

“拍照赚钱”的定价问题

摘要

当今信息时代，以互联网+众包模式进行户外信息采集与广告监测的“拍照赚钱”APP 应运而生。本文通过建立任务定价模型，对会员预订任务过程进行仿真模拟，同时计算出了不同定价策略下的任务的完成率。

针对问题一：对已结束任务的定价规律进行研究。首先，通过数据筛选剔除异常数据，利用 Matlab 将数据可视化，观察到定价与任务位置有较强的关系。计算经纬度与价格间的简单相关系数并进行显著性检验（见表 1），保留相关性更强的纬度 W_i 作为影响因素之一。将附件 1 和附件 2 导入智图交互地图，通过定性分析找出影响平台任务定价的另外四个因素，即附近任务数 Q_i 、任务难度 H_i 、任务附近会员数 Q_v 、离任务最近会员的距离 L_i 。利用 Stata 拟合出平台定价的线性回归函数（见公式 1）。最后，研究未完成任务分布，得出任务未完成的原因主要有：任务配发不均衡、会员分布密集、生活区域对会员任务预订存在影响以及任务难度过大。

针对问题二：为已结束任务设计新的定价方案。保留附近任务数 Q_i 、附近会员数 Q_v 、任务点与附近会员最近距离 L_i 这三个影响因素，并基于任务未完成的原因重新定义三个影响任务定价的因素，即根据城市的经济实力与繁华程度设置的会员预期价格 P_0' 、任务难度系数 H_i 以及会员预订批次 t_k 。对相关因素进行机理性分析，并量化各因素对定价的影响，由此建立新的定价模型（见公式 1）。根据已完成任务数据设计计算机仿真程序（见程序），模拟任务预订过程与任务完成过程。将新定价方案下任务数据代入仿真系统得到新方案下任务完成率为 79.55%，高于原方案的 62.98%，而且新方案下任务定价均值为 69.03 元，与原方案的 69.11 元相近，新任务定价方案效果较优。

针对问题三：在聚类打包销售模式下，给出新的任务定价方案。首先，利用任务之间距离数据，通过 Matlab 的 K-means 聚类方式把任务聚成了 167 类，把任务和聚类后的任务包都看成单一任务，对任务包价格根据包含的任务数量进行定价“打折”，并由此求解出各任务的任务位置点集合。以任务定价总和最小、总吸引力最大为目标函数建立优化模型，利用任务位置数据通过 Lingo 求解出各任务的最优定价，并将任务及其新的任务定价代入问题二的仿真系统（见程序）求解聚类模式下定价方案的任务完成率。完成率值为 80.03%，且任务定价均值为 65.81 元，通过与之前方案的结果对比，打包聚类销售的方式明显降低了任务定价的均值，为平台节省了成本，并且保持了较高的任务完成率。

关键词：任务定价 多元线性回归 赋权栅格电子地图 k-means 聚类 仿真模拟

一、问题重述

随着互联网时代的发展，劳务众包模式应运而生，为企业提供信息搜集和商业检查的平台，同时也为有闲暇时间的市民提供了轻松赚钱的机会。在“拍照赚钱”类 APP 上，平台发布订单，闲暇的会员从平台上预订并完成拍照任务，从而赚取赏金（如图 1）。

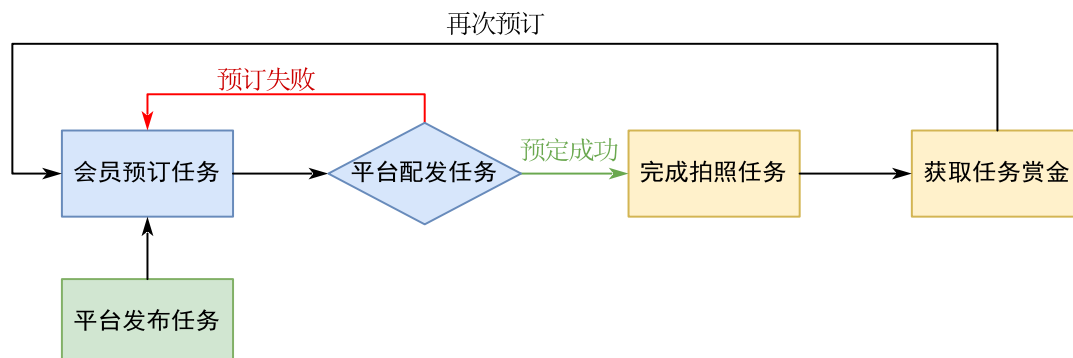


图 1 “拍照赚钱”流程

但是此类“拍照赚钱”APP 良好运营的关键则是对任务合理的定价，只有合理的定价才能吸引到 APP 会员注册，从而保证服务的提供数量；只有合理的定价才能切实为企业减少调查利润，从而吸引足够多的企业发布任务。据以上背景和给出的三个附件，需要解决以下问题：

- (1)分析已结束任务的定价规律，并分析任务未完成的原因。
- (2)为已结束任务设计新的任务定价方案，并与原方案进行比较。
- (3)在任务打包发布的模式下，修改问题二的定价方案，并分析该定价方案对最终任务完成情况的影响。

其中，附件 1 给出了已结束任务的号码、GPS 经纬度坐标、任务定价和任务完成情况，附件 2 给出平台会员的会员编号、GPS 经纬度坐标、预订任务限额、预订任务开始时间和信誉值信息，附件 3 给出了新任务的任务号码和 GPS 经纬度。

二、问题分析

2.1 问题一的分析

问题一要求我们研究附件 1 中项目的任务定价规律，分析任务未完成的原因。

(1) 首先进行数据预处理，检查附件中是否有异常数据，处理异常数据之后进行数据可视化，绘制经纬度与价格三维关系图。考虑到经纬度可能与价格有一定的影响，用 stata 对价格与经纬度两个变量进行相关性分析，如果相关关系显著，就进行多元线性回归，并分析误差。

(2) 为了进一步研究定价规律，用智图交互地图绘制任务价格与地理位置、会员位置等其他因素的影响关系图，观察到任务与附近会员最近距离、附近任务数、附近会员数和任务难度与任务定价存在较大联系。

(3) 由此，考虑使用多元线性回归的方法找出各个相关因素对任务定价影响的定量关系，由于任务难度不便确定，我们可以用除任务难度外参数的回归方程误差衡量任务难度参数对定价的影响，根据误差大小确定每个任务的难度等级，最后将所有参数重新进行多元线性回归，并对结果和误差进行分析。

(4) 同样地，用智图交互地图在地图上标注出所有任务与所有会员的位置，结合定价规律的几个分析角度以及给出的定价规律，根据未完成任务的分布情况对其未完成的原因进行分析。

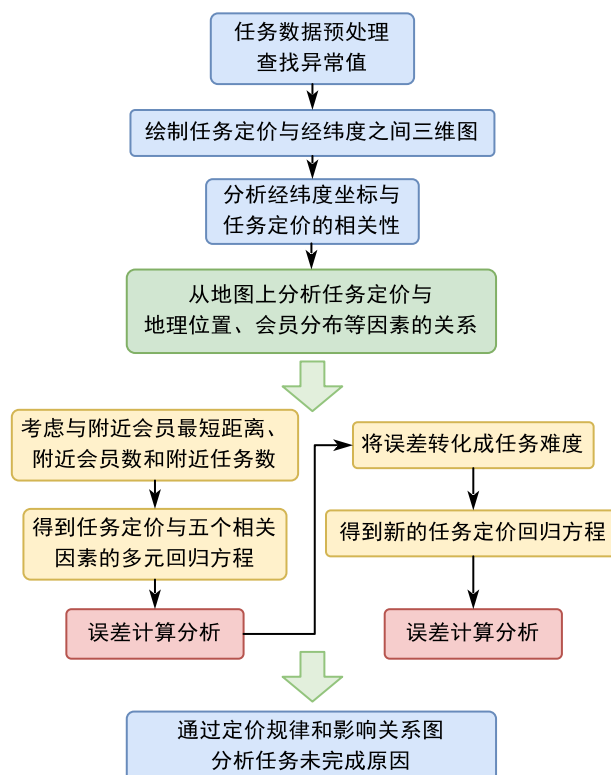


图 2 问题一思路流程图

2.2 问题二的分析

问题二要求为已结束任务重新设计定价方案，并与原方案进行比较。

(1) 基于问题一中对任务未完成原因的分析，为了保证新定价方案的合理性，首先将影响程度大的任务未完成因素作为新定价方案中一部分考虑的因素。

(2) 对任务进行定价时，我们只有附件 1 中任务相关信息和会员信息，通过问题一的分析可以发现已结束任务中有四个聚类，分别对应电子地图上的佛山市、广州市、东莞市和深圳市，确定相互之间最相近距离后，可以根据这个距离绘制栅格网络地图，由于栅格边长较小，栅格内的各因素基本相同，将每个栅格中的区域看作同一地区。通过给栅格赋权来表示该区域会员对任务的心理预期价格，并通过栅格的颜色差别、色块重叠直观地体现出会员心理预期价格随地理位置的变化规律。

(3) 之后依次量化任务难度、附近任务数、附近会员数、任务点与附近会员最近距离以及会员开放批次对任务定价的影响，从而给出合理的定价方案。

(4) 考虑到比较定价方案优劣最显著的指标是任务完成度，而任务的完成还受到会员的任务数量、对任务的满意度等因素影响，所以我们考虑采用计算机模拟来计算新定价方案的任务完成度。通过旧方案中完成度等数据和实际场景设置模拟参数，从而可以计算出新方案可信度较高的任务完成度。

(5) 比较两方案的任务完成率，分析产生差异的原因。

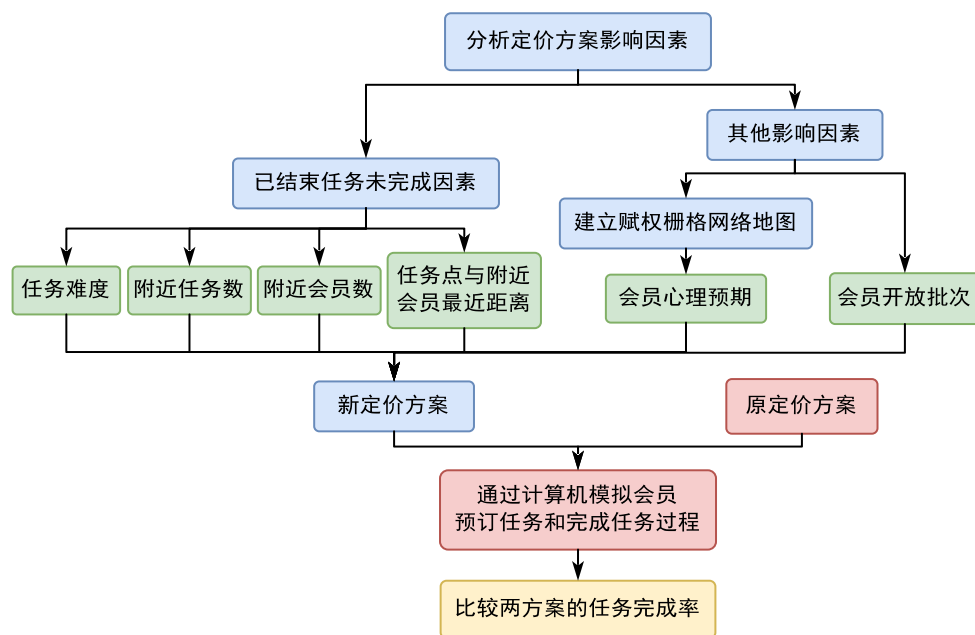


图3 问题二的思路流程图

2.3 问题三的分析

问题三要求我们站在平台的角度上考虑任务的联合打包发布，也就是考虑对任务进行聚类，对聚类后的任务给出新的定价方案。

(1) 在为任务定价前首先要考虑任务的打包，由于本问可以直接使用的任务相关信息只有经纬度坐标以及前两问关于任务定价相关因素的分析，而且位置相近的任务具有大致相同定价，所以采用以任务间距离为依据的聚类方式。

(2) 为使给出的定价方案更合理，以任务的总定价之和最小、吸引力最大为目标函数建立优化模型进行求解。

(3) 约束条件中一定有定价的约束，借鉴实际生活中商品降价联合促销的思想，根据聚类中的任务数对聚类中的每个任务价格分级“打折”。

(4) 通过定价和完成度相关约束，求解出各个任务的最优定价。

(5) 因为本问仍然是对已结束的任务给出定价方案，所以仍然可以使用第二问中设计的计算机模拟程序，比较该方案与第二问方案的完成度和总定价均值，并对结果进行分析，得出更为合理的定价策略。

三、模型假设

假设 1：会员是理性的，总会预订离自己距离最近、定价最高的任务；

假设 2：题目中给出的数据是真实可靠的；

假设 3：位置因素、价格因素、距离因素对任务订单的完成影响是独立的。

假设 4：所有任务都被预订成功。

四、符号说明

符号	说明
Q_i	任务 i 所在地附近的其他任务的数量
Q'_i	任务 i 所在地附近的会员数量
P_i	任务 i 的定价
L_i	任务 i 所在地最近会员距离
Q_t	任务附近 10km 的任务数
Q_v	任务附近 10km 的会员数
L_i	任务点离附近会员最近距离
μ_i	任务 i 对会员的吸引度之和
n_j	会员 j 的任务预订限额

注：未列出符号及重复符号以出现处为准

五、模型建立与求解

5.1 问题一的模型建立与求解

5.1.1 数据预处理

1. 识别异常数据

在对数据进行分析之前，首先对给出的数据进行预处理。附件 1 和附件 2 中分别给出了 835 个任务和 1877 个会员的经纬度坐标，用 stata 软件对会员的信誉度、预订限额等数值数据进行描述性统计分析，并没有发现过大或过小等异常数据。下一步检查数据中是否有超出经纬度各自范围的值，发现第 1175 行出现异常数据(如图 4)。也就是编号为 B1175 的会员位置信息错误，可能是在登记会员经纬度时记录失误，这里我们将 B1175 的经度与纬度交换位置。

B1173	22.95841	113.083468
B1174	22.870735	113.070645
B1175	113.131483	23.031824
B1176	23.199288	112.862711

图 4 数据中的异常值展示

2. 计算节点之间的实际距离

为了得到各任务点、各会员以及各个任务点和会员间的直线距离，我们将附件 1 和附件 2 中的经纬度坐标转换成三维坐标。以地心 O 为坐标原点，赤道平面为 xOy 平面，0 度经线圈所在的平面为 xOz 平面建立三维直角坐标系^[1]，则各个任务和会员的三维坐标为：

$$\begin{cases} x = R \cos u \cos v \\ y = R \sin u \cos v \\ z = R \sin v \end{cases} \quad (1)$$

其中， u 表示纬度， v 表示经度，地球半径 $R = 6371km$ 。根据解析几何的相关知识，任意两点 $A(u_A, v_A), B(u_B, v_B)$ 间的实际距离为：

$$d = R \arccos[\cos(u_A - u_B) \cos v_A \cos v_B + \sin v_A \sin v_B] \quad (2)$$

由此计算得到各点间的实际距离。

5.1.2 分析已结束任务的定价规律

对附件 1 中项目的价格进行统计,发现任务定价的最小值为 65 元,最大值为 85 元,可以认为平台对每个任务的定价范围是[65,85]元,下面通过进一步分析研究影响任务价格变动的因素。

1.数据可视化

为了更清晰地分析附件 1 中项目的定价规律,我们首先将任务定价可视化,从三维图中初步观察经纬度与价格的关系。利用附件 1 中任务经纬度坐标和任务定价数据,通过 Matlab 软件将任务定价可视化。

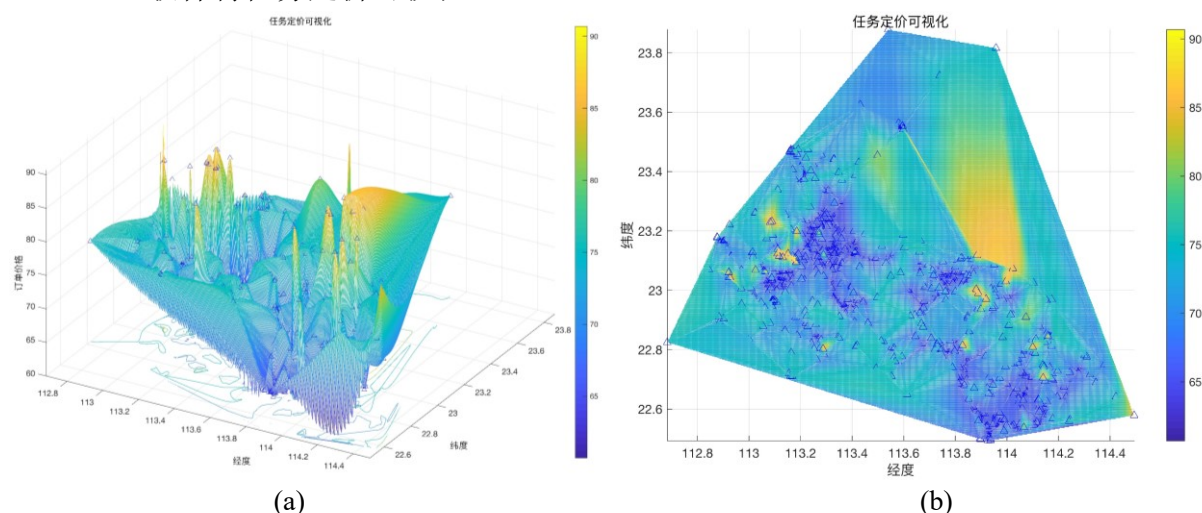


图 4 任务定价可视化

从图 4(a)中可以发现,任务的价格大部分处在较低的分段,在低分段的周边地区有若干个较高的波峰,说明任务定价很有可能随着经纬度的增减而增减。低价任务区段或存在数个定价高的任务,根据实际情况考虑,可能是因为有些任务难度较大,为了任务能够顺利完成,平台提高了任务价格来刺激会员的积极性。从图 4(b)俯视视角中可以发现,价格高的任务多分布在纬度高的地方,而低纬度地区任务价格偏低,据此,我们首先考虑平台对任务定价时是基于任务的经纬度坐标的,下面进行进一步的探究。

2.对价格和经纬度进行相关性分析

经过对三维图直观的分析,可以认为任务定价与经纬度坐标有着一定的相关性,所以我们利用 stata 对任务定价与任务的经度和纬度进行相关性的显著性检验(检验水平 $\alpha = 0.01$)并求出它们之间的相关系数(如表 1),可以看到在 99%的置信水平下,纬度与价格的相关性是显著的,说明纬度与价格是有相关关系的,且该相关关系的出现具有统计学的意义,而不是偶然的^[1]。

表 1 价格与经纬度相关性的显著性检验

	价格	纬度	经度
价格	1.0000		
纬度	0.1205* 0.0015	1.0000	
经度	-0.0597 0.2333	-0.6144* 0.0000	1.0000

由表 1 知,其中纬度与价格的相关系数为 0.1205,经度与价格的相关系数为-0.0597,纬度对价格的影响比经度大,而经度与纬度之间的相关系数高达-0.6144,可能存在共线性。为了消除变量间的共线性,这里只保留纬度 W_i 作为影响平台定价的因素之一。

3. 任务价格与地理位置的关系

通过对经纬度与价格之间的分析,我们发现任务的地理坐标与任务的价格具有一定的相关关系,为了找到地理坐标与价格之间更精确的影响关系,我们利用智图交互地图导入已结束项目的位置数据,得到价格与任务位置的直观表示来观测它的分布情况。(如图 5)

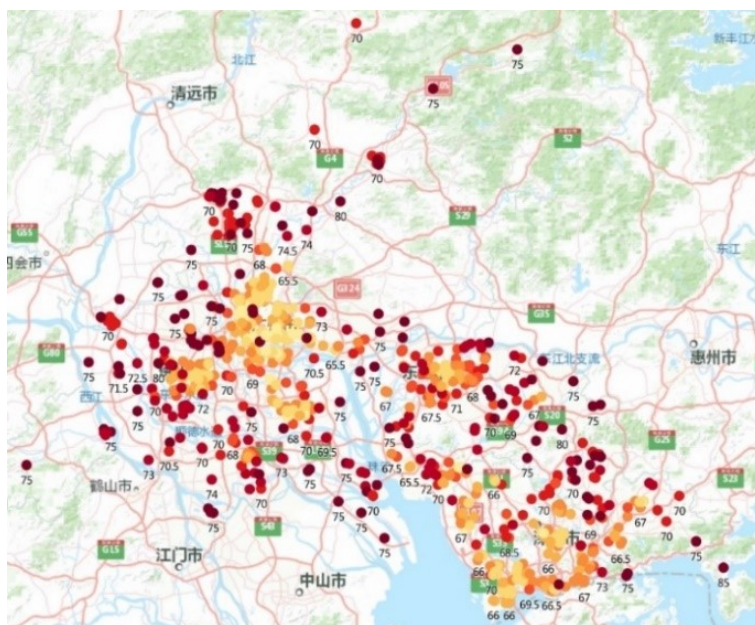


图 5 任务价格与任务位置关系图

图中圆点代表各任务,圆点下方的标注数字表示各任务的价格,圆点颜色由浅黄色过渡到深红色表示价格由低到高。从图中可以看出,任务密集区域的颜色较周边任务稀疏区域浅,也就是说任务分布越集中,价格越低。值得注意的是,有些任务虽然分布在任务密集的地方颜色却是红色,说明这些任务可能属于有一定的难度的任务,平台通过提高这类任务的价格来提高此类任务对用户的吸引力。

从图中还可以看出,在任务分布稀疏的区域一般会有较高的任务定价,因为此类任务附近很少有其他任务,若某会员在任务稀疏的区域预订某一任务后,若想再预订其他任务,会因为距离其他任务很远而增加他的任务完成成本,故平台将任务分布稀疏区域的任务价格定得较高。图中左上角与右下角等处出现了红色聚集的情况,考虑是否此处会员人数较少,平台为了吸引别处的会员完成这里的任务而提高价格。

由此看出平台会员分布也是影响定价的直接因素之一,下面对会员分布与任务价格关系进行进一步研究分析。

4. 任务价格与会员位置的关系

从上述分析可知,任务的密集程度对平台任务定价是负影响,由此联想到会员的密集程度可能对任务定价也是负影响,为了验证这一猜想,利用智图导入附件 2 中的会员数据,置于附件 1 数据的上层,观察会员分布情况与任务定价之间的关系(如图 6)。图中用蓝色圆点表示会员的位置,蓝色越深的地方表示会员数量越多,蓝色越浅的地方会员数量越少。

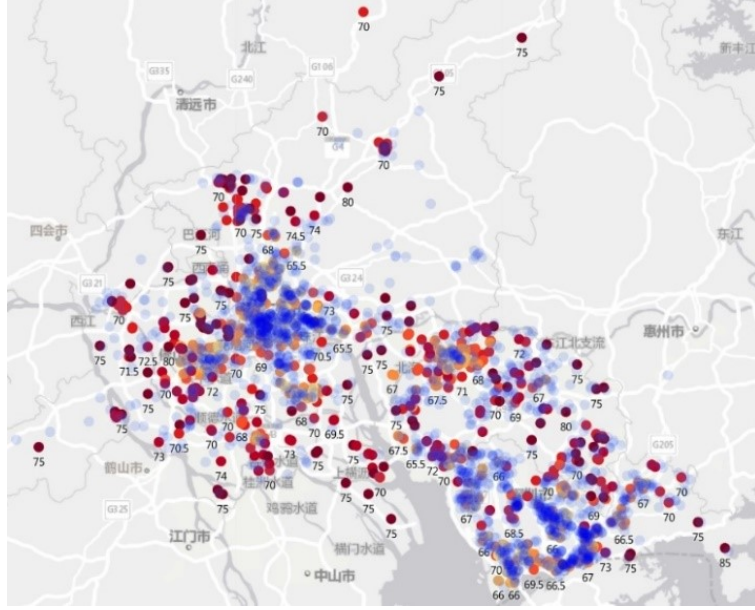


图 6 任务价格与会员数量关系图

图中，从深圳市的任务定价就可以明显看出会员数量与任务定价是反向变动的，深圳市上部区域的会员数量较下部区域少，故这片区域内的所有任务的价格普遍比下部高。左上角广州市与佛山市任务与会员都呈现中间多四周少的分布，因此任务价格由内向外随着会员数量与任务数量的减少越来越高。东莞市相比其他城市会员人数最少，且分布分散，所以该城市价格最低部分的黄颜色也比其余部分深。

对于远离任务聚集区域的单个任务，考虑平台衡量的是某一任务最近的会员离该任务的距离大小，若某一任务与附近会员的最短距离都很大的话，平台定价会高一些。

5. 任务定价与相关因素的回归分析

基于上述分析，我们确定了影响平台定价的五个因素：纬度、附近任务数、任务难度、附近会员数以及与附近会员的最短距离。其中，任务难度不容易量化，假设平台的定价只由上述五个因素决定，我们首先计算四个变量的具体数值，对除任务难度外的四个参数进行多元线性回归。

根据误差大小确定每个任务的难度等级，最后将这五个参数重新进行多元线性回归，并对结果和误差进行分析，得到已结束任务的定价规律。

➤ 任务 i 所在地附近的其他任务的数量 Q_t :

$$Q_t = \sum_{k=1}^{n-1} x_{ik}, \quad x_{ik} = \begin{cases} 1, & d(i,k) \leq r \\ 0, & d(i,k) > r \end{cases} \quad (3)$$

➤ 任务 i 所在地附近的会员数量 Q_v :

$$Q_v = \sum_{j=1}^m x_{ij}, \quad x_{ij} = \begin{cases} 1, & d(i,j) \leq r \\ 0, & d(i,j) > r \end{cases} \quad (4)$$

由此计算出每一个任务附近的任务数量与会员数量。

➤ 任务 i 所在地最近会员距离 L_i

$$L_i = \min d(i,j) \quad (5)$$

得到各因素数据后，通过 stata 软件对这四个变量进行多元线性回归分析。

(1) 首先, 对所有数据进行描述性分析。数据总体质量可以, 没有极端异常值, 变量间的量纲差距不大, 且变量的偏度与峰度也是可以接受的, 可以进行下一步分析。

(2) 在进行回归分析前, 必须要检验变量之间是否存在相关关系。相关性检验结果显示, 在 99% 的置信水平下, 任务标价与任务附近会员数、任务附近任务数、任务的纬度是之间的相关关系是显著的。

(3) 由此, 利用 stata 软件对这四个变量进行多元线性回归, 回归结果如下:

表 2 对纬度、任务数、会员数、最短距离回归结果

P	$Coef.$	$Std.Err.$	t	$P > t $	[95% $Conf. Interval$]	
L_i	0.2955191	0.1201639	2.46	0.014	0.596582	0.53138
Q_i	-0.013845	0.0077582	-1.78	0.075	-0.029073	0.0013831
Q'_i	-0.0119601	0.0026885	-4.45	0.000	-0.0172371	-0.006683
W_i	2.756873	0.6803556	4.05	0.000	1.421453	4.092293
$cons$	8.659847	15.52819	0.56	0.577	-21.81929	39.13899

由表 2 得 $P_i = 0.296L_i - 0.014Q_i - 0.012Q'_i + 2.757W_i + 8.660$, 该模型的可决系数 $R^2 = 0.2372$, 说明这四个变量只解释了定价规律的 23.72%。由于任务的定价还受任务难度的影响, 为了刻画任务难度, 我们对该模型因变量的拟合值进行预测, 得到残差图如下:

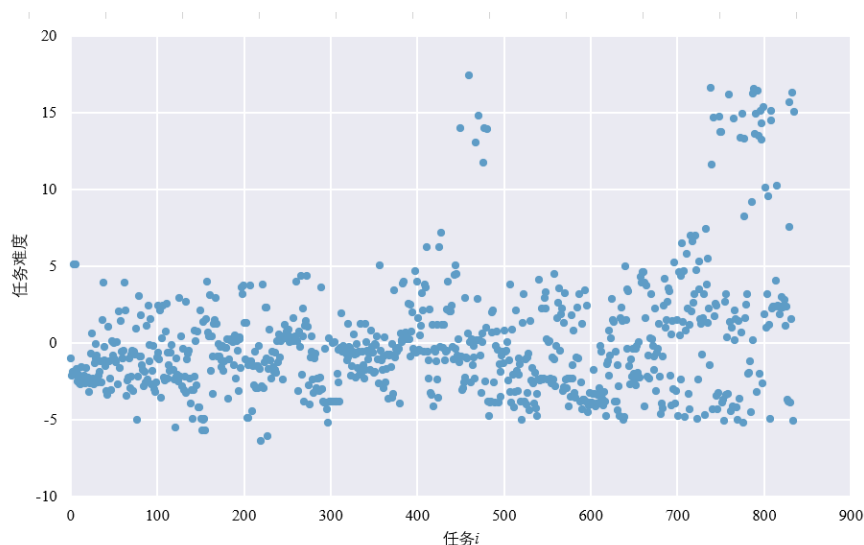


图 7 四因素的回归分析残差图

(4) 从图 7 中可以看出, 残差大部分处于 $[-5, 5]$ 之间, 部分超过 10, 这说明大部分任务难易度适中, 非常困难的任务只占少数, 由此印证了用任务难度解释预测差异是合理的。将残差的大小作为每个任务的难易度, 以此衡量任务的难度等级, 共分为七级, 价格差异与难度等级对照表如下:

表 3 价格差异与难度等级对照表

价格预测偏差	$(-10, -5)$	$(-5, -2.5)$	$(-2.5, 0)$	$(0, 2.5)$	$(2.5, 5)$	$(5, 10)$	$(10, 20)$
难度等级 H_i	1	2	3	4	5	6	7

根据表 3 得出每个任务的难度等级, 将上述四个变量加上任务难度等级后重新进行回归, 回归结果如下:

表 4 最终多元线性回归分析的结果

P	$Coef.$	$Std.Err.$	t	$P > t $	[95% $Conf. Interval$]	
Hl_i	2.99919	0.0345502	86.81	0.000	2.931374	3.067006
L_i	0.2137045	0.0378643	5.64	0.000	0.1393833	0.2880257
Q_i	-0.0083182	0.0024447	-3.40	0.001	-0.0131168	-0.0035196
Q'_i	-0.0146319	0.0008475	-17.27	0.000	-0.012953	-0.0129685
W_i	2.68814	0.2143188	12.54	0.000	2.267469	3.108811
$cons$	0.1829274	4.892477	0.04	0.970	-9.420172	9.786027

通过表 4，得到新的回归方程：

$$P_i = 2.999Hl_i + 0.214L_i - 0.008Q_i - 0.015Q'_i + 2.688W_i + 0.183 \quad (5)$$

由 stata 分析的结果知，共有 835 个任务标价参与了分析，模型的 F 值(5,829)=948.97，P 值(Prob>F)=0.0000，可以说模型整体上是非常显著的，且模型中每个系数的 p 值都是 0.000 或者 0.001，系数也是非常显著的。模型的可决系数(R-squared)=0.9244，说明模型的解释能力还是不错的，计算预测价格有 99%的数据都控制在[-2,2]的误差范围内（如图 8），即我们找到了平台任务定价的一般规律。

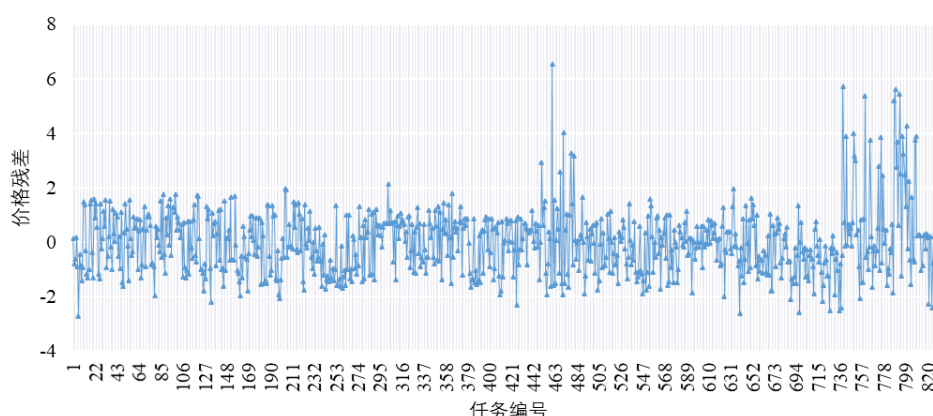


图 8 考虑五个因素定价方案的价格残差图

5.1.3 分析任务未完成的原因

要分析任务未完成的原因，就要从任务价格、位置、难度等多个角度进行分析，还要考虑会员自身的许多因素。因此，首先将数据 1 中未完成的数据利用智图交互地图在电子地图中展示出来，通过定性分析得出相关原因。

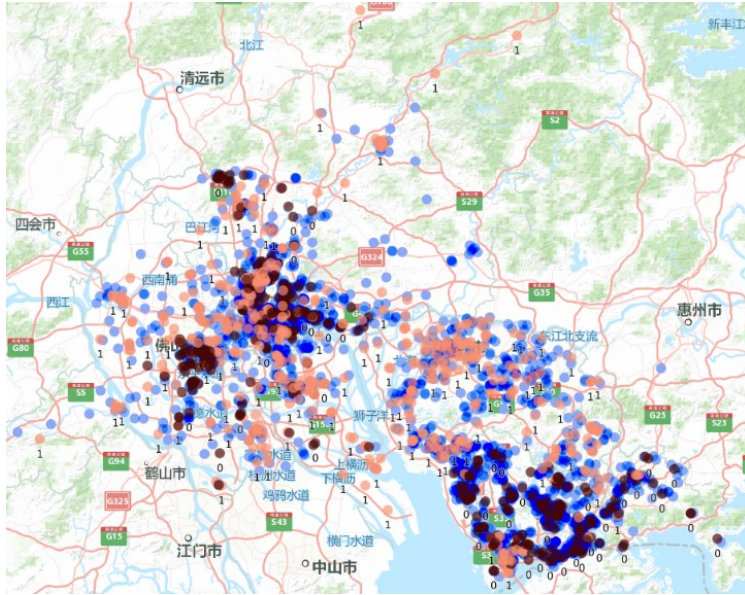


图 9 任务未完成情况图

黑色圆点表示未完成的任务，0 即未完成，1 表示已完成。从图中可以看到，黑色圆点也就是未完成的任务不仅不在城市外缘，反而是在城市中心地带，任务数与会员数都较多的地带出现了大量任务未完成的情况。根据未完成任务的分布，我们可以将未完成原因归结为以下几点：

1.任务配发不均衡，信誉度高的会员任务数过多。可能是因为在这些地方信誉度高的会员首先预订了大量的任务，但因为预订任务数太多实际能力有限导致任务无法完成，而其余有时间完成任务的会员却因为信誉度低无法成功预订。

2.会员分布密集，任务数少。因为任务与会员的分布十分密集，可能在同一预订时刻很多会员都预订了同一个任务，预订限额低的会员可能只有这一个选择的机会。

3.区域生活对会员任务预订的影响。经过对广东省的调查^[1]（调查结果如图 1），图中任务分布地区处于广东省 GDP 前四位的地市，会员消费水平较高，任务价格制定与地区繁华程度不匹配，对会员的吸引力较小(右下角深圳市)。即使附近会员分布密集，也少有人去预订拍照任务。

4.任务难度过大。在深圳和广州的部分任务可能会有风景照任务或者某些拍照区域管制严格无法入内，这都会导致任务无法完成。

2020年一季度广东各市GDP总量排行榜			
排名	地区	GDP (亿元)	增速 (%)
—	广东	22518.67	-6.7%
1	深圳	5785.6	-6.6%
2	广州	5228.8	-6.8%
3	佛山	2153.98	-7.9%
4	东莞	1923.7	-8.8%
5	惠州	879.14	-8.7%
6	珠海	703.46	-9.2%
7	茂名	676.62	-9.7%
8	江门	631.93	-11.5%
9	湛江	629.24	-7.8%
10	中山	597.22	-13.1%

图 10 2020 年广东各市 GDP 总量排行榜

5.2 问题二的模型建立与求解

问题二要求为已结束项目设计新的定价方案，根据问题一中对已结束任务的定价规律的研究，我们发现原有的定价方案的价格总体呈现低价和高价两处波峰的分布，而正常生活中商品的价格应是类似正态分布，这导致了原有定价方案下任务未完成数过多。因此，我们根据定价中的主要影响因素以及任务未完成原因制订新的任务定价方案，并通过计算机模拟的方式与原定价方案进行对比。

5.2.1 模型建立

从附件 1 的任务分布图中我们可以明显地看出，任务分布以及会员分布都是以市中心分布最为密集，并逐渐向外呈环状扩散递减。而且在两个城市的交接地带，对任务定价的影响应该是加倍的，为了更好地表示任务数和会员数对同一区域任务定价的影响，我们在电子地图基础上建立起赋权栅格电子地图，每个栅格内表示受到相同的影响，用栅格的权重代表影响的大小，并可以通过权重的相加来代表权重的相加。

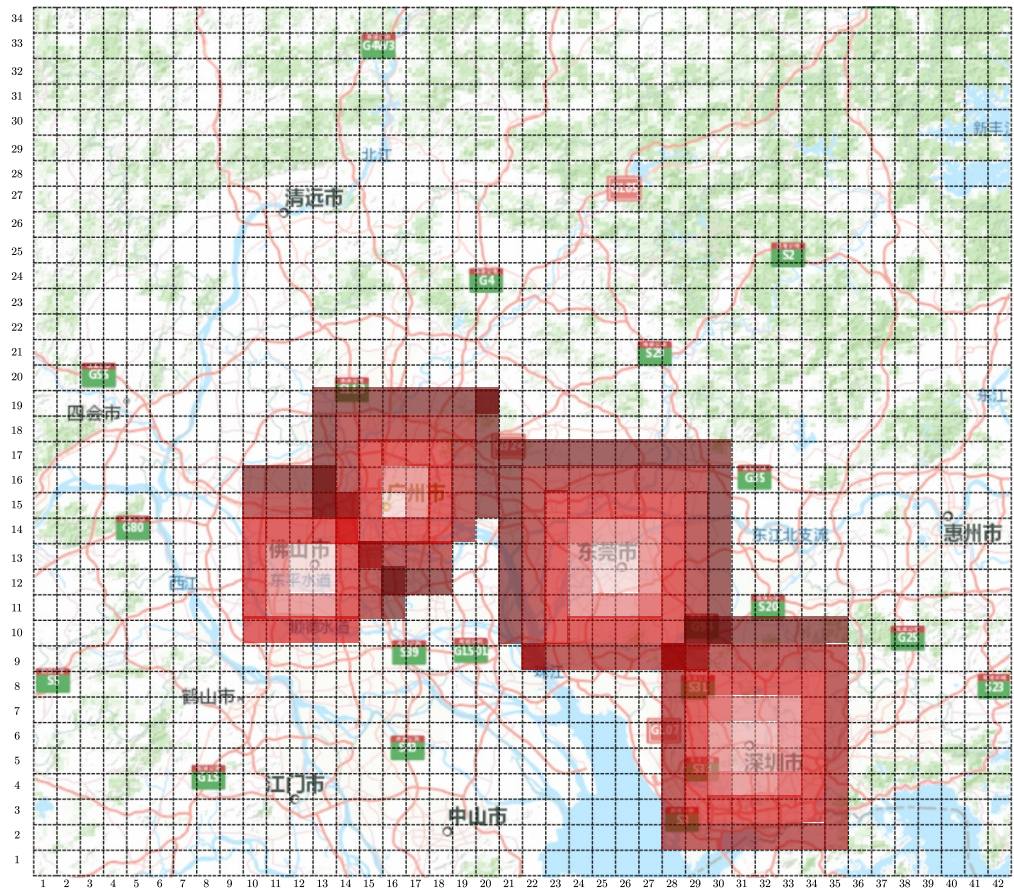


图 11 任务地区网格图（42×34）

附件 1 中的任务点几乎全都位于佛山市、广州市、东莞市和深圳市，每个城市的中心位置任务价格低且因城市不同各异，订单价格也从城市中心呈辐射状向外递增，并且在城市与城市之间的地带价格会受到两个城市的叠加影响，从而使价格更高。为了确定更合理的定价方案，并充分表示任务定价与任务点所处位置之间的关系，我们在电子地图上确定城市中心，在 1: 20 公里的比例尺下发现地图中 4 公里之内的订单价格差别不大，根据城市的大小构造 34×42 的正方形赋权栅格网络（如图 11）。

针对影响任务未完成程度大的因素，重新考虑影响任务定价的因素如下：

➤ 会员心理预期价格 P_0'

以深圳市为例，最里面的四个正方形为1级($n=1$)，再外面的十二个正方形为第二级($n=2$)，以此向外扩散，其中每个正方形的权重代表了该区域任务订单的初始任务价格 P_0 。根据城市的经济实力、影响范围以及居民习惯的不同，每个城市的初始价格各异，通过查阅《广东省统计年鉴》^[2]了解四个城市的居民收支情况，给出任务聚集城市佛山市、广州市、东莞市和深圳市的会员对任务的心理预期，也就是任务的初始定价，如下表：

表 5 各城市会员心理预期表	
城市	会员心理预期 P_0 (元)
广州市	67
佛山市	66
东莞市	66.5
深圳市	68.5

从市中心向外，随着人流量的减少和会员数量的减少，会员去完成该任务想要得到的报酬逐渐增大，由此可以归纳为：

$$P_0' = P_0 + c - 1 \quad (6)$$

式中， c ——任务点所属网格权重级别，即任务点的距离市中心的等级。

➤ 任务难度 H_i

平台发布的拍照任务难度有很大的区别。拍照检查超市商品上架情况属于难度低的任务吗，不需要花费太长的时间而且对拍照水平要求不高，而风景图等任务不仅需要较长的时间，而且对拍照水平要求较高。因此拍照任务难度越大任务的定价越高，任务难度对定价的影响 H_i 可以表示为：

$$H_i = \begin{cases} 1.1 & \text{困难} \\ 1.05 & \text{较困难} \\ 1 & \text{一般} \\ 0.95 & \text{容易} \end{cases} \quad (7)$$

任务难度作为任务的基本属性，直接影响任务的初始定价，也就是会员对人任务的心理预期价格。

➤ 附近任务数 Q_t

从经济学的竞争导向定价角度，企业对商品定价需要考虑到附近商品的影响，为了保证自己的商品在价格竞争上处于优势，从而有助于销售，往往会采用根据附近商品进行降价的策略。同理，平台为了保证任务能够全部被成功预订，会根据某区域的任务数为该区域任务定价，任务数量越多，任务的价格越低。可以将附近任务数 Q_t 表示为：

$$Q_t = \sum_{i=1}^s x_{mn}^i - 1, \quad x_{mn}^i = \begin{cases} 1, & \text{任务 } i \text{ 在 } (m,n) \text{ 方格中} \\ 0, & \text{任务 } i \text{ 不在 } (m,n) \text{ 方格中} \end{cases} \quad (8)$$

其中， x_{mn}^i ——0-1 变量，代表任务 i 是否属于 (m,n) 网格，通过累加可以得出 (m,n) 网格中的所有任务数量。

➤ 附近会员数 Q_v

从经济学的需求导向定价角度，企业对某商品的定价是根据该地区的用户需求决定的。同理，平台为了保证所有的任务都能被预订完，而且要保证 APP 有者良好的用户粘性，必须考虑任务附近的会员数量，即对任务的需求。会员数多的区域对任务的争抢热度大，任务即使价格较低也有很大可能性被成功预订，考虑到任务发布成本，平台应选择适当降价；相反，会员数少的地方，为了保证任务能够被成功预订，平台应抬高任务定价以刺激会员预订任务。附近会员数 Q_v 可以表示为：

$$Q_v = \sum_{j=1}^r x_{mn}^j, \quad x_{ij} = \begin{cases} 1, & d(i,j) \leq r \\ 0, & d(i,j) > r \end{cases} \quad (9)$$

式中， x_{mn}^j ——0-1 变量，表示会员 j 是否属于(m,n)网格，通过累加可以得出(m,n)网格中的所有会员数量。

➤ 任务点与附近会员最近距离 L_i

假设会员都是理性人，也就是所有会员都会预订离自己最近的且价格最高的任务，任务单相同价格下，距离所有会员都距离较远的任务单就不容易被成功预订，因此从平台的角度考虑，要提高与附近会员距离较远的任务价格来提高任务的预订成功率。附件 1 和附件 2 中给出了任务点和会员的经纬度信息，通过数据预处理得出各任务点与会员之间的距离，因此任务点与附近会员的最近距离 L_i ：

$$L_i = \min \{l_{i1}, l_{i2}, \dots, l_{ij}\}, i = 1, 2, \dots, s \quad (10)$$

➤ 会员开放批次 t_k

从附件 2 中可以看出，会员开放预订任务的时间也是有差别的，每三分钟将会有不定数量的会员被允许参与抢单。从平台的角度看，为了使任务都被成功预订且保证早期抢单和后期抢单的会员都有任务可选，可以对不同开放批次的任务制造价格差异，早开放的会员原则上信誉度高，对平台的忠诚度高，所以开放批次早的任务应该定价较高，同时有利于激发平台会员的积极性^[3]。

其中，每三分钟会有一批会员被允许预订任务，在 6: 30-9: 00 时间内共有 31 个批次，根据任务数目统计，前半小时共有 369 个会员开放，7: 00-8: 00 间有 449 个会员开放，8: 00 之后有 1059 个会员开放，由于前半小时的会员信誉度较高，所以考虑给予一个较高的价格，而八点之后开放的会员大部分为新会员或者活跃度不高的会员，所以适当降低该时段的任务定价。会员开放批次 t_k 对任务价格的影响可以表示为：

$$t_k = \begin{cases} 1.1 & 1 \leq k \leq 11 \\ 1 & 12 \leq k \leq 30 \\ 0.95 & k = 31 \end{cases} \quad k = 1, 2, \dots, 31 \quad (11)$$

综合以上定价相关因素，给出新的任务定价方案为：

$$\begin{aligned}
P_i &= t_k (H_i \times P_0' - \alpha Q_t - \beta Q_v + \varepsilon L_i) \\
\left\{ \begin{array}{l}
P_0' = P_0 + c - 1 \\
H_i = \begin{cases} 1.1 & \text{困难} \\ 1.05 & \text{较困难} \\ 1 & \text{一般} \\ 0.95 & \text{容易} \end{cases} \\
Q_t = \sum_{i=1}^s x_{mn}^i - 1 \\
Q_v = \sum_{j=1}^r x_{mn}^j \\
L_i = \min \{l_{i1}, l_{i2}, \dots, l_{ij}\}, \quad i = 1, 2, \dots, s \\
t_k = \begin{cases} 1.1 & 1 \leq k \leq 11 \\ 1 & 12 \leq k \leq 30 \\ 0.95 & k = 31 \end{cases} \quad k = 1, 2, \dots, 31 \\
x_{mn}^i = \begin{cases} 1, & \text{任务 } i \text{ 在 } (m, n) \text{ 方格中} \\ 0, & \text{任务 } i \text{ 不在 } (m, n) \text{ 方格中} \end{cases} \\
x_{ij} = \begin{cases} 1, & d(i, j) \leq r \\ 0, & d(i, j) > r \end{cases}
\end{array} \right. \quad (12)
\end{aligned}$$

5.2.2 模型求解

为了比较新定价方案与原来定价方案，我们主要通过计算两方案的任务定价和完成度来进行说明，可以通过逻辑斯蒂回归、计算机仿真模拟等方式得到方案的完成度，考虑到计算完成度时与会员自身选择有关，直接通过逻辑斯蒂回归等数据分析方式得到的结果说服力较差，选择通过计算机仿真模拟的方法计算并比较两种方案的完成度。

1. 预订任务过程

考虑会员对任务的选择。通过分析会员预订任务的开始时间可得所有会员分 31 批次进行预订，每批次间隔 3 分钟，总共会员数为 1877 人，除了最后一轮有 1059 人外，其他 30 批的会员分布都比较分散。那么，会员只能预订有限数量的任务，最大数量为会员的任务预订限额，因此会员在预订任务时不是随机选择，而且会员都是理性的，所以所有的会员都会优先选择距离近且任务定价高的任务。

会员选择并预订任务后，平台根据会员预订限额所占比例对任务进行分配，采用蒙特卡洛模拟法求解每个会员预订任务的成功概率。由于会员预订限额是影响任务分配的直接原因，选择某任务的会员中，任务限额大的会员占优势。我们将当前预订批次中某会员的预订限额与某订单附近 5km 内所有会员预订限额之和的比值，作为该会员能够成功分配到该订单的概率：

$$P_{si} = \frac{e_i}{\text{sum}_e}, i = 1, 2, \dots, r \quad (13)$$

式中， e_i ——该会员的任务预订限额；

sum_e ——该会员所选任务当前 5km 范围内所有会员的任务预订限额之和。

2. 任务完成过程

从问题一的分析中可以看出，任务被会员选中后不是百分之百会完成的，会员对任务的完成还受到多个方面因素的影响，如任务环境、任务价格、距任务的距离等。

（一）位置因素

为研究任务环境对任务完成率的影响，利用智图交互地图将附件 1 中所有未完成的任务映射到真实地图上，如下图

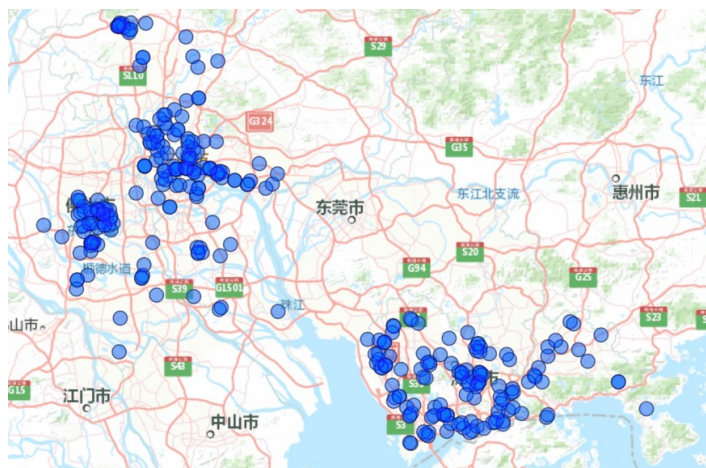


图 12 未完成订单的真实位置

从图中可以看出未完成的订单中的大部分集中在三个区域，为了更好的得出这种聚集因素，结合已有的完成订单数据通过 Matlab 利用 k-means 聚类将未完成的任务根据地理位置分作 3 类，聚类结果如下：

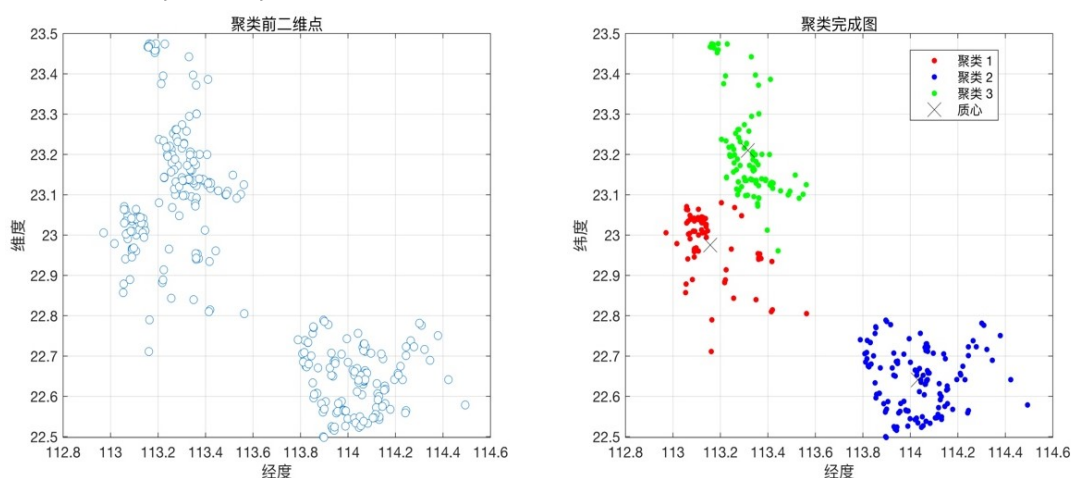


图 13 未完成任务的聚类图

得到聚类后的三个质心经纬度坐标分别为 $\{113.1571, 22.9753\}$ ， $\{114.0312, 22.6409\}$ ， $\{113.3156, 23.2101\}$ 。对应真实地图，可以得到这三类分别对应佛山市、深圳市和广州市，而且深圳和广州的未完成数量明显多，结合第一问中对任务未完成原因的分析，可以认为城市的生活水平以及繁忙程度是影响任务未完成的重要原因，城市越繁忙，会员能够完成任务的概率就越小。通过相关数据调查，深圳市、广州市及佛山市的人均 GDP 分别为广东省的第一、二、三名（如表 6），由此可以认为任务的未完成率与城市的人居环境成一定的比例。

表 6 深圳、广州、佛山三市人均 GDP 表

城市	深圳	广州	佛山
人均 GDP 元/年	5785.6	5288.8	2153.98

据此确定了订单的未完成概率与订单所在城市的繁华程度呈现一定的比例关系：

$$P_{lij} = \begin{cases} 0.7 & \text{深圳市} \\ 0.8 & \text{广州市} \\ 0.9 & \text{佛山市} \\ 0.98 & \text{其他} \end{cases} \quad (14)$$

（二）距离因素

距离对于订单完成与否的影响规律与价格的相反，会员会更愿意去完成定价高的任务，而且会考虑是否有必要去完成距离远的任务，因此二者呈现反比关系，当前距离下完成任务单的概率 P_{dis} ：

$$P_{dis} = \frac{\alpha_{dis}}{d} \quad (15)$$

式中， α_{dis} ——当前距离比例系数；

D ——会员到当前订单的距离。

对于距离对任务是否完成概率的影响，我们求出来所有会员到所有任务的距离，经过统计距离的分布如下表所示

根据现实生活中拍照赚钱类 app “蚂蚁众包”，我们可以得到会员只能够选择距离自己 30km 内的任务，app 内图片如下图所示



图 14 app 内距离限制

根据确定的距离对于完成的影响是负向的，我们大概确定距离对于任务能否完成符合如下的函数关系式：

$$P_{dis} = 1 - \lfloor d_{ij} \rfloor \times \rho, \rho = 0.2 \quad (16)$$

数值上面的体现即为下面的分段函数：

$$P_{dis} = \begin{cases} 0.9 & 25 \leq d_{ij} < 30 \\ 0.92 & 20 \leq d_{ij} < 25 \\ 0.94 & 15 \leq d_{ij} < 20 \\ 0.96 & 10 \leq d_{ij} < 15 \\ 0.98 & 0 \leq d_{ij} < 10 \end{cases} \quad (17)$$

式中， P_{dij} ——会员 i 完成距离 d_{ij} 的任务 j 的概率。

(三) 价格因素

对于拍照赚钱任务，价格是任务完成与否的决定性因素。成功预订任务的会员在执行拍照赚钱任务过程中会进行决策，会衡量当前任务价格是否能够给他带来满意的收益，因此任务价格与任务完成概率呈正相关，当前价格下完成某任务单的概率：

$$P_{price} = \alpha_{price} \times Pay \quad (18)$$

式中， α_{price} ——当前价格比例系数；

Pay ——完成当前订单可以获得的收益。

对于价格对于任务完成与否的影响，因为我们已经结合实际 app 对于距离的限制将距离限定在 30km 内，根据平时的生活常识，价格对于任务能否完成的影响应比距离要显著，且这样能更方便的比较新的定价对于所有任务完成率所带来的影响，根据前面分析得到的价格和任务完成率的正比关系，我们直接给出其分段函数如下：

$$P_{pij} = \begin{cases} 0.7 & 65 \leq price_j \leq 70 \\ 0.8 & 70 < price_j \leq 75 \\ 0.9 & 75 < price_j \leq 80 \\ 0.98 & 80 < price_j \leq 85 \end{cases} \quad (19)$$

式中， P_{pij} ——会员 i 完成价格为 $price_j$ 的任务 j 的概率。

综上，我们假设这三个因素对于任务能否完成的影响是相互独立的，假设会员 i 完成订单 j 的地理位置影响的概率为 P_{lij} ，任务价格的影响概率为 P_{pij} ，会员 i 与任务 j 之间距离影响的概率为 P_{dij} ，则会员 i 完成任务 j 的概率 P_{ij} 可表示为：

$$P_{ij} = P_{lij} \times P_{dij} \times P_{pij} \quad (20)$$

5.2.3 仿真模拟

为了使仿真过程更贴近现实，从而具有较强的说服力，在任务预订与配发过程，成功预订概率与生成的随机数都服从均匀分布 $X \sim U(0,1)$ ，因此我们使用 Matlab 中自带的 rand 函数生成一个(0,1)之间的随机数 t ，用会员的选单成功的概率 P_{si} 与 t 作比较，如果 $t < P_{si}$ ，就认为会员成功预订此任务。对于每个会员的任务完成过程，通过上面确定的模型，得到了 P_{lij} 、 P_{dij} 、 P_{pij} ，完成的概率为 $P_{ij} = P_{lij} \times P_{dij} \times P_{pij}$ ，同样使用 matlab 生成一个服从均匀分布的随机数 t ，通过大小比较判断当前的任务是否能够完成，为了避免模型的偶然性，我们对于原来的方案和新的定价方案都模拟 1000 次，取其平均值作为作为模拟得到的完成订单数。具体的会员选单的模拟系统流程图如下：

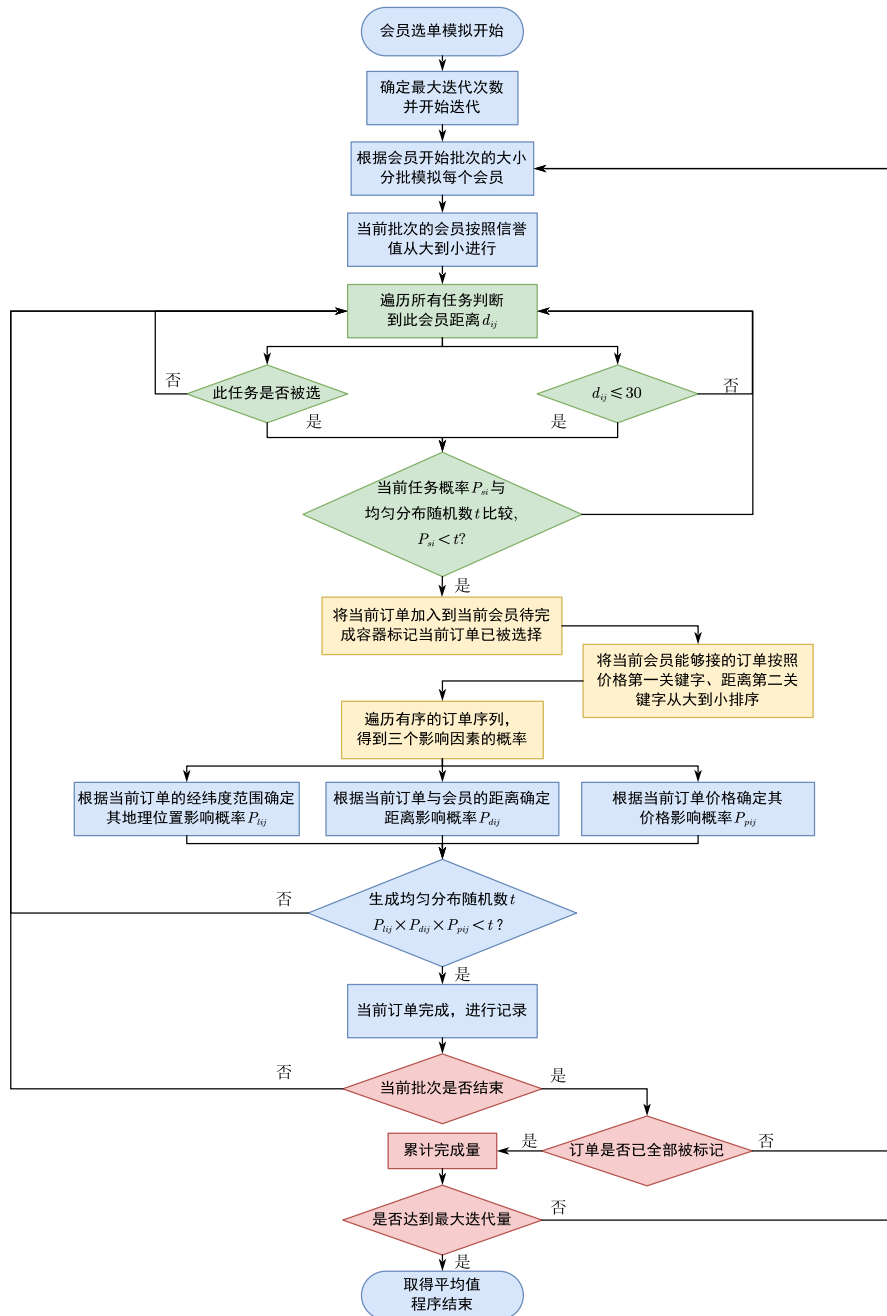


图 15 仿真模拟程序流程图

5.2.4 仿真结果

在对已结束任务进行新定价方案模拟前, 首先代入原定价方案模拟, 以检验仿真程序的准确性。根据原订单的价格, 为了避免随机概率的偶然性, 进行 1000 次蒙特卡洛模拟, 取这 1000 次完成任务数量的平均值, 得到一次模拟的平均完成任务数量为 525.861, 完成率为 62.9773%。而据附件 1 给出的数据, 我们统计得到其完成的任务数量为 522, 完成率为 62.5149%, 仿真和真实数据得到的任务完成数量和完成率几乎一致, 基于这样的结果我们认为当前的模拟仿真系统是可行的。

检验通过后, 将新定价方案代入仿真系统, 求得新定价方案下任务的完成数。

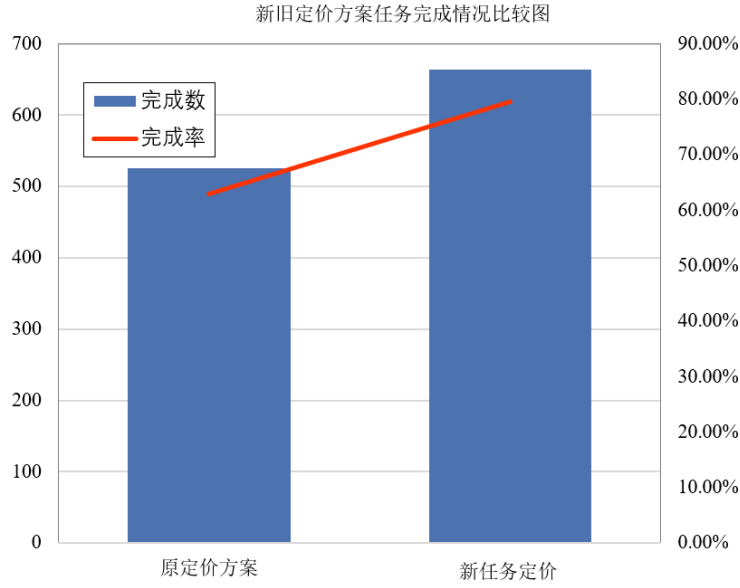


图 16 新旧定价方案任务完成情况对比

通过仿真模拟，我们得到新定价方案的完成数为 664.2010，新定价方案的完成率为 79.55%，通过与原定价方案的对比，可以发现新定价方案下任务的完成率明显提高，说明新定价方案效果显著。并且原定价方案下定价均值为 69.11 元，价格总体呈低价区高价区两波峰分布，新定价方案下定价均值为 69.03 元，价格成类似正态分布。综合以上两点，可以认为新定价方案纠正了原定价方案不切合实际的错误，节省平台任务费用的同时，提高了任务的完成率。

5.3 问题三的模型建立与求解

针对于位置集中的多个任务引起用户争抢的现象，问题三要求考虑将任务进行聚类打包并重新定价。由于问题二定价模型是根据影响因素直接确定，而聚类之后得到的任务包既存在外部因素影响，又存在包内任务内部之间影响，所以直接用相关因素量化的影响定价过于繁琐，并且定价会普遍偏低，从而影响完成率。因此，我们通过目标函数和条件约束建立优化模型从而得到满足要求的任务定价。

5.3.1 模型建立

1. 目标函数

任务吸引的对象是会员，但不只是距离某任务最近的会员才能成功预订该任务，我们需要保证的是每个任务对所有会员的吸引度之和最大，这样才有利于任务的完成。从会员角度，任务定价越高，吸引度越大且完成意愿越强；而对平台来说，成本最小且效果更好才是目标。因此，平台考虑吸引度之后，还要考虑任务的总定价最小：

$$\text{目标函数: } \begin{cases} \max \sum_{i=1}^s \mu_i \\ \min \sum_{i=1}^s P_i \end{cases} \quad (21)$$

式中， μ_i ——第 i 个任务的吸引度；

P_i ——第 i 个任务的定价。

由于双目标优化的由于直接解决双目标规划问题比困难，而且精度比较差，于是，

给首要目标吸引力一个最小值 U_0 ，将目标函数转换为约束条件，然后进行单目标优化：

$$\text{目标函数：} \min \sum_{i=1}^s P_i$$

$$\text{转化后的吸引力约束：} \sum_{i=1}^s \mu_i \geq U_0$$

2. 约束条件

➤ 吸引力约束：

我们假设吸引力 μ_i 的取值范围为 $[0,1]$ 。当吸引力的值接近零时，说明这个任务对会员毫无吸引力；当吸引力的值接近于1时，说明这个任务非常具有吸引力。考虑到任务的吸引力可以使用类似于正态分布的函数图形来描述，构造类似高斯函数：

$$\mu_i = \sum_{j=1}^r 1 - e^{-\left(\frac{P_i'}{\bar{P}_j}\right)^2} \quad (22)$$

式中， μ_i ——任务 i 对所有会员的吸引力之和， $\mu_i = \sum_{j=1}^r \mu_{ij}$ ；

P_i' ——任务 i 的最终定价；

\bar{P}_j ——会员 j 的5km范围内所有任务定价的均值。

➤ 任务定价约束

单个任务的价格一般在(50,100)元的范围内比较合适，通过聚类打包后，将所有单个任务和任务包统一看成一个任务，由任务包转化来的任务价格可以看作它组成子任务的定价之和。现实生活中的打包联合销售方式会对打包的商品进行打折，因此将打包的任务据其所含任务数进行一定的折扣，可以将任务定价表示为：

$$P_i' = \begin{cases} P_i [1 - 0.005(|S_n| - 1)] & 1 < |S_n| \leq 20 \\ P_i & |S_n| = 1 \end{cases} \quad (23)$$

式中， S_n ——任务包集合，即 $S_n = \{m_1, m_2, \dots, m_i\}$ 。

➤ 会员选择约束

由于会员都是理性人，可以认为他们总是用所有有限额来预订对自己吸引力最高的几个任务，也就是距离近且定价高的任务。由此可以表示会员选择的首要约束：

$$\sum_{i=1}^s x_{ij} = n_j \quad \forall i' \in M, \mu_{i'j} < \mu_{ij} \quad (24)$$

式中， x_{ij} ——0-1变量，1表示任务 i 被会员 j 选择；

n_j —— j 会员的任务预订限额；

M ——所有任务的集合；

μ_{ij} ——任务 i 对会员 j 的吸引力。

由于每个任务包中具有不同数量的任务数，而会员可以预订的任务数不能超出他的预订限额，记 n_i 为第 i 个任务包内的任务数量， n_j 为第 j 个会员的任务预订限额，则该约束表示为：

$$\sum_{i=1}^s x_{ij} \times n_i = n_j \quad (25)$$

当会员根据自己的意愿预订完任务后，针对每个任务，平台后台会根据预订该任务的会员的限额和信誉值分配任务，预订限额所占比例大的会员具有分配的优先权，用 y_{ij} 表示 j 任务被分配给 i 会员， h_j 表示会员 j 的信誉值，任务被会员成功预订可表示为：

$$\sum_{j=1}^r y_{ij} = 1 \quad j = \max\{h_j\}, i = 1, \dots, s \quad (26)$$

➤ 成本约束

通过定价的约束，即可满足平台花费最小的目标，在会员预订任务的配发关系确定后，用 P_i 表示任务 i 的定价，平台总的花费 c 就可以表示出来：

$$c = \sum_{i=1}^s x_{ij} \times P_i \quad (27)$$

➤ 任务吸引力约束

通过制定合理的标准，任务吸引力最大的目标函数可以转化为任务吸引力约束，从而把双目标优化问题转化为单目标优化问题，用 U_0 表示吸引力阈值，任务吸引力约束克表示为：

$$\sum_{i=1}^s \mu_i \geq U_0 \quad (28)$$

综上，任务定价优化模型可表示为：

$$\begin{aligned} & \text{目标函数：} \min \sum_{i=1}^s P_i \\ & \left\{ \begin{array}{l} \mu_i = \sum_{j=1}^r 1 - e^{-\left(\frac{P_i'}{P_j}\right)^2} \\ \sum_{i=1}^s \mu_i \geq U_0 \\ \sum_{i=1}^s x_{ij} \times n_i = n_j \\ c = \sum_{i=1}^s x_{ij} \times P_i \\ \sum_{i=1}^s x_{ij} = n_j \quad \forall i' \in M, \mu_{i'j} < \mu_{ij} \\ \sum_{j=1}^r y_{ij} = 1 \quad j = \arg \max h_j, i = 1, \dots, s \\ P_i' = \begin{cases} P_i [1 - 0.005(|S_n| - 1)] & 1 < |S_n| \leq 20 \\ P_i & |S_n| = 1 \end{cases} \end{array} \right. \quad (29) \end{aligned}$$

5.3.2 模型求解

模型的求解大致可以分为两个部分，一方面是利用任务位置数据通过 Matlab 软件进行聚类打包，接下来用 Lingo 软件对最优的任务定价进行求解，得到任务定价后，将数据代入仿真系统，求解任务的完成率。

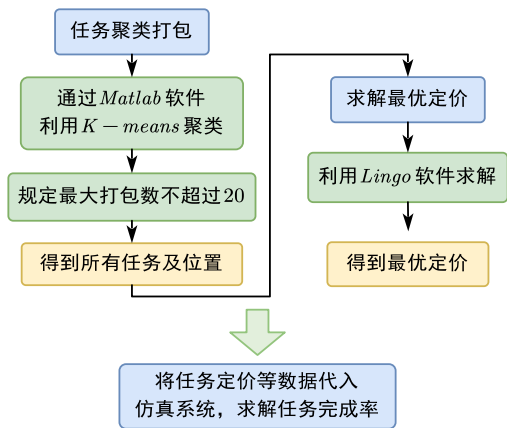


图 17 修改后新任务定价求解流程图

通过 Matlab 聚类后，得到的最大聚类数为 19，即最大的任务包所含任务数为 19，符合实际应用场景。将任务所占预订限额数、任务中心位置以及任务定价等信息代入 Lingo 软件，通过任务定价优化模型，得到每个任务定价的全局最优解。将修改前后定价方案的完成率和任务平均定价进行对比：

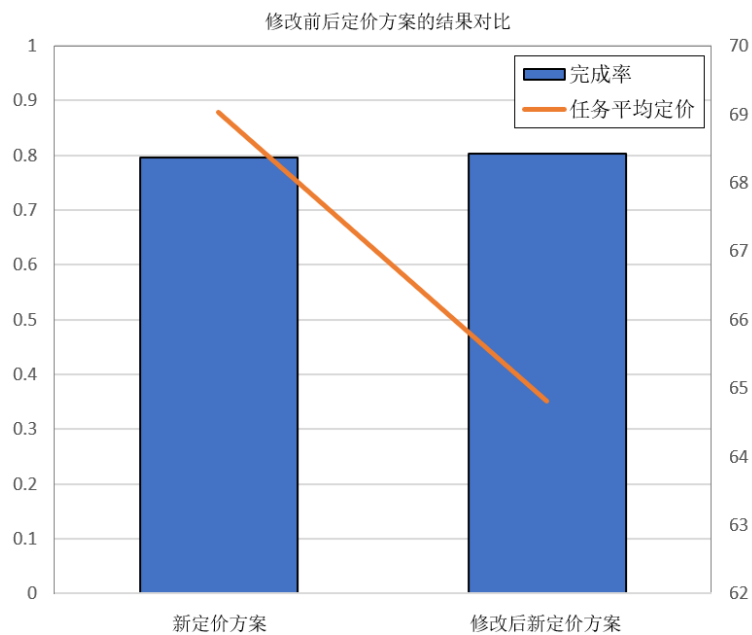


图 18 修改前后定价方案结果对比图

从图中可以看出，修改后的任务完成率并没有较大的变动，而任务平均定价大幅下降，说明打包联合发布的方案在稳定任务完成率的同时，大幅缩减了平台不必要的成本。一方面，成功预定任务包的会员若有突发状况，造成的是多个任务同时不能完成，因此使用打包联合发布的方式可能会降低任务完成率，但是修改后的定价方案通过聚类的方式提高了预订任务的门槛，信誉值小的会员无法预订，信誉值大的会员任务完成率高，所以保证了总的任务完成率。另一方面，任务平均定价的大幅下降，为平台节省大量任务成本。

八、模型的评价与推广

8.1 模型一的评价

优点：

- 1.从定性与定量两个角度分析任务的定价规律，通过在地图上绘制任务点直观清晰地观察定价的定性规律，并利用统计软件 `stata` 给出影响因素对定价的影响程度。
- 2.考虑到回归分析时变量间的共线性，只保留与任务标价相关系数大的参数。
- 3.理论分析并联系实际，对任务未完成的原因作出了合理解释。

缺点及改进：

对除任务难度外的其他参数回归得到的方程误差衡量任务难度大小，并以此划分任务难度等级，再次回归得到的方程解释能力为 92.44%，仍存在误差，说明影响任务定价的因素可能未找全。为提高定价规律的拟合程度，可采用非线性回归的方式。

8.2 模型二的评价

优点：

1. 从原定价方案的未完成原因出发，控制任务定价的同时大幅提高了任务完成率。
2. 通过计算机仿真，模拟任务预订过程与任务完成过程，充分考虑会员自身心理期望与行为属性，契合题目描述，任务优先分配给预订限额大的会员，任务完成时考虑位置、距离、价格三大因素，

缺点及改进：

仿真系统中把价格对任务完成率影响设置偏高，和实际情况有一定的误差。考虑将位置、距离及价格三大完成率影响因素的系数平均，会更符合实际，但对完成率实际的影响并不大。

8.3 模型三的评价

优点：

- 1.以任务完成率为目标建立优化模型来确定任务定价，确保了任务定价方案的质量。
- 2.仍然代入原仿真系统进行模拟，得到的结果与原定价方案可比性较强。
- 3.将双目标优化问题转化为单目标优化问题，保证了结果精确性。

缺点及改进：

Lingo 软件对迭代次数大的问题求解能力较弱，求解最优解时间太长。考虑使用 Matlab 软件通过模拟退火等智慧算法加快求解速度。

8.5 模型的推广

定性并定量地对定价影响因素进行分析，一步步归纳出定价规律，这种分析方式可以用到现实生活中基于数据挖掘的规律研究。

任务定价的优化模型可以确定一定条件下最优的任务价格，同样地，将任务换成商品也可以求解市场上大多数商品的最优定价，使得商家能够以最低的成本获得最大的反馈。

参考文献

[1]司守奎, 孙兆亮, 孙玺菁. 数学建模算法与应用[M]. 国防工业出版社, 2015.

[2]广东省统计局.广东统计年鉴[J].广州:中国统计出版社,2019.

[3]刘晓钢. 众包中任务发布者出价行为的影响因素研究[D].重庆大学,2012.

附 录

程序编号	1-1	文件名称	dataprework.m	程序说明	数据处理
<pre> clc,clear,close; % 数组中存在非数值数据如字符串时候用元胞数组读取 [~,~,finor] = xlsread('已结束项目任务数据.xlsx'); [~,~,vip] = xlsread('会员信息数据.xlsx'); [~,~,newor] = xlsread('新项目任务数据.xlsx'); %% 处理已经完成任务单数据 finor(1,:) = []; N = size(finor,1); % 处理元胞数组中的字符串 for i = 1:N string = char(finor(i,1)); % 当前元胞单元格的字符串 string(1) = []; finor(i,1) = {str2double(string)}; % 将任务号码转化为数字形式 end finor = cell2mat(finor); % 经纬度异常值处理 for i = 1:N if finor(i,2) >= 90 && (finor(i,3) <= 90 && finor(i,3) >= 0) fprintf('已完成订单第 %d 条数据出现异常值\n',i); tmp = finor(i,2); finor(i,2) = finor(i,3); finor(i,2) = tmp; elseif finor(i,2) > 180 finor(i,3) > 180 fprintf('已完成订单 %d 条数据出现异常值\n',i); finor(i,:) = []; end end end %% 处理会员信息数据 vip(1, :) = []; N = size(vip,1); for i = 1:N string = char(vip(i,1)); % 当前元胞单元格的字符串 string(1) = []; vip(i,1) = {str2double(string)}; % 将任务号码转化为数字形式 tmp = char(vip(i,2)); index = find(tmp == ' '); string1 = tmp(1:index-1); string2 = tmp(index+1:end); vip(i,2) = {str2double(string1)}; vip(i,6) = {str2double(string2)}; end vip = cell2mat(vip); </pre>					

```

tmp1= vip(:,1);tmp2=vip(:,2);tmp3=vip(:,6);
vip(:,1)=[];vip(:,1)=[];vip(:,4)=[];
vip = [tmp1,tmp2,tmp3,vip];

% 经纬度异常值处理
for i = 1:N
    if vip(i,2) >= 90 && (vip(i,3) <=90 && vip(i,3) >=0)
        fprintf('会员第 %d 条数据出现异常值\n',i);
        tmp = vip(i,2);
        vip(i,2)=vip(i,3);
        vip(i,2) = tmp;
    elseif vip(i,2) >180 || vip(i,3) > 180
        fprintf('会员第 %d 条数据出现异常值\n',i);
        vip(i,:)=[];
    end
end
%% 处理新项目任务数据

newor(1,:) = [];
N = size(newor,1);
% 处理元胞数组中的字符串

for i =1:N
    string = char(newor(i,1)); % 当前元胞单元格的字符串
    string(1) = [];
    newor(i,1) = {str2double(string)}; % 将任务号码转化为数字形式
end
newor = cell2mat(newor);
% 经纬度异常值处理
for i = 1:N
    if newor(i,2) >= 90 && (newor(i,3) <=90 && newor(i,4) >=0)
        fprintf('新订单第 %d 条数据出现异常值\n',i);
        tmp = newor(i,2);
        newor(i,2)=newor(i,3);
        newor(i,2) = tmp;
    elseif newor(i,2) >180 || newor(i,3) > 180
        fprintf('新订单第 %d 条数据出现异常值\n',i);
        newor(i,:)=[];
    end
end
N = size(finor,1);
M = size(vip,1);

viptoor=zeros(M,N);
R=6370;
for i = 1:M
    for j=1:N
        % 点 a 的经度为 u_a , 纬度 v_a
        ua = vip(i,3);    va = vip(i,2);
        ua = deg2rad(ua);va = deg2rad(va);

        %点 b 的经度为 u_b , 纬度 v_b
        ub = finor(j,3);    vb = finor(j,2);
        ub = deg2rad(ub);    vb = deg2rad(vb);

        % 计算城市 i 和 j 的距离
        viptoor(i,j) = R * acos( cos(ua - ub) * cos(va) * cos(vb) + sin(va) * sin(vb) );
    end
end

```

```

end

vipd = zeros(M,M);
% 转化经纬度求距离矩阵
for a = 1 : M
    for b = 1 : a
        if(a == b)
            vipd(a,b) = 1000000;
            continue;
        end
        % 点 a 的经度为 u_a, 纬度 v_a
        ua = vip(a,3);    va = vip(a,2);
        ua = deg2rad(ua); va = deg2rad(va);
        %点 b 的经度为 u_b, 纬度 v_b
        ub = vip(b,3);    vb = vip(b,2);
        ub = deg2rad(ub); vb = deg2rad(vb);
        % 计算城市 i 和 j 的距离
        vipd(a , b) = R * acos( cos(ua - ub) * cos(va) * cos(vb) + sin(va) * sin(vb) );
    end
end
vipd = vipd + vipd';

N = size(finor,1);
d = zeros(N); % 初始化两个城市的距离矩阵全为 0
% 转化经纬度求距离矩阵
for a = 1 : N
    for b = 1 : a
        if(a == b)
            d(a,b) = 1000000;
            continue;
        end

        % 点 a 的经度为 u_a, 纬度 v_a
        ua = finor(a,3);    va = finor(a,2);
        ua = deg2rad(ua); va = deg2rad(va);

        %点 b 的经度为 u_b, 纬度 v_b
        ub = finor(b,3);    vb = finor(b,2);
        ub = deg2rad(ub); vb = deg2rad(vb);

        % 计算城市 i 和 j 的距离
        d(a , b) = R * acos( cos(ua - ub) * cos(va) * cos(vb) + sin(va) * sin(vb) );
    end
end

d=d+d';
M = size(vip,1);
ornearvip=zeros(1,N);
ornearor = zeros(1,N);
for i =1:N
    for j=1:N
        if i==j
            continue;
        end
        if d(i,j) <= 10
            ornearor(i)=ornearor(i)+1;
        end
    end
end
end

```



```

end
for i = 1:N
    for j = 1:M
        if viptoor(j,i) < 10
            ornearvip(i) = ornearvip(i)+1;
        end
    end
end

Li = min(viptoor);
save('data.mat','finor','vip','newor','viptoor','vipd','d','ornearvip','ornearor','Li');

```

程序编号	1-2	文件名称	ShowInMap.m	程序说明	价格分布可视化
------	-----	------	-------------	------	---------

```

clc,clear,close;
load data.mat;

x = finor(:,3);y=finor(:,2);z=finor(:,4);
x_ =min(x(:,:)):(max(x)-min(x))/500:max(x(:,:));
y_ =min(y(:,:)):(max(y)-min(y))/500:max(y(:,:));
[xi,yi]=meshgrid(x_,y_);

zi=griddata(x , y , z , xi , yi , 'cubic'); %这里有三种方法 linear cubic 可以自行选择
figure(1)
scatter3(x,y,z,'b^');
hold on;
meshc(xi , yi , zi);
hold on
colorbar;
hold on;
shading interp
hold on
title('任务定价可视化');
xlabel('经度');ylabel('纬度');zlabel('订单价格');

```

程序编号	1-3	文件名称	main1.m	程序说明	k-means 聚类
------	-----	------	---------	------	------------

```

clc,clear,close;
load data.mat;
y= finor(:,2);x=finor(:,3);

figure(1), plot(x,y,'o','MarkerSize',6,'MarkerFaceColor',[255,255,255]/255);
title('聚类前二维点');xlabel('X 轴');ylabel('Y 轴');

%随机获取 150 个点
% X = [randn(50,2)+ones(50,2);randn(50,2)-ones(50,2);randn(50,2)+[ones(50,1),-ones(50,1)]];
X = [x,y];
opts = statset('Display','final');

%调用 Kmeans 函数
%X N*P 的数据矩阵
%Idx N*1 的向量,存储的是每个点的聚类标号
%Ctrs K*P 的矩阵,存储的是 K 个聚类质心位置
%SumD 1*K 的和向量,存储的是类间所有点与该类质心点距离之和
%D N*K 的矩阵, 存储的是每个点与所有质心的距离;

[Idx,Ctrs,SumD,D] = kmeans(X,7,'Replicates',10,'Options',opts);

%画出聚类为 1 的点。X(Idx==1,1),为第一类的样本的第一个坐标；X(Idx==1,2)为第二类的样本的第二个坐标
figure(2);

```

```

plot(X(Idx==1,1),X(Idx==1,2),'r','MarkerSize',14)
hold on
plot(X(Idx==2,1),X(Idx==2,2),'b','MarkerSize',14)
hold on
plot(X(Idx==3,1),X(Idx==3,2),'g','MarkerSize',14)
hold on
plot(X(Idx==4,1),X(Idx==4,2),'k','MarkerSize',14)
hold on
plot(X(Idx==5,1),X(Idx==5,2),'m','MarkerSize',14)
hold on
plot(X(Idx==6,1),X(Idx==6,2),'c','MarkerSize',14)
hold on
plot(X(Idx==7,1),X(Idx==7,2),'y','MarkerSize',14)
hold on

%绘出聚类中心点,kx 表示是 x
plot(Ctrs(:,1),Ctrs(:,2), 'kx', 'MarkerSize', 15, 'MarkerEdgeColor', [180,100,100]/255,'linewidth',5);
legend('Cluster 1','Cluster 2','Cluster 3','Cluster 4','Cluster 5','Cluster 6','Cluster 7','Centroids','Location','NW')

Ctrs
SumD

```

程序编号	2-1	文件名称	Setprice.m	程序说明	价格设置
<pre> clc,close,clear; load data.mat; center = [83,68.5; 379,67; 608,66; 772,66.5]; n = size(finor,1); m = size(finor,2); new = zeros(n,m); new(:,1:3) = finor(:,1:3); alpha = 0.0294; beta = 0.058; theta = 0.8053; for i = 1:n id = new(i,1); dis = [d(id , center(1,1)) ,1,center(1,2); % 分别存储，距离、城市号、中心点 d(id,center(2)) ,2,center(2,2); d(id,center(3)) ,3,center(3,2); d(id,center(4)) ,4,center(4,2)]; dd = sort(dis,1);belong = dd(1,2);tbelong = dd(2,2); mi=dd(1,1);tmi =dd(2,1); % 按照距离排序， 获取距离最近的城市编号以及距离 if mi <= 4 new(i,4) = center(belong,2); continue; end if mi > 24 new(i,4) = 75; continue; end if mi<16 && (ceil(mi/4) == ceil(tmi/4)) </pre>					

```

        price = max(center(belong,2),center(tbelong,2));
        price = price + 2*ceil(mi/4);
        if price > 85
            price = 85;
        end
        if price < 65
            price = 65;
        end
        new(i,4) = price;
        continue;
    end
    if mi > 4 && mi <= 24
        price = floor(mi) + center(belong,2);
        if price > 85
            price = 85;
        end
        if price < 65
            price = 65;
        end
        new(i,4) = price;
    end
    price = new(i,4);
    price = price - alpha*ornearor(id) - belta * ornearvip(id) + theta * Li(id);
    if price > 85
        price = 85;
    end
    if price < 65
        price = 65;
    end
    new(i,4) = price;
end

save('newpri.mat','new');

```

程序编号	2-2	文件名称	main2_2.m	程序说明	对任务聚类
<pre> load data.mat; clc,clear,close; id= find(finor(:,5) ==0); id = id' uncom = finor(id(:),:); y= uncom(:,2);x=uncom(:,3); subplot(1,2,1), plot(x,y,'o','MarkerSize',6,'MarkerFaceColor',[255,255,255]/255),grid on; title('聚类前二维点');xlabel('经度');ylabel('纬度'); X = [x,y]; opts = statset('Display','final'); %调用 Kmeans 函数 %X N*P 的数据矩阵 %Idx N*1 的向量,存储的是每个点的聚类标号 %Ctrs K*P 的矩阵,存储的是 K 个聚类质心位置 %SumD 1*K 的和向量,存储的是类间所有点与该类质心点距离之和 %D N*K 的矩阵, 存储的是每个点与所有质心的距离; [Idx,Ctrs,SumD,D] = kmeans(X,3,'Replicates',10,'Options',opts); </pre>					

%画出聚类为 1 的点。X(Idx==1,1),为第一类的样本的第一个坐标； X(Idx==1,2)为第二类的样本的第二个坐标

```
subplot(1,2,2)
```

```
plot(X(Idx==1,1),X(Idx==1,2),'r.','MarkerSize',14)
```

```
hold on
```

```
plot(X(Idx==2,1),X(Idx==2,2),'b.','MarkerSize',14)
```

```
hold on
```

```
plot(X(Idx==3,1),X(Idx==3,2),'g.','MarkerSize',14)
```

```
hold on
```

```
plot(X(Idx==4,1),X(Idx==4,2),'m.','MarkerSize',14)
```

```
hold on
```

%绘出聚类中心点,kx 表示是 x

```
plot(Ctrs(:,1),Ctrs(:,2) , 'kx' , 'MarkerSize' , 15 , 'MarkerEdgeColor' , 'k','linewidth',14);
```

```
grid on;
```

```
title('聚类完成图')
```

```
xlabel('经度');ylabel('纬度');
```

```
legend('聚类 1','聚类 2','聚类 3','质心','Location','NW')
```

```
Ctrs
```

```
SumD
```

程序编号	2-3	文件名称	Main2.m	程序说明	蒙特卡洛模拟原 价格
<pre> clc,clear,close; load data.mat; com = 0; tmp = sortrows(finor,5); idx= find(tmp(:,5) ==1); sort(idx); idx = idx(1);idx = idx-1; tmp1 = tmp(1:idx,:); tmp2 = tmp(idx+1:end,:); figure(1); plot(tmp1(:,3),tmp1(:,2),'kx','MarkerSize',7); hold on; plot(tmp2(:,3),tmp2(:,2),'o','MarkerSize',5,'MarkerFaceColor',[36,169,255]/255,'MarkerEdgeColor',[36,169,255]/255); grid on ; xlabel('经度');ylabel('纬度'); legend('未完成的任务','完成的任务'); for round =1:1000 [vipinc,id] = sortrows(vip,5); time = vipinc(:,5); time = unique(time); totvipcnt = size(vip,1); nowor = finor; % 动态维护当前还未被选择的订单 vipmaxor = 30; % 会员可以选取的最大订单数量 timeid = zeros(length(time),2); N = length(time); R = 57; % 会员可以选择的任务距离半径 </pre>					

```

totorcnt = size(finor,1);
tmp = zeros(totorcnt,1);
nowor=[nowor,tmp];
complete = zeros(1,totorcnt); % 统计最后每个订单的完成情况
oldprice = finor(:,4);

%% 得到同一时间所有会员的起始下标和终止下标
for i=1:N
    tmpid = find(vipinc(:,5) == time(i));
    tmpid = sort(tmpid);
    timeid(i,1) = time(1);
    timeid(i,2) = tmpid(1);
    timeid(i,3) = tmpid(end);
end

%% 模拟会员选择过程

border = [113.75,114.5,22.5,22.8; % 深圳
          113.230591,113.432465, 23.063307,23.248917; % 广州
          113.023224,113.230591,22.928042,23.054462]; % 佛山
% 三个发达城区对任务完成与否的影响
pb = [0.7,0.8,0.9];

%% 按照时间段去依次枚举

for i =1:N %枚举不同的时间段

    st = timeid(i,2); ed = timeid(i,3); % 当前时间段在时间有序 vip 序列中的起始位置
    nowvip = vipinc(st:ed,:);
    nowvip = sortrows(nowvip,-6); % 按照信誉度降序排序
    nowvipcnt = size(nowvip,1); % 当前时间段的会员数量

    %% 获得当前人选择订单成功的概率
    p0=zeros(1 , totvipcnt);
    for u = 1 : nowvipcnt

        id1 = nowvip(u,1) ;
        div = 0;
        this = nowvip(u,4);
        for k = 1 : nowvipcnt % 求出所有当前订单 10km 内的当前会员数量

            id2 = nowvip(k,1);
            if id1==id2 || u == k
                continue;
            end
            dist = vipd(id1,id2);
            if dist<=5 % 两个会员之间的距离
                div = div + nowvip(k,4);
            end
        end

        if div == 0
            p0(id1) = 1;
        else
            div = div + this;
            p0(id1) = this / div;
        end
    end
end
end

```

```

%% 当前时间段的所有会员取得订单
for j = 1 : nowvipcnt % 依次枚举当前时间段的所有会员

    vipid = nowvip(j,1); % 当前会员号

    thisviporid = 0; % 初始化当前会员的订单容器
    nowvipor = [0,0,0,0,0,0,0];

    % 生成当前人选择的订单
    for s = 1 : totorent
        if nowor(s , 6) == 1
            continue;
        end

        D = viptoer(vipid , nowor(s , 1));

        if D <= 30 % average
            id = nowor(s,1);
            thisviporid = [thisviporid , id];
        end

    end

    thisviporid(1) = []; % 去除当前会员容器头部
    if isempty(thisviporid) % 当前会员周围无可接订单，直接跳过当前会员
        continue;
    end
    thisvipor = [0,0,0,0,0,0,0];
    len = length(thisviporid); %

    for s = 1:len
        id = thisviporid(s);
        thisvipor = [thisvipor; nowor(id , :)];
    end
    thisvipor(1,:) = [];

    for s = 1:len
        id = thisvipor(s,1);
        D = viptoer(vipinc(j,1),id);
        D = 210 - D;
        thisvipor(s,7) = thisvipor(s , 4) * 1000 + D;
    end

    thisvipor = sortrows(thisvipor , -7); % 订单首先按照价格其次按照距离进行排序
    view = 0;

    last = vipinc(j,4);
    last = min(vipmaxor , last); % 当前人能够抢到的单数

%% 贪心首先选择价格高的和距离近的遍历进行选择
    leng = size(thisvipor,1);
    for q = 1:leng % 遍历所有订单判断距离
        id = thisvipor(q,1);
        if nowor(id,6) == 1 % 当前的订单已经被人选择过
            continue;
        end
    end

```



```

%% 计算抢到当前单的概率
p1 = p0(vipid);% 计算抢到订单的概率
t1 = rand(1);

%% 当前订单能够抢到，计算完成的概率做标记

if(t1 <= p1) % 当前订单被当前会员抢走更新
    nowor(id,6)=1;
    last = last - 1;
    nowvipor = [nowvipor;thisvipor(q , :)];
    if last <=0
        break;
    end
end

nowvipor(1,:) = [];
noworcnt = size(nowvipor,1);

%% 遍历当前人选择的所有订单，计算每个订单的完成概率

for k = 1:noworcnt
    % 根据价格计算一个概率
    orid = nowvipor(k,1);
    x = nowvipor(k,3); y = nowvipor(k,2);
    p23=1;
    for r = 1:3
        if x >= border(r,1) && x <= border(r,2)
            if y >=border(r,3) && y <=border(r,4)
                p23 = pb(r);
            end
        end
    end
    nowprice = oldprice(orid);
    if nowprice <= 70 % 根据价格生成一个完成概率
        p21= 0.7;
    elseif nowprice >70 && nowprice <=75
        p21 = 0.8;
    elseif nowprice >75 && nowprice <=80
        p21= 0.9;
    else
        p21= 0.98;
    end

    % 根据距离计算一个概率
    dis = viptoer(vipid,orid);
    if dis >=25 && dis<30
        p22 = 0.9;
    elseif dis >=20 && dis<25
        p22 = 0.92;
    elseif dis >=15 && dis<20
        p22 = 0.94;
    elseif dis >=10 && dis<=15
        p22 = 0.96;
    else
        p22 = 0.98;
    end
end

```

```

        p2= p21*p22*p23;
        t2=rand(1); % 当前任务完成的概率

        if(t2 <= p2)
            complete(orid) = 1;
        else
            complete(orid) = 0;
        end
    end
end

end
ans = sum(complete)
tmp = finor;
tmp(:,5) = complete;
tmp = sortrows(tmp,5);
idx= find(tmp(:,5) ==1);

sort(idx); idx = idx(1);idx = idx-1;

tmp1 = tmp(1:idx,:);
tmp2 = tmp(idx+1:end,:);
figure(2);

plot(tmp1(:,3),tmp1(:,2),'kx','MarkerSize',7);
hold on;
plot(tmp2(:,3),tmp2(:,2),'o','MarkerSize',5,'MarkerFaceColor',[36,169,255]/255,'MarkerEdgeColor',[36,169,255]/255);
grid on ;
xlabel('经度');ylabel('纬度');
legend('未完成的任务','完成的任务');
end

com = com / 1000
rate = com / 835

```

程序编号	2-4	文件名称	Main2_3.m	程序说明	蒙特卡洛模拟新价格
<pre> clc,clear,close; load data.mat; load newpri.mat; com = 0; % tmp = sortrows(finor,5); % idx= find(tmp(:,5) ==1); % % sort(idx); idx = idx(1);idx = idx-1; % % tmp1 = tmp(1:idx,:); % tmp2 = tmp(idx+1:end,:); % figure(1); % % plot(tmp1(:,3),tmp1(:,2),'kx','MarkerSize',7); % hold on; </pre>					

```

%
plot(tmp2(:,3),tmp2(:,2),'o','MarkerSize',5,'MarkerFaceColor',[36,169,255]/255,'MarkerEdgeColor',[36,169,255]/255);
% grid on ;
% xlabel('经度');ylabel('纬度');
% legend('未完成任务','完成任务');
for round = 1:1000

[vipinc,id] = sortrows(vip,5);
time = vipinc(:,5);
time = unique(time);
finor = new;
timeid = zeros(length(time),2);
N = length(time);
totvipcnt = size(vip,1);
vipmaxor = 30; % 会员可以选取的最大订单数量
R = 57; % 会员可以选择的任务距离半径
totorcnt = size(finor,1);
nowor = finor;
tmp = zeros(totorcnt,1);
nowor=[nowor,tmp];
complete = zeros(1,totorcnt); % 统计最后每个订单的完成情况
oldprice = finor(:,4);

border = [113.759033,114.444305,22.502661,22.758453; % 深圳
113.023224,113.230591,22.928042,23.054462; % 广州
113.230591,113.432465,23.063307,23.248917]; % 佛山
% 三个发达城区对任务完成与否的影响
pb = [0.7,0.8,0.9];
%% 得到同一时间所有会员的起始下标和终止下标
for i=1:N
    tmpid = find(vipinc(:,5) == time(i));
    tmpid = sort(tmpid);
    timeid(i,1) = time(1);
    timeid(i,2) = tmpid(1);
    timeid(i,3) = tmpid(end);
end

%% 模拟会员选择过程

%% 按照时间段去依次枚举

for i = 1:N %枚举不同的时间段

    st = timeid(i,2); ed = timeid(i,3); % 当前时间段在时间有序 vip 序列中的起始位置
    nowvip = vipinc(st:ed,:);
    nowvip = sortrows(nowvip,-6); % 按照信誉度降序排序
    nowvipcnt = size(nowvip,1); % 当前时间段的会员数量

    %% 获得当前人选择订单成功的概率
    p0 = zeros(1 , totvipcnt);
    for u = 1 : nowvipcnt

        id1 = nowvip(u,1) ;
        div = 0;
        this = nowvip(u,4);

```

```

for k = 1 : nowvipcnt % 求出所有当前订单 10km 内的当前会员数量

    id2 = nowvip(k,1);
    if id1==id2 || u == k
        continue;
    end
    dist = vipd(id1,id2);
    if dist<=3 % 两个会员之间的距离
        div = div + nowvip(k,4);
    end
end

if div == 0
    p0(id1) = 1;
else
    div = div + this;
    p0(id1) = this / div;
end
end

%% 当前时间段的所有会员取得订单
for j = 1 : nowvipcnt % 依次枚举当前时间段的所有会员

    vipid = nowvip(j,1); % 当前会员号
    thisviporid = 0; % 初始化当前会员的订单容器
    nowvipor = [0,0,0,0,0,0,0];

    % 生成当前人选择的订单
    for s = 1 : totorcnt
        if nowor(s , 6) == 1
            continue;
        end

        D = viptoer(vipid , nowor(s , 1));

        if D <= 30 % average
            id = nowor(s,1);
            thisviporid = [thisviporid , id];
        end
    end

    thisviporid(1) = []; % 去除当前会员容器头部
    if isempty(thisviporid) % 当前会员周围无可接订单，直接跳过当前会员
        continue;
    end
    thisvipor = [0,0,0,0,0,0];
    len = length(thisviporid); %

    for s = 1:len
        id = thisviporid(s);
        thisvipor = [thisvipor; nowor(id , :)];
    end
    thisvipor(1,:) = [];

    for s = 1:len % 当前会员订单处理后按照价格、距离进行排序
        id = thisvipor(s,1);

```

```

        D = viptoer(vipinc(j,1),id);
        D = 210 - D;
        thisvipor(s,7) = thisvipor(s , 4) * 1000 + D;
    end

    thisvipor = sortrows(thisvipor , -7); % 订单首先按照价格其次按照距离进行排序
    view = 0;

    last = vipinc(j,4);
    last = min(vipmaxor , last); % 当前人能够抢到的单数

    % 贪心首先选择价格高的和距离近的遍历进行选择
    leng = size(thisvipor,1);
    for q = 1:leng % 遍历所有订单判断距离
        id = thisvipor(q,1);
        if nowor(id,6) == 1 % 当前的订单已经被人选择过
            continue;
        end

        % 计算抢到当前单的概率
        p1 = p0(vipid); % 计算抢到订单的概率
        t1 = rand(1);

        %% 当前订单能够抢到，计算完成的概率做标记

        if(t1 <= p1) % 当前订单被当前会员抢走更新
            nowor(id,6)=1;
            last = last - 1;
            nowvipor = [nowvipor;thisvipor(q , :)];
            if last <=0
                break;
            end
        end
    end

    nowvipor(1,:) = [];
    noworcnt = size(nowvipor,1);

    %% 遍历当前人选择的所有订单，计算每个订单的完成概率

    for k = 1:noworcnt
        % 根据价格计算一个概率
        orid = nowvipor(k,1);
        x = nowvipor(k,3); y = nowvipor(k,2);
        p23=1;
        for r = 1:3
            if x >= border(r,1) && x <= border(r,2)
                if y >=border(r,3) && y <=border(r,4)
                    p23 = pb(r);
                end
            end
        end
        nowprice = oldprice(orid);
        if i <=11 && i >=1
            nowprice = nowprice * 1.1;
        elseif i == 31
            nowprice = nowprice * 0.95;
        end
    end

```

```

        if nowprice > 85
            nowprice = 85;
        elseif nowprice < 65
            nowprice = 65;
        end
        if nowprice <= 70    % 根据价格生成一个完成概率
            p21 = 0.7;
        elseif nowprice > 70 && nowprice <= 75
            p21 = 0.8;
        elseif nowprice > 75 && nowprice <= 80
            p21 = 0.9;
        else
            p21 = 0.98;
        end

        % 根据距离计算一个概率
        dis = vptoor(vipid, orid);
        if dis >= 25 && dis < 30
            p22 = 0.9;
        elseif dis >= 20 && dis < 25
            p22 = 0.92;
        elseif dis >= 15 && dis < 20
            p22 = 0.94;
        elseif dis >= 10 && dis <= 15
            p22 = 0.96;
        else
            p22 = 0.98;
        end

        p2 = p21 * p22 * p23;
        t2 = rand(1); % 当前任务完成的概率

        if(t2 <= p2)
            complete(orid) = 1;
        else
            complete(orid) = 0;
        end
    end
end

com = com + sum(complete);

% tmp = finor;
% tmp(:,5) = complete;
% tmp = sortrows(tmp,5);
% idx = find(tmp(:,5) == 1);
%
% sort(idx); idx = idx(1); idx = idx - 1;
%
% tmp1 = tmp(1:idx,:);
% tmp2 = tmp(idx+1:end,:);
% figure(2);

% plot(tmp1(:,3), tmp1(:,2), 'kx', 'MarkerSize', 7);
% hold on;
%
% plot(tmp2(:,3), tmp2(:,2), 'o', 'MarkerSize', 5, 'MarkerFaceColor', [36,169,255]/255, 'MarkerEdgeColor', [36,169,255]/255);

```



```
% grid on ;
% xlabel('经度');ylabel('纬度');
% legend('未完成的任务','完成的任务');
end
```

```
com = com / 1000
rate = com / 835
```

程序编号	3	文件名称	Main3.m	程序说明	按照距离聚类并模拟
<pre>clc,clear,close; load data.mat; load newpri.mat; n = size(finor,1); st = zeros(1,n); finor = new; result = zeros(140,6); [vipinc,id] = sortrows(vip,5); time = vipinc(:,5); time = unique(time); totvipcnt = size(vip,1); timeid = zeros(length(time),2); N = length(time); % 不同时间段的数量 border = [113.759033,114.444305,22.502661,22.758453; % 深圳 113.023224,113.230591,22.928042,23.054462; % 广州 113.230591,113.432465,23.063307,23.248917]; % 佛山 % 三个发达城区对任务完成与否的影响 pb = [0.7,0.8,0.9]; vipmaxor = 30; maxorder = 0; maxfun = 0; com = 0; for times = 1:100 % 枚举不同的聚类情况 st = zeros(n,1); order = [0,0,0,0,0]; % 纬度、经度、价格、数量 %% 对订单按照距离进行聚类 for i = 1 : n if st(i) == 1 continue; end price = finor(i,4); lon = finor(i,3); lat = finor(i,2); id = i; cnt = 1; for j = 1 : n if cnt > 5 % 订单聚类数量超过 break; end if i == j</pre>					

```

        continue;
    end
    if st(j) == 1
        continue;
    end

    if d(i,j) <= 4
        cnt = cnt + 1;
        price = price + finor(j,4);
        st(i) = 1; st(j) = 1;
    end
end
tmp = [id,lon,lat,price,cnt];
order = [order;tmp];
end
% 得到当前聚类完成你的订单
order(1,:) = [];
totcnt = size(order,1);
tmp = zeros(totcnt,1);
order = [order,tmp,tmp];
complete = zeros(1,totcnt); % 统计最后每个订单的完成情况
for i = 1:totcnt
    order(i,7) = i;
end
len = size(order,1);
%% 画图可视化

%% 得到同一时间所有会员的起始下标和终止下标

for i = 1 : N
    tmpid = find(vipinc(:,5) == time(i));
    tmpid = sort(tmpid);
    timeid(i , 1) = time(1);
    timeid(i , 2) = tmpid(1);
    timeid(i , 3) = tmpid(end);
end

%% 求解当前的聚类订单结果

for i = 1:N %枚举不同的时间段

    st = timeid(i,2); ed = timeid(i,3); % 当前时间段在时间有序 vip 序列中的起始位置
    nowvip = vipinc(st:ed,:);
    nowvip = sortrows(nowvip,-6); % 按照信誉度降序排序
    nowvipcnt = size(nowvip,1); % 当前时间段的会员数量
%% 获得当前人选择订单成功的概率
p0 = zeros(1 , totvipcnt);
for u = 1 : nowvipcnt

    id1 = nowvip(u,1) ;
    div = 0;
    this = nowvip(u,4);
    for k = 1 : nowvipcnt % 求出所有当前订单 10km 内的当前会员数量

        id2 = nowvip(k,1);
        if id1==id2 || u == k
            continue;

```

```

end
dist = vipd(id1,id2);
if dist<=3 % 两个会员之间的距离
    div = div + nowvip(k,4);
end
end

if div == 0
    p0(id1) = 1;
else
    div = div + this;
    p0(id1) = this / div;
end
end

%% 当前时间段的会员
for j = 1 : nowvipcnt % 依次枚举当前时间段的所有会员
    vipid = nowvip(j,1); % 当前会员号
    thisviporid = 0; % 初始化当前会员的订单容器
    nowvipor = [0,0,0,0,0,0,0]; % 原 id、经度、纬度、总价格、数量、是否选择、id

    % 生成当前人选择的订单
    viporent = 0;
    for s = 1 : totent
        if order(s, 6) == 1 % 当前聚类的任务已经被人选择
            continue;
        end

        D = viptoer(vipid, order(s, 1)); % 当前的会员到当前缩聚点的距离

        if D <= 30
            id = order(s,7); % 当前聚为一类的代表编号
            thisviporid = [thisviporid, id];
        end
    end

    thisviporid(1) = []; % 去除当前会员容器头部
    if isempty(thisviporid) % 当前会员周围无可接订单，直接跳过当前会员
        continue;
    end

    thisvipor = [0,0,0,0,0,0,0]; % 原 id、经度、纬度、总价格、数量、是否选择、id
    len = length(thisviporid);

    for s = 1:len
        id = thisviporid(s);
        thisvipor = [thisvipor; order(id, :)];
    end

    thisvipor(1,:) = [];

    for s = 1:len % 当前会员订单处理后按照价格、距离进行排序
        id = thisvipor(s,1);
        D = viptoer(vipid,id);
        D = 210 - D;
        thisvipor(s,8) = thisvipor(s, 4) * 10000 + D;
    end
end

```

```

thisvipor = sortrows(thisvipor , -8); % 订单首先按照价格其次按照距离进行排序
thisvipor(:,8) = [];
view = 0;

last = vipinc(j,4);
last = min(vipmaxor , last); % 当前人能够抢到的单数

leng = size(thisvipor,1);

for q = 1:leng % 遍历所有订单判断距离
    id = thisvipor(q,7);
    if order(id,6) == 1 % 当前的订单已经被人选择过
        continue;
    end

    % 计算抢到当前单的概率
    p1 = p0(vipid); % 计算抢到订单的概率
    t1 = rand(1);

    % 当前订单能够抢到，计算完成的概率做标记
    if(t1 <= p1) % 当前订单被当前会员抢走更新
        order(id,6) = 1;
        if last < order(id,5)
            continue;
        end
        last = last - order(id,5);
        nowvipor = [nowvipor;thisvipor(q , :)];
        if last <=0
            break;
        end
    end
end

nowvipor(1,:) = [];
noworcnt = size(nowvipor,1);

%% 遍历当前人选择的所有订单，计算每个订单的完成概率
for k = 1:noworcnt
    % 根据价格计算一个概率
    orid = nowvipor(k,7);
    thiscnt = nowvipor(k,5);
    x = nowvipor(k,2); y = nowvipor(k,3);
    p23=1;
    for r = 1:3
        if x >= border(r,1) && x<= border(r,2)
            if y >=border(r,3) && y<=border(r,4)
                p23 = pb(r);
            end
        end
    end
    nowprice = order(orid,4);
    if i <=11 && i>=1
        nowprice = nowprice * 1.1;
    elseif i == 31
        nowprice = nowprice * 0.95;
    end
    if thiscnt >1 && thiscnt <= 6
        nowprice = nowprice * (1-0.1*(thiscnt-1));
    end
end

```

```

        end
        nowprice = nowprice / thiscnt;
        if nowprice > 85
            nowprice = 85;
        elseif nowprice < 65
            nowprice = 65;
        end
        if nowprice <= 70 % 根据价格生成一个完成概率
            p21 = 0.7;
        elseif nowprice > 70 && nowprice <= 75
            p21 = 0.8;
        elseif nowprice > 75 && nowprice <= 80
            p21 = 0.9;
        else
            p21 = 0.98;
        end
        orid = nowvipor(k,1);
        % 根据距离计算一个概率
        dis = viptoer(vipid,orid);
        if dis >= 25 && dis < 30
            p22 = 0.9;
        elseif dis >= 20 && dis < 25
            p22 = 0.92;
        elseif dis >= 15 && dis < 20
            p22 = 0.94;
        elseif dis >= 10 && dis <= 15
            p22 = 0.96;
        else
            p22 = 0.98;
        end

        p2 = p21 * p22 * p23;
        t2 = rand(1); % 当前任务完成的概率
        orid = nowvipor(k,7);
        if(t2 <= p2)
            complete(orid) = 1;
        else
            complete(orid) = 0;
        end
    end
end

for i = 1:totcnt
    if complete(i) == 1
        com = com + order(i,5);
    end
end

end

figure(1);
plot(finor(:,3),finor(:,2),'o','markersize',8,'markerfacecolor',[36,169,255]/255);
hold on;
plot(order(:,2),order(:,3),'o','markersize',10,'markerfacecolor','r');
hold on;
com = com / 100

```