user_infor(i,2)-k(j,3))^2)<gate
            flag=1;

k(j,2)=(log(user_infor(i,3))*user_infor(i,1)+k(j,1)*k(j,2))/(log(user_infor(i,3))+k(j,1));

k(j,3)=(log(user_infor(i,3))*user_infor(i,2)+k(j,1)*k(j,3))/(log(user_infor(i,3))+k(j,1));
            k(j,1)=k(j,1)+log(user_infor(i,3)+1);
            break
        end
    end
end
if flag==1
    flag=0;
    continue
end
m=m+1;
k(m,1)=log(user_infor(i,3)+1);
k(m,2)=user_infor(i,1);

```



```

        k(m,3)=user_infor(i,2);
    end
    N=@(m)(1/(1+(1-0.001)/0.001*(m+1)^(-2)));
    for i=1:1876
        user_infor(i,4)=N(user_infor(i,3));
    end

function [g]=acti(lat,lng)
g=0;
ra=1000;
k=evalin('base','k');
[row,~]=size(k);
for i=1:row
    g=g+ra*k(i,1)*exp(200*(-(lat-k(i,2))^2-(lng-k(i,3))^2));
end
g;
g=g/10^3+1;

y=22:0.01:24;
x=112.6:0.01:114.6;
[Y,X]=meshgrid(y,x);
for i=1:201
    for j=1:201
        z(i,j)=acti(y(i),x(j));
    end
end
mesh(X,Y,z)
title('活跃度指标函数的分布');
x1=xlabel('经度');
x2=ylabel('纬度');
x3=zlabel('活跃度指标');

```

附录 2.2 公司收益与定价函数关系

```

f=@(s)(1/(1+1/PoCom(27.5,109.01,s)));
g=@(m)(f(m)*(150-m));
figure
fplot(g,[1,100],0.1)
title('(27.5,109.01)处的任务价格和收益期望的关系');
xlabel('任务价格');
ylabel('收益期望');
l=@(s)(1/(1+1/PoCom(23,113.8,s)));
q=@(m)(l(m)*(150-m));
figure
fplot(q,[1,100],0.1)

```

```

title('(23.00,113.80)处的任务价格和收益期望的关系');
xlabel('任务价格');
ylabel('收益期望');
figure
fplot(f,[1,100],0.1)
title('(27.5,109.01)处的任务价格和任务被完成概率的关系');
xlabel('任务价格');
ylabel('完成概率');
figure
fplot(l,[1,100],0.1)
title('(23.00,113.80)处的任务价格和任务被完成概率的关系');
xlabel('任务价格');
ylabel('完成概率');

function [p]=PoCom(lat,lng,s)
k=0.5;
gate=0.001;
m=150;
%R=(-s/m*log(gate/k))^(1/2);
R=1.5767;
syms a
%root=solve(['(-log(13*exp(-a^2)-12))^(1/2)='],num2str(R/s*m)],a);
root=0.28291817326180982704289519397144;
r1=double(root(1)*s/m);
r1=(r1^2)^(1/2);
r2=(-log(5*exp(-r1^2-4)))^(1/2);
r3=R;
c=@(lat0,lng0)(k*exp(-(lat0-lat)^2-(lng0-lng)^2)/s*m);
p=s/m*k*(1-exp(-r1^2/s*m))*2*pi*(acti(lat,lng)...
+acti(lat+(r1+r2)/2,lng)+acti(lat,lng+(r1+r2)/2)+acti(lat-(r1+r2)/2,lng)+acti(lat,lng-
(r1+r2)/2)...
+acti(lat+(r3+r2)/2,lng)+acti(lat,lng+(r2+r3)/2)+acti(lat-(r2+r3)/2,lng)+acti(lat,lng-
(r2+r3)/2)...
+acti(lat+(r3+r2)/2^1.5,lng+(r3+r2)/2^1.5)+acti(lat-
(r3+r2)/2^1.5,lng+(r3+r2)/2^1.5)...
+acti(lat+(r3+r2)/2^1.5,lng-(r3+r2)/2^1.5)+acti(lat-(r3+r2)/2^1.5,lng-
(r3+r2)/2^1.5));

```

附录 3 问题三主要代码

附录 3.1 打包算法的主体代码

```

% means=[22.6107, 113.927;
% 23.124, 113.2424;
% 22.9732, 113.7467;
% 23.0271, 113.1287;

```

```

% ];%四个城市聚类中心

means=[23.1585691454010,113.320688468191;
        22.6560986482226,114.098545112926];
city_num=size(means,1);

dist=zeros(datanum,city_num);
for i=1:datanum;
    for j=1:city_num;
        dist(i,j)=sqrt((x(i,1)-means(j,1))^2+(x(i,2)-means(j,2))^2);
    end
end
[Y, U]=min(dist');% U 返回 dist 最小值所在行，即分类；Y 返回最小距离；
cindex=U';% 每一个点所属的类别
distTomeans=Y';% 每一个点到所属类的中心的距离

c1=1;c2=1;c3=1;c4=1;
for i=1:datanum;
    switch cindex(i)
        case 1
            classBycity(c1,:,1)=x(i,:);
            distInclass(c1,:,1)=distTomeans(i,:);% distInclass 为每一类的距离
            c1=c1+1;
        case 2
            classBycity(c2,:,2)=x(i,:);
            distInclass(c2,:,2)=distTomeans(i,:);
            c2=c2+1;
        case 3
            classBycity(c3,:,3)=x(i,:);
            distInclass(c3,:,3)=distTomeans(i,:);
            c3=c3+1;
        case 4
            classBycity(c4,:,4)=x(i,:);
            distInclass(c4,:,4)=distTomeans(i,:);
            c4=c4+1;
    end
end
%% 将每一类中按距离分为两部分
for n=1:city_num
    dmax(n,:)=max(distInclass(:,1,n));
    dmin(n,:)=min(distInclass(:,1,n));
    % t(n)=(dmin(n,:)+dmax(n,:))/2;
    t(n)=7*dmax(n,:)/12;
    B=(distInclass~=0);

```

```

classsize(n)=sum(B(:,n));
n1(n,1)=0;n2(n,1)=0;
for i=1:classsize(n)
    if distInclass(i,1,n)<t(n)
        n1(n)=n1(n)+1;
        classIncity1(n1(n),:,n)=classBycity(i,:,n);

    else n2(n)=n2(n)+1;
        classIncity2(n2(n),:,n)=classBycity(i,:,n);

    end

end

end

end
for i=1:n1(1)
    for j=1:2
        class1city1(i,j)=classIncity1(i,j,1);
    end
end
for i=1:n1(2)
    for j=1:2
        class1city2(i,j)=classIncity1(i,j,2);
    end
end
if city_num>3
    for i=1:n1(3)
        for j=1:2
            class1city3(i,j)=classIncity1(i,j,3);
        end
    end
end
if city_num>4
    for i=1:n1(4)
        for j=1:2
            class1city4(i,j)=classIncity1(i,j,4);
        end
    end
end
for i=1:n2(1)
    for j=1:2
        class2city1(i,j)=classIncity2(i,j,1);
    end
end

```

```

end
for i=1:n2(2)
    for j=1:2
        class2city2(i,j)=classIncity2(i,j,2);
    end
end
if city_num>3
    for i=1:n2(3)
        for j=1:2
            class2city3(i,j)=classIncity2(i,j,3);
        end
    end
end
if city_num>4
    for i=1:n2(4)
        for j=1:2
            class2city4(i,j)=classIncity2(i,j,4);
        end
    end
end
end
%% 方法一
%根据最近邻法打包 调用函数 pac
index11=pac(40,'class1city1');
%index21=pac(10,'class2city1');
% index12=pac(6,'class1city2');
%index22=pac(12,'class2city2');
%index13=pac(6,'class1city3');
%index23=pac(10,'class2city3');
%index14= pac(6,'class1city4');
%index24=pac(10,'class2city4');

%% 方法二
%%k 均值打包画图 调用函数 clusterIn 和函数 plottu
% [class11, E11, F11]=clusterIn(class1city1,25);[class21, E21,
F21]=clusterIn(class2city1,35);
% [class12, E12, F12]=clusterIn(class1city2,45);[class22, E22,
F22]=clusterIn(class2city2,28);
% [class13, E13, F13]=clusterIn(class1city3,23);[class23, E23,
F23]=clusterIn(class2city3,28);
% [class14, E14, F14]=clusterIn(class1city4,23);[class24, E24,
F24]=clusterIn(class2city4,32);
%
%
```

```

% rect=[0,0,1024,600];
% figure('Position',rect);
% title('四类');
%     for j=1:classsize(1)
% plot(classBycity(j,2,1),classBycity(j,1,1),'yx');
% set(gca,'xlim',[112.8,114.5]);
% hold on
%     end
%     for j=1:classsize(2)
% plot(classBycity(:,2,2),classBycity(:,1,2),'ms');
% set(gca,'xlim',[112.8,114.5]);
%     end
%     for j=1:classsize(3)
% plot(classBycity(:,2,3),classBycity(:,1,3),'cd');
% set(gca,'xlim',[112.8,114.5]);
%     end
%     for j=1:classsize(4)
% plot(classBycity(:,2,4),classBycity(:,1,4),'rv');
% set(gca,'xlim',[112.8,114.5]);
%     end
% plot(means(:,2),means(:,1),'kp');
% hold off
%
%
%
% locat=[113.7,114.5;
%         113.0,113.7;
%         113.4,114.3;
%         112.8,113.4;];
% plottu(1,E11,E21,F11,F21,class11,class21,classsize(1),classBycity,locat(1,:));
% plottu(2,E12,E22,F12,F22,class12,class22,classsize(2),classBycity,locat(2,:));
% plottu(3,E13,E23,F13,F23,class13,class23,classsize(3),classBycity,locat(3,:));
% plottu(4,E14,E24,F14,F24,class14,class24,classsize(4),classBycity,locat(4,:));

```

附录 3.2 主体代码中方法一的函数

```

function [index]=pac(kn,fun)
rect=[0,0,1024,600];
figure('Position',rect);
scou=evalin('base',fun);
[mn,~]=size(scou);
d=zeros(mn);
index=zeros(mn,kn);
rindex=zeros(mn);
k=0;

```



```

l=0;
gate=0.02;
for j=1:mn
    for i=j+1:mn
        d(i,j)=((scou(i,1)-scou(j,1))^2+(scou(i,2)-scou(j,2))^2);
    end
end

m=0;
c=0;
te=zeros(2,100);
scatter(scou(1:mn,2),scou(1:mn,1))
for i=1:mn
    text(scou(i,2),scou(i,1),num2str(i))
end
for t=1:1000
    jj=[];
    ii=[];
    dmin=100;
    for j=1:mn
        for i=j+1:mn
            if rindex(i,j)
                continue
            end
            if dmin>d(i,j)
                dmin=d(i,j);
                k=i;
                l=j;
            end
        end
    end
    if dmin>0.0002
        break
    end
    [ii,~]=find(index==k);
    [jj,~]=find(index==l);
    if isempty(ii)
        if isempty(jj)
            c=c+1;
            index(c,1)=k;
            index(c,2)=l;
            line([scou(k,2),scou(l,2)], [scou(k,1),scou(l,1)])
            te(:,t)=[k,l];
        else

```

```

        pp=find(index(jj,:)==0);
        if ~isempty(pp)
            index(jj,pp(1))=k;
            line([scou(k,2),scou(1,2)],[scou(k,1),scou(1,1)])
            te(:,t)=[k,1];
        end
    end
else
    if isempty(jj)
        pp=find(index(ii,:)==0);
        if ~isempty(pp)
            index(ii,pp(1))=1;
            line([scou(k,2),scou(1,2)],[scou(k,1),scou(1,1)])
            te(:,t)=[k,1];
        end
    else
        a=find(index(ii,:)==0);
        b=find(index(jj,:)==0);
        if ~(isempty(a)||isempty(b)||ii==jj)
            [~,rowb]=size(b);
            if ii==10&&jj==1

                end
                if rowb-a(1)+1>0
                    for bb=1:6-rowb
                        index(ii,a(bb))=index(jj,bb);
                    end
                    index(jj,:)=index(c,:);
                    index(c,:)=zeros(1,kn);
                    c=c-1;
                    line([scou(k,2),scou(1,2)],[scou(k,1),scou(1,1)])
                end
            end
        end
    end
    rindex(k,1)=1;
end
end

```

附录 3.3 主体代码中方法二的函数一

```
function [class E F]=clusterIn(classcity,sizeofclass)
```

```

[E, F]=kmeans(classcity,sizeofclass);
ncount=zeros(sizeofclass,1);

```

```

for i=1:size(classcity,1)

    for n=1:sizeofclass
        if E(i)==n
            ncount(n)=ncount(n)+1;
            class(ncount(n),:,E(i))=classcity(i,:);

        end
    end
end
end

```

附录 3.4 主体代码中方法二的函数二

```

function plottu(k,E11,E21,F11,F21,class11,class21,classsize,classBycity,locat)
rect=[0,0,1024,600];
figure('Position',rect);
title('k');
plot(classBycity(1:classsize,2,k),classBycity(1:classsize,1,k),'go');
set(gca,'xlim',locat);
hold on
plot(F11(:,2),F11(:,1),'b+');
plot(F21(:,2),F21(:,1),'c+');
for i=1:size(class11,3)
    for j=1:size(class11,1)
        if class11(j,1,i)~=0
            plot(class11(j,2,i),class11(j,1,i),'ko');
        end
    end
end

X=zeros(2);
Y=zeros(2);
for j=1:size(class11,3)
    X(2)=class11(1,2,j);
    Y(2)=class11(1,1,j);
    for i=2:size(class11,1)
        if class11(i,2,j)~=0 && class11(i,1,j)~=0
            X(1)=X(2);
            X(2)=class11(i,2,j);
            Y(1)=Y(2);
            Y(2)=class11(i,1,j);
            line(X,Y);
        end
    end
end
end

```

```

for i=1:size(class21,3)
    for j=1:size(class21,1)
        if class21(j,1,i)~=0
            plot(class21(j,2,i),class21(j,1,i),'ro');

            end
        end
    end
X=zeros(2);
Y=zeros(2);
for j=1:size(class21,3)
    X(2)=class21(1,2,j);
    Y(2)=class21(1,1,j);
    for i=2:size(class21,1)
        if class21(i,2,j)~=0 && class21(i,1,j)~=0
            X(1)=X(2);
            X(2)=class21(i,2,j);
            Y(1)=Y(2);
            Y(2)=class21(i,1,j);
            line(X,Y);
        end
    end
end
hold off
end

```

附录 3.5 收益与价格关系代码

```

n=6;

f=@(s)(1/(1+1/PoCom(y(114),x(71),s))));% 最大值点
g=@(m)(n*f(m)*(150-m));
figure
fplot(g,[1,100])
title('在活跃度较大的地点不打包多任务总任务价格和总收益期望的关系');
xlabel('任务价格');
ylabel('收益期望');
figure
fplot(f,[1,100])
title('在活跃度较大的地点不打包多任务总任务价格和任务被完成概率的关系');
xlabel('任务价格');
ylabel('完成概率');

f=@(s)(1/(1+1/PoCom(y(114),x(71),n*s))));% 最大值点 打包

```

```

g=@(m)(f(m)*n*(150-m));
figure
fplot(g,[1,100])
title('在活跃度较大的地点打包后多任务总任务价格和总收益期望的关系');
xlabel('任务价格');
ylabel('收益期望');
figure
fplot(f,[1,100])
title('在活跃度较大的地点打包后多任务总任务价格和任务被完成概率的关系');
xlabel('任务价格');
ylabel('完成概率');

f=@(s)(1/(1+1/PoCom(y(95),x(43),s)));% 中间值
g=@(m)(n*f(m)*(150-m));
figure
fplot(g,[1,100])
title('在活跃度中等的地点不打包多任务总任务价格和总收益期望的关系');
xlabel('任务价格');
ylabel('收益期望');
figure
fplot(f,[1,100])
title('在活跃度较中地点不打包多任务总任务价格和任务被完成概率的关系');
xlabel('任务价格');
ylabel('完成概率');

f=@(s)(1/(1+1/PoCom(y(95),x(43),n*s)));% 中间值 打包
g=@(m)(f(m)*n*(150-m));
figure
fplot(g,[1,100])
title('在活跃度中等的地点打包后多任务总任务价格和总收益期望的关系');
xlabel('任务价格');
ylabel('收益期望');
figure
fplot(f,[1,100])
title('在活跃度中等的地点打包后多任务总任务价格和任务被完成概率的关系');
xlabel('任务价格');
ylabel('完成概率');

f=@(s)(1/(1+1/PoCom(y(1),x(1),s)));% 最低值
g=@(m)(n*f(m)*(150-m));
figure
fplot(g,[1,100])
title('在活跃度较低的地点不打包多任务总任务价格和总收益期望的关系');
xlabel('任务价格');

```

```

ylabel('收益期望');
figure
fplot(f,[1,100])
title('在活跃度较低的地点不打包多任务总任务价格和任务被完成概率的关系');
xlabel('任务价格');
ylabel('完成概率');

f=@(s)(1/(1+1/PoCom(y(1),x(1),n*s)));% 最低值 打包
g=@(m)(f(m)*n*(150-m));
figure
fplot(g,[1,100])
title('在活跃度较低的地点打包后多任务总任务价格和总收益期望的关系');
xlabel('任务价格');
ylabel('收益期望');
figure
fplot(f,[1,100])
title('在活跃度较低的地点打包后多任务总任务价格和任务被完成概率的关系');
xlabel('任务价格');
ylabel('完成概率');

```

用到的函数为

```

function [p]=PoCom(lat,lng,s)
k=0.5;
gate=0.001;
m=150;
%R=(-s/m*log(gate/k))^(1/2);
R=1.5767;
syms a
%root=solve(['(-log(13*exp(-a^2)-12))^(1/2)='],num2str(R/s*m)],a);
root=0.28291817326180982704289519397144;
r1=double(root(1)*s/m);
r1=(r1^2)^(1/2);
r2=(-log(5*exp(-r1^2-4)))^(1/2);
r3=R;
c=@(lat0,lng0)(k*exp(-(lat0-lat)^2-(lng0-lng)^2)/s*m);
p=s/m*k*(1-exp(-r1^2/s*m))*2*pi*(acti(lat,lng)...
+acti(lat+(r1+r2)/2,lng)+acti(lat,lng+(r1+r2)/2)+acti(lat-(r1+r2)/2,lng)+acti(lat,lng-
(r1+r2)/2)...
+acti(lat+(r3+r2)/2,lng)+acti(lat,lng+(r2+r3)/2)+acti(lat-(r2+r3)/2,lng)+acti(lat,lng-
(r2+r3)/2)...
+acti(lat+(r3+r2)/2^1.5,lng+(r3+r2)/2^1.5)+acti(lat-(r3+r2)/2^1.5,lng+(r3+r2)/2^1.5)...
+acti(lat+(r3+r2)/2^1.5,lng-(r3+r2)/2^1.5)+acti(lat-(r3+r2)/2^1.5,lng-(r3+r2)/2^1.5));

```

附录 4 问题（4）主要代码

附录 4.1 计算吸引力与完成能力

```

% dmu=zeros(2066,1876);
Ga=0.5;
AFF=@(d)(1/(1+10*uf*d));
% for j=1:1876
%     for i=1:2066
%         temp=(miss_posi2(i,1)-user_infor(j,1))^2+(miss_posi2(i,2)-user_infor(j,2))^2;
%         if temp<Ga
%             dmu(i,j)=(1/(1+10*user_infor(j,4)*temp));
%         else
%             dmu(i,j)=-1;
%         end
%     end
% end
MU=ones(2066,1876,5);
UM=ones(2066,1876,5);

for l=2:5
    for i=1:2066
        su=0;
        for j=1:1876
            if dmu(i,j)~= -1
                su=su+user_infor(j,4)*(dmu(i,j));
            end
        end
        for j=1:1876
            if dmu(i,j)~= -1
                MU(i,j,l)=UM(i,j,l-1)*(dmu(i,j))/su;
            else
                MU(i,j,l)=0;
            end
        end
    end
end
for j=1:1876
    su=0;
    for i=1:2066
        if dmu(i,j)~= -1
            su=su+MU(i,j,1)*(dmu(i,j));
        end
    end
    for i=1:2066
        if dmu(i,j)~= -1
            UM(i,j,1)=MU(i,j,1)*(dmu(i,j))/su;
        else
    
```

```

                UM(i,j,1)=0;
            end
        end
    end
end

```

附录 4.2 画图程序

```

%
% hold on
% for i=1:2066
%     [~,t]=max(MU(i,:,5));
%     switch A(t)
%         case 1
%             scatter(miss_posi2(i,2),miss_posi2(i,1),1,'r')
%         case 2
%             scatter(miss_posi2(i,2),miss_posi2(i,1),1,'y')
%         case 3
%             scatter(miss_posi2(i,2),miss_posi2(i,1),1,'g')
%         case 4
%             scatter(miss_posi2(i,2),miss_posi2(i,1),1,'b')
%     end
% end
% hold off
figure
hold on
for i=1:1876
    switch A(i)
        case 1
            scatter(temp(i,2),temp(i,1),1,'r')
        case 2
            scatter(temp(i,2),temp(i,1),1,'k')
        case 3
            scatter(temp(i,2),temp(i,1),1,'g')
        case 4
            scatter(temp(i,2),temp(i,1),1,'b')
    end
end
end
hold off

```

附录 4.2 寻找不同任务完成能力最大的用户所在城市

```

points=zeros(2066,2,3);
count=zeros(1,3);
for i=1:2066
    [~,t]=max(MU(i,:,5));

```



```

switch A(t)
case 1
    count(1)=count(1)+1;
    points(count(1),:,1)=miss_posi2(i,:);
case 2
    continue
case 3
    count(2)=count(2)+1;
    points(count(2),:,2)=miss_posi2(i,:);
case 4
    count(3)=count(3)+1;
    points(count(3),:,3)=miss_posi2(i,:);
end
end

```

附录 4.3 第一类定价结果

第一类	定价/任务	任务个数	未打包编号	定价	个数
包 1	27.9903924	3	6	54.79	1
包 2	21.3416037	7	7	49.08	1
包 3	27.1608516	4	8	48.69	1
包 4	27.4346897	13	11	52.09	1
包 5	24.6783241	4	15	49.24	1
包 6	28.1703784	9	17	54.34	1
包 7	27.4938471	3	22	51.59	1
包 8	25.5875997	2	23	49.7	1
包 9	20.7362023	2	33	55.06	1
包 10	29.6328321	4	34	56.51	1
包 11	23.1177156	4	35	43.33	1
包 12	27.0468728	2	43	45.41	1
包 13	26.8320707	5	45	53.6	1
包 14	23.2907638	2	48	54.6	1
包 15	21.4269664	3	49	55.95	1
包 16	20.455109	2	50	44.22	1
包 17	24.3371104	18	51	54.9	1
包 18	28.2448138	21	55	45.27	1
包 19	23.0896417	3	56	46.38	1
包 20	25.422729	2	57	53.19	1
包 21	20.7456771	7	59	54.26	1
包 22	27.0644505	40	63	53.23	1
包 23	28.2029752	2	65	56.05	1
包 24	28.2464901	4	67	52.21	1
包 25	27.0910266	2	72	51.23	1
包 26	28.5371189	2	75	50.83	1

包 27	24.4759798	2	76	51	1
包 28	21.1007071	2	78	43.34	1
包 29	26.775541	7	82	43.83	1
包 30	22.9702804	3	83	49.52	1
包 31	27.4370176	2	85	52.21	1
包 32	27.4304315	2	89	55.77	1
包 33	24.0533355	6	90	43.85	1
包 34	27.8203597	4	91	45.21	1
包 35	20.454391	10	92	44.64	1
包 36	20.8407545	3	94	56.44	1
包 37	25.5455077	2	99	47.72	1
包 38	22.4053355	4	105	44.71	1
包 39	29.8170219	2	106	56.06	1
包 40	26.971111	4	109	51.24	1
包 41	27.0842319	3	110	56.66	1
包 42	21.2712905	2	112	47.63	1
包 43	21.2769427	2	114	51.42	1
包 44	28.9648911	2	115	43.29	1
包 45	21.6105583	2	118	46.77	1
包 46	29.8889281	8	119	51.95	1
包 47	21.2208254	4	120	55.26	1
包 48	27.8219853	3	121	50.69	1
包 49	21.692193	8	124	55.14	1
包 50	21.3705886	3	125	48.19	1
包 51	21.4766525	2	127	55.91	1
包 52	28.7383119	2	130	54.08	1
包 53	22.2397038	3	132	55.85	1
包 54	20.0807329	2	133	53.43	1
包 55	21.2788618	2	135	50.93	1
包 56	24.0934637	6	137	54.69	1
包 57	21.035591	3	139	46.32	1
包 58	28.0448298	2	140	46.14	1
包 59	28.6635525	9	141	52.41	1
包 60	26.5345674	2	142	52.98	1
包 61	23.2073143	3	143	46.58	1
包 62	27.4648857	3	144	47.6	1
包 63	29.5820823	3	145	45.2	1
包 64	20.2233018	4	147	50.67	1
包 65	27.4374636	2	148	54.59	1
包 66	22.2382868	6	149	46.83	1
包 67	25.4971965	4	152	47.35	1
包 68	28.5037906	3	153	51.4	1
包 69	21.9173297	3	154	46.23	1

包 70	23.2451015	4	155	47.92	1
包 71	22.8011567	8	157	53.51	1
包 72	21.1601635	2	163	55.12	1
包 73	26.3659369	2	179	54.94	1
包 74	22.8089521	3	189	56.09	1
包 75	26.9794373	6	190	50.79	1
包 76	24.2534093	2	191	50.21	1
包 77	29.9237192	2	196	45.99	1
包 78	26.668791	2	197	51.84	1
包 79	26.8125148	3	198	53.96	1
包 80	28.400445	2	199	47.02	1
包 81	23.6814154	6	200	46.93	1
包 82	22.3376788	3	201	50.67	1
包 83	27.036886	5	202	48.63	1
包 84	24.8761632	5	203	56.16	1
包 85	24.9212322	12	205	49.47	1
包 86	20.6964895	2	206	54.57	1
包 87	24.1668656	12	208	49.53	1
包 88	22.1696696	2	212	49.5	1
包 89	29.3032056	3	221	44.15	1
包 90	24.6345234	2	222	52.38	1
包 91	24.1578852	6	223	43.85	1
包 92	25.2146179	9	225	50.94	1
包 93	29.7518678	3	226	56.86	1
包 94	21.6862856	4	230	47.96	1
包 95	28.5957661	4	231	46.17	1
包 96	21.22787	3	232	52.3	1
包 97	27.1716832	4	235	43.79	1
包 98	27.5180205	4	236	47.99	1
包 99	28.6684586	3	240	49.09	1
包 100	23.7357368	4	241	48.69	1
包 101	27.3308561	3	242	44.74	1
包 102	20.0741263	3	245	48.91	1
包 103	23.4792686	18	246	48.44	1
包 104	28.9437812	4	247	54.05	1
包 105	24.7905118	2	249	53.64	1
包 106	23.710912	20	250	46.07	1
包 107	20.9088528	4	253	53.18	1
包 108	25.2671388	16	256	44.77	1
包 109	27.1670294	2	258	44.45	1
包 110	23.0637035	8	264	50.34	1
包 111	22.6945245	2	267	44.83	1
包 112	23.2957121	7	268	52.51	1

包 113	22.5688161	13	272	52.77	1
包 114	25.8095194	4	274	52.57	1
包 115	29.1610843	2	280	49.22	1
包 116	23.4544028	2	281	45.36	1
包 117	21.6010059	2	283	55.92	1
包 118	23.7227625	4	284	52.81	1
包 119	23.6164986	3	288	55.05	1
包 120	23.4712767	4	290	43.57	1
包 121	21.4238216	7	291	43.64	1
包 122	28.6818878	2	293	51.92	1
包 123	22.871832	4	294	45.13	1
包 124	22.8134724	2	296	48.45	1
包 125	29.7732779	5	300	49.65	1
包 126	28.3002043	2	301	43.87	1
包 127	22.7237611	4	303	52.03	1
包 128	26.4924358	2	304	46.13	1
包 129	24.2225982	13	305	52.63	1
包 130	20.2442434	10	306	44.93	1
包 131	27.3011875	5	307	54.62	1
包 132	29.3045016	2	308	52.97	1
包 133	25.3452556	2	311	49.95	1

322	52.03602	1	654	46.32202	1
325	49.53261	1	655	53.15091	1
329	51.52624	1	660	44.46183	1
330	53.95205	1	662	48.59098	1
331	55.94606	1	663	43.50943	1
334	46.33412	1	666	47.61738	1
335	43.11321	1	668	43.28356	1
337	54.26194	1	672	53.93217	1
341	46.2421	1	673	52.69836	1
350	47.75887	1	674	53.74189	1
353	56.28831	1	678	44.51878	1
355	45.37491	1	680	55.62501	1
359	44.93353	1	688	45.82616	1
360	48.06919	1	693	51.90966	1
362	48.75331	1	694	50.76726	1
365	52.96899	1	697	48.37146	1
366	53.53355	1	701	48.68222	1
371	47.40483	1	704	48.45798	1
372	43.44729	1	705	53.29794	1
374	44.20594	1	710	48.58015	1

375	51.13422	1	714	51.00806	1
378	43.25785	1	715	48.05638	1
379	49.50016	1	716	43.69646	1
387	50.68834	1	717	52.16256	1
390	49.87694	1	721	53.15874	1
391	48.81756	1	727	46.06468	1
393	46.88262	1	732	52.94541	1
395	53.01366	1	733	45.93958	1
396	50.99024	1	735	56.27316	1
397	50.57921	1	740	50.82881	1
398	55.56129	1	745	48.41404	1
407	52.06205	1	750	54.13674	1
410	54.10094	1	751	43.22319	1
411	45.99617	1	754	45.82011	1
413	46.52386	1	755	55.64294	1
414	48.94888	1	758	47.27166	1
415	50.6068	1	760	55.93849	1
417	43.98145	1	761	49.84893	1
420	46.27834	1	764	49.26853	1
422	47.79922	1	765	53.44939	1
423	44.3662	1	770	49.48635	1
424	46.40055	1	776	55.48717	1
432	49.60294	1	777	48.89299	1
433	45.48058	1	779	45.80638	1
435	43.51885	1	780	48.48836	1
436	47.63437	1	781	54.13003	1
437	51.45157	1	785	52.7233	1
439	56.32474	1	789	47.89971	1
447	43.30856	1	790	49.14893	1
452	46.85147	1	791	47.70547	1
453	55.73759	1	800	46.30825	1
455	56.71405	1	808	53.19494	1
458	49.61815	1	810	53.86607	1
459	49.56277	1	812	53.82059	1
465	49.18825	1	814	47.67796	1
466	53.85829	1	817	43.98029	1
472	47.2211	1	824	48.36486	1
475	56.78855	1	825	49.4743	1
483	51.73047	1	826	48.1933	1
485	54.15473	1	827	48.75371	1
486	52.31973	1	832	43.31688	1
498	56.58432	1	833	55.58039	1
499	49.45585	1	835	48.91756	1

501	44.92424	1	837	47.07042	1
503	43.27446	1	841	47.29285	1
504	47.35765	1	845	49.9764	1
509	46.60993	1	847	52.8204	1
513	43.80339	1	849	56.16303	1
514	49.39087	1	857	55.7718	1
518	43.731	1	859	56.56805	1
523	55.46715	1	861	49.77738	1
531	55.05658	1	863	49.04186	1
532	49.90774	1	864	56.81282	1
534	44.71051	1	867	50.99446	1
538	43.86571	1	868	46.98085	1
540	53.94935	1	872	54.3765	1
542	50.96261	1	873	43.35602	1
544	43.41658	1	874	50.26687	1
549	48.6521	1	875	43.54293	1
559	56.15812	1	876	48.10496	1
562	55.66693	1	877	47.55666	1
563	52.50842	1	883	56.78875	1
564	47.39839	1	884	47.80275	1
566	56.81216	1	888	46.73584	1
571	45.11079	1	901	50.0893	1
573	55.93325	1	905	50.43797	1
577	54.52528	1	906	56.34855	1
579	54.94797	1	907	56.88528	1
580	49.21232	1	909	51.25841	1
582	47.924	1	913	50.31162	1
583	45.45842	1	915	52.23106	1
586	43.72704	1	919	48.08263	1
587	55.59252	1	921	49.17058	1
590	47.08775	1	922	55.60643	1
591	55.7668	1	924	49.82801	1
594	48.63403	1	929	50.21529	1
596	53.49499	1	930	52.36811	1
598	53.55926	1	931	48.69139	1
599	56.19283	1	933	47.11776	1
601	54.22617	1	935	54.38901	1
605	47.37164	1	936	48.58019	1
608	45.86271	1	963	52.40997	1
609	53.03806	1	969	45.82948	1
610	56.06061	1	972	51.22819	1
614	49.9065	1	975	43.6472	1
616	46.25804	1	977	53.61334	1

617	50.24174	1	981	48.46973	1
620	46.08236	1	982	45.00865	1
621	43.53741	1	983	47.75385	1
624	47.31576	1	986	53.16323	1
625	55.60712	1	987	48.19514	1
627	51.54629	1	990	56.40228	1
628	54.68001	1	992	48.53321	1
633	50.49342	1	995	49.51767	1
635	48.74539	1	999	50.55408	1
637	50.12054	1	1003	46.0451	1
639	49.89187	1	1006	44.82179	1
648	55.43284	1	1007	53.21027	1
649	43.61682	1	1008	54.19554	1
650	55.65372	1	1011	54.23866	1
652	44.60382	1			