

赛区评阅编号（由赛区组委会填写）：

2017 年高教社杯全国大学生数学建模竞赛

承 诺 书

我们仔细阅读了《全国大学生数学建模竞赛章程》和《全国大学生数学建模竞赛参赛规则》（以下简称为“竞赛章程和参赛规则”，可从全国大学生数学建模竞赛网站下载）。

我们完全明白，在竞赛开始后参赛队员不能以任何方式（包括电话、电子邮件、网上咨询等）与队外的任何人（包括指导教师）研究、讨论与赛题有关的问题。

我们知道，抄袭别人的成果是违反竞赛章程和参赛规则的，如果引用别人的成果或资料（包括网上资料），必须按照规定的参考文献的表述方式列出，并在正文引用处予以标注。在网上交流和下载他人的论文是严重违规违纪行为。

我们以中国大学生名誉和诚信郑重承诺，严格遵守竞赛章程和参赛规则，以保证竞赛的公正、公平性。如有违反竞赛章程和参赛规则的行为，我们将受到严肃处理。

我们授权全国大学生数学建模竞赛组委会，可将我们的论文以任何形式进行公开展示（包括进行网上公示，在书籍、期刊和其他媒体进行正式或非正式发表等）。

我们参赛选择的题号（从 A/B/C/D 中选择一项填写）： B

我们的报名参赛队号（12 位数字全国统一编号）： B201610061

参赛学校（完整的学校全称，不含院系名）： 南京邮电大学

参赛队员（打印并签名）： 1. 梁展豪

2. 成楚凡

3. 陈 慧

指导教师或指导教师组负责人（打印并签名）： 李 雷

（指导教师签名意味着对参赛队的行为和论文的真实性负责）

日期： 2018 年 8 月 29 日

（请勿改动此页内容和格式。此承诺书打印签名后作为纸质论文的封面，注意电子版论文中不得出现此页。以上内容请仔细核对，如填写错误，论文可能被取消评奖资格。）

赛区评阅编号（由赛区组委会填写）：

2017 年高教社杯全国大学生数学建模竞赛

编 号 专 用 页

赛区评阅记录（可供赛区评阅时使用）：

评 阅 人						
备 注						

送全国评阅统一编号（赛区组委会填写）：

全国评阅随机编号（全国组委会填写）：

（请勿改动此页内容和格式。此编号专用页仅供赛区和全国评阅使用，参赛队打印后装订到纸质论文的第二页上。注意电子版论文中不得出现此页。）

“拍照赚钱”的任务定价

摘要

本文围绕“拍照赚钱”的任务定价问题，建立了基于偏最小二乘回归分析的定价模型，研究了任务定价规律并分析任务未完成的原因；建立了时空效率定价模型和任务匹配规划模型，设计了新的任务定价方案并与原方案比较；建立了模糊 C-均值聚类模型和基于成本和打包规模的定价模型，解决了在任务打包发布的情况下定价模型的修改问题；建立了基于 BP 神经网络的动态定价问题，解决了任务和会员分布不协调情况下的定价问题。

针对问题一，建立了基于偏最小二乘回归分析的定价规律模型，结合 K-means 聚类算法，分别研究了任务定价规律和任务未完成的原因。首先，根据附件利用 K-means 聚类算法对任务整体聚类，得到分类情况离散图，初步定性得到任务分布和会员分布的相关性。再进一步定量分析，建立基于偏最小二乘回归分析的模型，拟合出定价与相关变量的回归方程，分析回归系数大小得到任务定价规律。接着，分析任务未完成的原因时，增加自变量任务价格，改变因变量为任务完成率，回归出任务完成率与各因素的函数，通过系数比较分析其原因。最终得到结论：密集区定价较低，偏远地区定价较高；任务未完成的原因为任务分布与标价和会员分布不合理。

针对问题二，建立了时空效率定价模型和任务匹配规划模型，分别从任务定价和任务匹配方面研究。基于会员选择任务和任务匹配会员的双向关系，从时间、空间和效率三维影响因素着手，首先对会员信誉度进行分级，并对各任务时空高效和时空可抵两类距离构建备选会员集，建立了时空效率定价模型和任务匹配规划模型；考虑任务预订开始时间和任务配额等约束，设计二维多阶段轮盘赌多项式时间算法进行求解。结果显示：与旧方案相比，新方案在任务完成率和任务完成额方面均有显著提升，分别提升了 8.9% 和 15.47%。在进行灵敏度分析时，将信誉分级函数的临界指标变动 2%，观察得到任务完成率也会随之波动，符合实际。

针对问题三，建立基于成本和打包规模的定价模型，结合模糊 C-均值聚类方法，解决了在任务打包发布的情况下定价模型的修改问题。首先，采用模糊 C-均值聚类对任务进行打包，并通过设置距离阈值优化打包效果。其次，以打包后包内的任务数作为定价因素，综合考虑平台效率和成本，建立了基于成本和打包规模的定价模型，对问题二的定价模型进行修改。最后，针对修改后的定价，使用支持向量机的方法预测完成情况。求解结果与问题二相比，任务匹配率提高了 16.9%，已完成的任务降低了 5.03%，说明该方案可以兼顾效率和成本，且更优于原方案。

针对问题四，采用 BP-神经网络模型，对新项目的任务定价给出了动态决策方案。首先，对任务进行聚类 and 打包，得到其分布特点。其次，采取附件一的数据作为 BP-神经网络的训练集，以此预测出任务会员匹配情况下的合理定价。再次建立基于 BP-神经网络的动态价格预测模型，通过将预测数据加入数据集并再次预测，最终得到最优任务定价方案。最后对定价方案进行评估时，沿用问题三中的模型。模型求解结果为：附件三中任务点主要分布在深圳、广州两地，总体任务完成率为 60.5%；由于任务较为密集，所以任务完成率与之前两题相比有所下降。

关键词：K-means 聚类算法、偏最小二乘回归、二维多阶段轮盘赌算法、模糊 C-均值聚类、支持向量机、BP-神经网络

一、 问题重述

1.1 问题的背景

“拍照赚钱”是移动互联网下的一种自助式服务模式。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 问题的提出

根据所给附件，完成下面的问题：

- 1、研究附件一中项目的任务定价规律，分析任务未完成的原因。
- 2、为附件一中的项目设计新的任务定价方案，并和原方案进行比较。
- 3、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，一种考虑是将这些任务联合在一起打包发布。在这种考虑下，如何修改前面的定价模型，对最终的任务完成情况又有什么影响？
- 4、对附件三中的新项目给出你的任务定价方案，并评价该方案的实施效果。

二、问题分析

4.1 问题一的分析

问题一要求针对附件一中的数据，研究出项目的任务定价规律，并分析任务未完成的原因，可以分为两部分进行作答。

针对任务定价规律，主要目的是研究标价与相关数据之间的关系，首先通过绘制任务分布和会员分布的离散情况图，定性分析出任务分布和会员分布的粗略关系。再定量分析项目的任务定价规律，可通过聚类算法^[1]对整体任务分布聚类，划分出不同中心，再根据任务分布密集程度分析定价的异同。在对每个聚集区具体分析时，需要选取合适的指标，使得其能大致反映对价格变动的情况。根据所给数据，本文定义了与任务价格强相关的指标：该地区任务密集程度、任务数、会员数、任务最低价格。针对这个典型的统计分析问题，由于偏最小二乘原理^[1]兼具主成分分析、典型相关分析和线性回归分析等方法的特点，使用它可以很好地分析拟合出定价与各项指标之间的回归方程，进一步分析即可得到项目的任务定价规律。

针对第二部分任务未完成原因的分析，考虑到任务未完成可能与地域有关，所以可以对不同地区分开进行分析。在对每个地区进行分析时，由于任务完成情况是 0-1 变量，因此可以类比任务定价，自定义出任务未完成的回归方程模型，根据其回归系数的大小分析任务未完成的主要原因。

4.2 问题二的分析

问题二需要为附件一中的项目设计新的任务定价方案，并和原方案进行比较。问题二的建立是在问题一的基础上，对原定价方案的改善。首先，需要明确任务与会员的关系，任务的价格、距离等因素会影响会员的接单，价格越高、距离越近，接单率越高；

同时会员的信誉值、位置等因素也会影响任务的匹配，信誉值越高、位置越靠近，匹配率也就越高。这两者相互影响，所以这是一个双向的过程。所以，设计任务定价方案的目的不仅要提高任务的接单率，还要同时使得任务的定价总体提升最小。求解时分两部分进行：一是任务定价，二是任务匹配。针对任务定价，充分考虑信誉度对定价与分配的影响，可以建立时空效率定价模型，从3个维度进行考虑分析，时间维度、空间维度、效率维度，其中时间维度反映会员与任务间的距离影响，空间维度反映任务附近会员聚集情况的影响，效率维度反映考虑信誉度^[5]对定价与分配的影响。针对任务匹配，要基于任务成功执行数最大的原则，对各任务整合时空高效和时空可抵两类距离构建了备选会员集，可引入限制条件：一个任务至多被一个会员成功执行、被同一会员成功执行的任务数不得超过该会员的任务配额、成功匹配的任务不一定被执行，采用任务匹配规划模型进行求解。任务定价方案设计完成后，与原方案进行比较时，可以从任务总定价、任务完成情况两个方面进行分析。

4.3 问题三的分析

对于问题三，由于多个任务可能因为位置比较集中，导致用户会争相选择，一方面，由于任务之间存在定价、位置等因素的差异，会导致会员“挑单”。另一方面，多个任务位置比较集中，如果分派给多人去完成，会造成不必要的人力与资源浪费。因此，考虑将相对位置比较集中的任务一起打包发布，并且制定新的打包定价模型^[2]。

首先是对任务打包的研究，显而易见需要对任务进行聚类分析。考虑到每个坐标没有明确的类别标记，界限不是很分明，但需要对整个地图范围内的任务点进行聚类以求得最优解。而传统聚类方法只能在局部最好最优解，为了能够通过极小化目标函数求得全局最优解，可采取模糊聚类中的模糊C-均值（FCM）进行聚类。聚类后需要对数据进行筛选，当打包中任务间距离太远时，会员对于任务的选择会产生矛盾，降低任务选择率和完成率；当打包中任务距离中心太近时，很容易出现局部集聚现象，而使打包失去意义，综合考虑后可以设置距离阈值，对数据进行筛选。

接着是新的打包定价模型的制定，基于实际生活中一般大包装商品比小包装商品的平均单价略低的原则，可以建立基于成本和打包规模的定价模型。根据理论，推导出任务定价与相对数量间的函数关系，之后采用最小二乘法确定出各参数值。为了使定价模型能有效解决该问题，需要事先对该模型进行评估，可使用支持向量机^[7]的方法，对新定价方案的完成情况进行预测，根据预测结果而后采用确定新定价方案，对最终的任务完成情况进行综合分析。

4.4 问题四的分析

问题四要求对附件三中的新项目制定任务定价方案，并评价该方案的实施效果。首先需要对附件三中任务的分布进行分析，根据分布图得知任务聚集度很高。其次，在制定任务定价方案时，略高或略低的价格都会影响会员的积极性，从而影响整个任务的完成率，而仅仅按照实时任务完成程度进行统一定价的方法也是较为片面的。因此，本题的大致思路即利用大数据机器学习的思想，根据对历史数据的学习训练预测出新的样本值，通过判断该预测价格是否能调动会员的参与积极性评价该定价的合理性，并不断进行动态修正与完善。综合分析，可采取BP-神经网络对任务定价进行动态确定。对于BP-神经网络需要的训练集，可以使用附件一的数据，对任务的基本情况进行训练，并以此为基础预测任务会员匹配情况下的合理定价。之后将BP-神经网络^[4]应用于附件三的数据时，考虑到附件三中任务密集度较大，可以沿用问题三建立的任务打包模型，先将任务集打包，在将预测数据加入数据集并再次预测，最终得到最优任务定价方案。由于求

解过程中，对任务采取打包的方式，所以最后对该方案的实施效果进行评价时，同样可沿用问题三中的模型进行评价

三、模型假设

- 1、任务难度相同；
- 2、本题定价方案不考虑任务的难易程度、竞争因素；
- 3、假设后续问题中会员的分布不发生变化；
- 4、假设每个任务被匹配后仅有“成功执行”和“未执行”两种状态，任务匹配后因“未执行”不再重新返回平台进行再匹配；
- 5、假设会员选择任务时，不受外界因素干扰，只受其信誉度、开始预订时间和配额确定的选择概率影响。

四、符号说明

x_1 : 任务密集程度
 x_2 : 该区域任务数
 x_3 : 该区域会员数
 y : 该区域的平均定价
 α_0 : 平台参考价格
 P : 该区域任务完成的概率
 r : 区域半径

D_{ij} : 在时空可抵范围内会员 i 与任务 j 的距离

D'_i : D_{ij} 的均值方差处理值

q_i^B : 会员 i 的信誉值

Q_i^B : 会员信誉分段函数

p : 聚类中心编号

j : 任务编号

r_1 : 时空高效范围

r_2 : 时空可抵范围

t_i^B : 任务开始预订时间

c_i^B : 预订配额

G_j^{up} : 可抵会员集

G_j^{low} : 高效会员集

G_j : 备选会员集

F_j : 每个任务对应的匹配解

Pr: 会员选中任务的概率

Po_j : 平台原有价格

五、模型的建立与求解

5.1 问题一的模型建立和求解

在研究任务定价规律时，通过定量定性相结合的方法进行分析。在定量分析时，通过观察任务分布和会员分布离散图得到两者的相关性；在定性分析时，建立偏最小二乘回归方差实现。在分析任务未完成的原因时，类比于任务定价回归方程，自定义出任务未完成的回归方程模型，进行求解。

5.1.1 模型的建立

5.1.1.1 定价影响因素相关性分析——定性研究定价规律

根据附件一的项目任务的经纬度数据，画出任务的分布图如下：

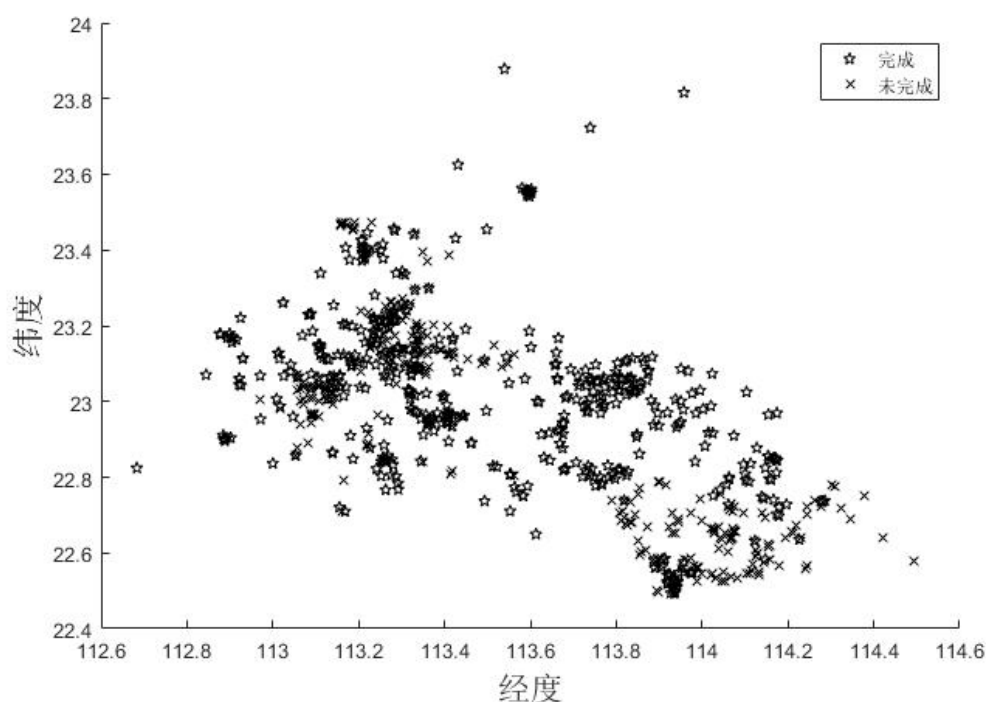


图 5.1-1 项目任务分布图

由图可以得到，已完成的任务比较集中，多集中在 $22.6^{\circ}\text{N}\sim 23.6^{\circ}\text{N}$ ， $113^{\circ}\text{E}\sim 113.5^{\circ}\text{E}$ 和 $22.5^{\circ}\text{N}\sim 22.8^{\circ}\text{N}$ ， $113.8^{\circ}\text{E}\sim 114.3^{\circ}\text{E}$ 两个地区；而未完成任务的散点分布较零散，比较集中分布在 $23^{\circ}\text{N}\sim 23.2^{\circ}\text{N}$ ， $113.6^{\circ}\text{E}\sim 113.8^{\circ}\text{E}$ 。

根据附件二的会员位置信息数据，画出会员的分布图如下：

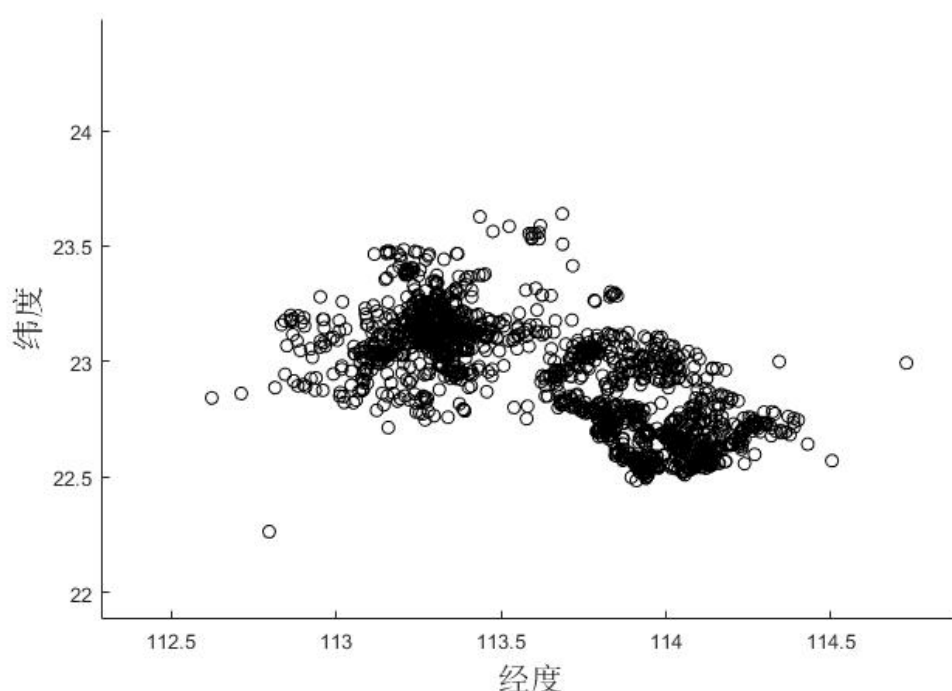


图 5.1-2 会员位置分布图

由图可以得到，会员的分布非常集中，基本集中分布在 22.8°N~23.5°N，113°E~113.5°E 和 22.5°N~23.2°N，113.6°E~114.3°E

综合比较两张分布图，可以大致看出：在会员分布比较密集的地方，任务派发也比较多；而在会员分布比较稀疏的地方，任务派发的比较少。

5.1.1.2 基于 K-Means 聚类的偏最小二乘回归模型——定量分析定价规律

由定性分析可知，任务位置大致集中于几块地域，可用聚类分析^[1]进一步对任务集中区域进行分类，继而对每个聚类的内部进行分析，从而寻求出各地的任务定价规律。

1、K-Means 聚类

首先，需要对数据进行预处理，去除一些离奇点

K-means 聚类算法被广泛的应用于配送网络中的区域的划分分析中，其聚类的目标是尽可能的使各聚类中的末端点相互紧凑，并尽可能使各聚类间相互分开。

其基本过程如下：

设 n 个数据样本为 $C = \{C_1, C_2, C_3 \dots C_n\}$ ， $k(2 \leq k \leq n)$ 是要将数据样本分成的类型的数目，即 C_k 表示第 k 个类簇。K-Means 算法以距离作为数据对象间相似性度量的标准，通常采用欧氏距离来计算数据对象间的距离。欧式距离的计算公式：

$$\text{dist}(x_i, x_j) = \sqrt{\sum_{d=1}^D (x_{i,d} - x_{j,d})^2} \quad (5.1.1)$$

其中， D 表示数据对象的属性个数。

K-Means 算法聚类过程中，每次迭代，对应的类簇中心需要重新计算（更新）：对应类簇中所有数据对象的均值，即为更新后该类簇的类簇中心。定义第 k 个类簇的类簇中心为 Center_k ，则类簇中心更新方式如下：

$$Center_k = \frac{1}{|C_k|} \sum_{x_i \in C_k} x_i \quad (5.1.2)$$

其中, C_k 表示第 k 个类簇, $|C_k|$ 表示第 k 个类簇中数据对象的个数, 这里的求和是指类簇 C_k 中所有元素在每列属性上的和, 因此 $Center_k$ 也是一个含有 D 个属性的向量, 表示为

$$Center_k = (Center_k, 1, Center_k, 2, \dots, Center_k, D) \quad (5.1.3)$$

K-Means 算法需要不断地迭代来重新划分类簇, 并更新类簇中心。一般情况, 有两种方法来终止迭代: 一种方法是设定迭代次数 T , 当到达第 T 次迭代, 则终止迭代, 此时所得类簇即为最终聚类结果; 另一种方法是采用误差平方和准则函数, 函数模型如下:

$$J = \sum_{k=1}^k \sum_{x_i \in C_k} d_{ist}(x_i, Center_k) \quad (5.1.4)$$

其中, K 表示类簇个数。当两次迭代 J 的差值小于某一阈值时, 即 $\Delta J < \delta \Delta J < \delta$ 时, 则终止迭代, 此时所得类簇即为最终聚类结果。

具体算法步骤为:

- (1) 从 n 个数据对象任意选择 k 个对象作为初始聚类中心;
- (2) 根据每个聚类对象的均值 (中心对象), 计算每个对象与这些中心对象的距离; 并根据最小距离重新对相应对象进行划分;
- (3) 重新计算每个 (有变化) 聚类的均值 (中心对象);
- (4) 计算标准测度函数, 当满足一定条件, 如函数收敛时, 则算法终止; 如果条件不满足则回到步骤 (2)。

这种划分使得下式最少:

$$E = \sum_{j=1}^k \sum_{x_i \in \omega_j} \|x_i - m_j\|^2 \quad (5.1.5)$$

其中 m_j 为各类的中心。

2、偏最小二乘回归分析

根据资料显示, 偏最小二乘回归^[1]分析方法具有主成分分析、典型相关分析和线性回归分析等方法的特点, 能有效解决变量存在多重共线性的问题。所以, 本文选用该方法对平台任务定价及其影响因子的相关关系进行定量拟合, 为定价规律的研究提供理论支持。

Step1: 自变量确定

根据价格市场的供需关系, 众包平台的任务定价要实际考虑到各个方面, 包括: 任务发布企业的收益和会员接受任务的成本, 其相对位置是达成两者间协调的主要因素。因此, 本文确定三项指标来研究价格变化:

- ① 任务密集程度=该区任务数/该市总任务数 x_1 , 反映某区域任务分布情况。
- ② 该区域任务数 x_2
- ③ 该区域会员数 x_3

由任务数和会员数的相互制约, 综合考虑出某区域会员密度和任务密度的相关性。

Step2: 因变量确定

考虑到任务和会员的分配存在很大的差异性, 本文使用规定区域的平均定价来作为

研究对象，从而使模型更加具有普遍性，记作 y 。

Step3: 观测 n 个样本点，偏最小二乘回归在 X 中提取出成分 X_i ，为了满足回归分析的需要，要求：

- (1) x_i 应尽可能多地携带它们各自数据表中的变异信息；
- (2) x_i 和 y 的相关程度能够达到最大。

Step4: 在第一个成分被提取后，偏最小二乘回归分别实施 X 对 X_i 的回归以及 Y 对 X_i 的回归。如果回归方程已经达到满意的精度，则算法终止；

Step5: 否则，利用 X 被 X_i 解释后的残余信息以及 Y 被 X_i 解释后的残余信息进行第二轮的成分提取。如此往复，直到能达到一个较满意的精度为止。

Step6: 最终得到标准化偏最小二乘回归方程

$$y = \alpha_0 + \alpha_1 x_1 + \alpha_2 x_2 + \alpha_3 x_3 \quad (5.1.5)$$

其中， α_0 定义为平台参考价格， x_1, x_2, x_3 分别为归一化后的任务密集程度、某区任务数、某区会员数。 $\alpha_1, \alpha_2, \alpha_3$ 分别表示三个指标的系数。

由回归过程，该方程能最大程度的满足某城市各密集程度不同的区域的实际情况，由此可以精准得出各区域的定价规律。

5.1.1.3 任务完成率模型

在数据表格中，任务是否完成，即表示为简单的多重 0-1 分布：

$$Y = \begin{cases} 1 & \text{任务完成} \\ 0 & \text{任务没有完成} \end{cases} \quad (5.1.6)$$

取划分好的几个聚集区域，对该地内部的三个任务密集程度不一样的区域进行回归分析。定义该地第 i 区的任务完成的概率为：

$$P = \frac{\text{任务完成数}}{\text{该区域任务总数}} \quad (5.1.7)$$

利用 5.1.1.2 中的偏最小二乘回归模型，此时，研究的是任务完成率与各因素之间的关系，所以设置因变量为某区域任务完成概率 P ，考虑到任务完成率同样和前文中的三个自变量有关，所以保留自变量 x_1, x_2, x_3 ，增加该区域平均价格 y ，以此综合考虑各项指标对于任务是否完成的影响。

通过回归分析，得出回归函数：

$$P = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 y \quad (5.1.8)$$

其中， β_0 定义为函数实参， x_1, x_2, x_3, y 分别为归一化后的任务密集程度、某区任务数、某区会员数、某区任务平均价格。 $\beta_1, \beta_2, \beta_3, \beta_4$ 分别表示四项指标的系数。最后根据这四项指标的系数值，评价出各指标对于任务完成概率的影响。

5.1.2 模型的求解

5.1.2.1 任务定价规律分析

1、聚类结果处理

由前期定性分析图可知，当 $n=4$ 时可以很好的进行分裂，由此通过 MATLAB 使用 K-Means 聚类分析求解得到聚类中心 (23.14°N, 113.32°E)，(22.65, 114.04°E)，(22.98°N, 113.76°E)，(23.03°N, 113.11°E)。具体分类范围如下图所示：

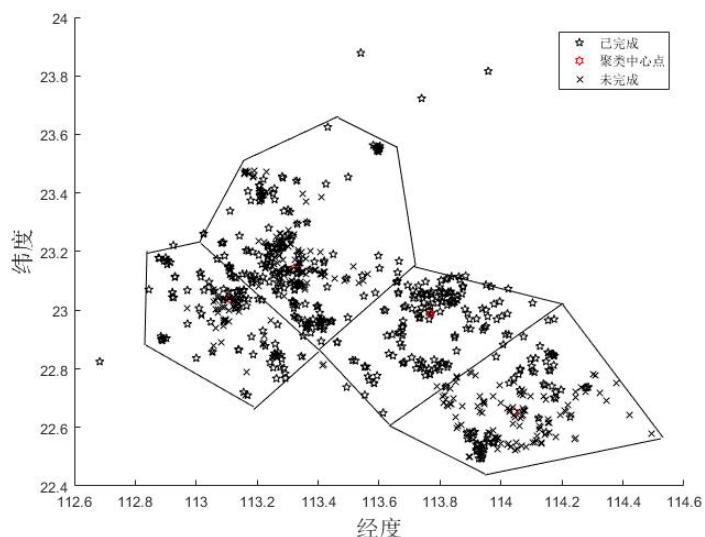


图 5.1-3 任务位置聚类分析图

因此，可将附件一中的所有任务分为四类，以每类中心城市作为标志。由四类区域的大概经纬度范围，得这 4 个区域分别位于广州、深圳、佛山、东莞。

在知道中心点之后，将各个任务点的经纬度转换为距离各自中心的距离，研究距离和定价之间的关系。

利用 MATLAB 进行拟合，得到的拟合图如下：

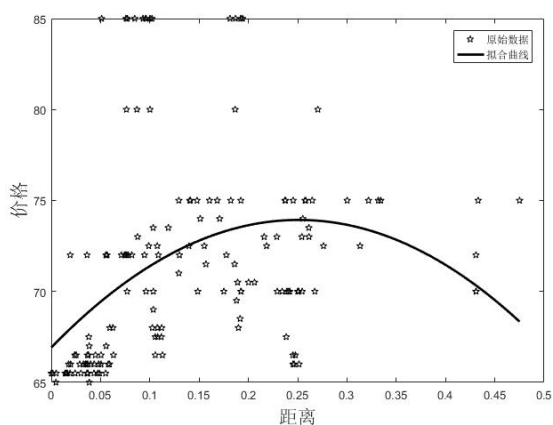


图 5.1-4 广州的定价与距离的关系

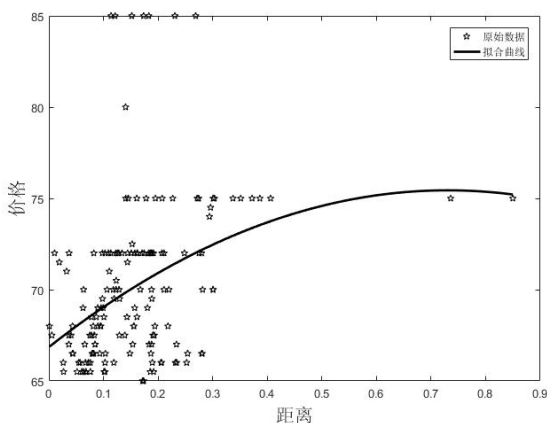


图 5.1-5 深圳的任务定价与距离的关系

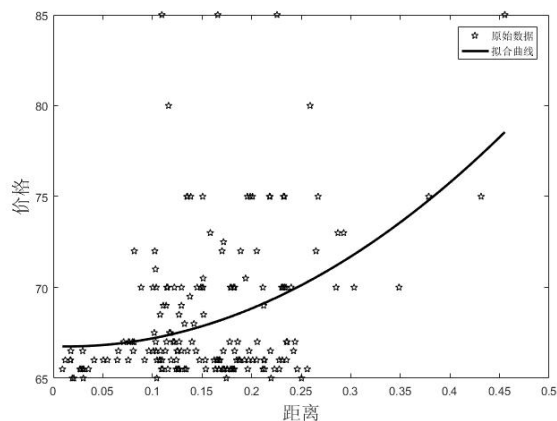


图 5.1-6 东莞的定价与距离的关系

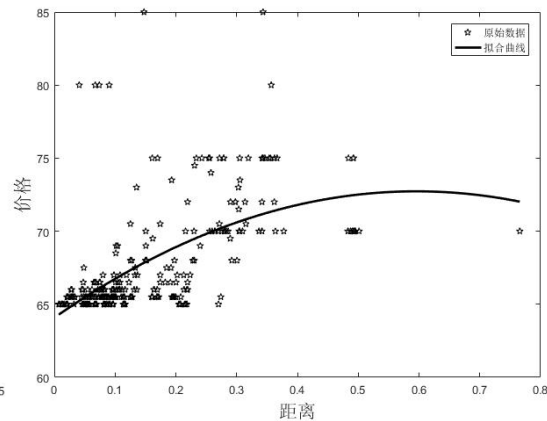


图 5.1-7 佛山的定价与距离的关系

通过拟合图可以看出，这四个函数具有很大的共性：定价和到各自中心的距离呈正相关。将四类的中心点坐标代入到谷歌地图当中，发现这四个点分别在：广州市天河体育中心、深圳福田、东莞南城行政中心、佛山市的祖庙。这四点分别是四个市的市中心，也是经济发展的中心。基于此我们发现距离市中心越近的地点，定价越高，距离市中心越远，越偏远的地区，定价越低，可见计算的结果也与事实相符。

2、偏最小二乘回归参数确定

由图 5.1-3 不难看出，每个聚类区域的任务分布不是均匀的，存在类中心任务密集，四周任务稀疏的现象。因此，本文将每一个城市分为 3 个区域，分别为任务密集区、一般区和稀疏区。其中密集区为以每一个聚类中心为圆心，半径 $r=d/3$ 的圆域；一般区为半径 $r=d/3 \sim 2d/3$ 的圆环；稀疏区即为该城市其他区域的点。在聚类基础上，得出四个城市各区域的任务密集程度 x_1 、该区域任务数 x_2 、该区域会员数 x_3 、平均价格 y 、任务完成率 P ，此处以广州为例，其余数据见附录

表 5.1-1 广州地区各指标统计情况

广州				
指标 区域	任务数量	平均价格	完成率	会员数
密集区	236	66.68	0.65	614
一般区	68	71.88	0.75	83
稀疏区	1	70	1	0

根据统计数据，得到任务平均定价与各影响因素之间的系数如下表所示：

表 5.1-2 任务平均定价回归方程中的各系数

任务密集程度	任务数	会员数	固定常数
-3.09439	-26.3188	4.668	73.37775

由此可以得到任务平均价格和各影响因素之间的函数关系：

$$y = -3.09439x_1 - 26.3188x_2 + 4.668x_3 + 73.37775$$

由回归函数可以得知，任务定价与区域任务数负相关，与该区域会员数正相关，说明任务越多，出现供过于求，当地任务价格越低；会员数越多，出现供少于求，各任务发放者为了吸引会员完成任务，发布的价格会提高。

5.1.2.2 任务未完成原因分析

将统计数据中的任务完成率 P 作为偏最小二乘回归拟合的因变量自变量 x_1, x_2, x_3 ，增加该区域平均价格 y

根据统计数据，得到任务完成率 P 与各影响因素之间的系数如下表所示：

表 5.1-3 任务完成率 P 与各影响因素之间的系数

任务密集程度	任务数	会员数	平均价格	固定常数
0.747625	-0.58977	-0.49059	7.92542	0.02909

由此可以得到任务完成率 P 与各影响因素之间的函数关系：

$$P = 0.747625x_1 - 0.58977x_2 - 0.49059x_3 + 7.92542y + 0.02909$$

根据拟合出的函数，可以看出任务完成的概率和任务数、会员数都负相关，和任务

密集程度、任务发布价格正相关。而其中，任务密集程度，任务和会员的相对分布情况的参数取值都很小，价格参数很大。由此可知，任务未完成，可能因为任务多的地方，注册会员也会很多，而价格指数对会员选择任务有很大影响，所以对于那些价格低的任务，就长时间得不到完成。

5.1.3 问题的结论

对任务定价规律进行研究，定量分析得到任务定价与距离相关性较强；定性分析得到其与任务密集程度、区域任务数、会员数和平台参考价格的回归方程：

$$y = -3.09439x_1 - 26.3188x_2 + 4.668x_3 + 73.37775$$

该回归方程表明：任务定价与区域任务数负相关，与该区域会员数正相关。

分析任务未完成的原因时，求解得到任务完成率与任务密集程度、任务数、会员数和任务平均价格的回归方程：

$$P = 0.747625x_1 - 0.58977x_2 - 0.49059x_3 + 7.92542y + 0.02909$$

通过方程系数大小可以看出，任务完成的概率和任务数、会员数都负相关，和任务密集程度、任务发布价格正相关。

5.2 问题二的模型建立与求解

新任务定价方案的制定从任务定价和任务匹配两方面着手，对于任务定价，从时间、空间和效率 3 个维度进行分析，充分考虑距离、会员聚集情况、信誉度^[6]的影响，建立时空效率定价模型进行求解；对于任务匹配，对各任务整合时空高效和时空可抵两类距离构建备选会员集，引入任务匹配规划模型，以任务总定价提升比例最小和任务成功执行数最大为目标，进行求解。

5.2.1 模型的建立

“拍照赚钱”商业模式本质为会员和任务间的一种匹配过程。从宏观条件下分析，这种匹配呈现“多对一”的特征，将影响因素按匹配主体双方分类，进行的两个过程分析：

(1) 会员选择任务：任务定价越高，距离越近，会员越乐意选择；

(2) 任务匹配会员：会员信誉度越高，一定空间内会员聚集数量越多，任务被执行概率越高。

分析单个任务点，周围会员分布情况很大程度上影响了该任务的执行效率。聚集的高信誉度、高配额会员数量越多，任务被执行的概率越大；反之则概率越小。参照匹配时空规范化距离概念，为了使任务能够尽可能的被会员执行，平台在进行任务匹配时，会先匹配距离较近的部分会员，在其没有接单后，再考虑匹配较远会员。因此，匹配时需对高效距离和可抵距离分类讨论。

5.2.1.1 时空效率定价模型

设计新的定价方案，建立时空效率定价模型，主要从以下三个维度，全面考虑了定价影响因素：

(1) 时间维度：会员与任务间的距离影响，距离影响会员抵达任务点的时间，从而影响完成率。

(2) 空间维度：任务附近会员聚集情况的影响，聚集的高信誉度会员越多，完成率越大。

(3) 效率维度：模型充分考虑信誉度对定价与分配的影响，算法中以任务开始时间和配额取代。

首先，对距离 D_{ij} 和信誉度 q_i^B 进行数据规范化处理，以规避影响定价的各类指标量纲间复杂的相关性和极端数据对定价模型计算带来的不良影响。

D_{ij} 为在时空可抵范围内会员 i 与任务 j 的距离， D'_i 为 D_{ij} 的均值方差处理值：

$$D'_i = \frac{D_{ij} - \overline{D_{ij}}}{\sigma_{ij}}, \quad i \in G_j, \quad j \in A \quad (5.2.1)$$

$$\overline{D_{ij}} = \frac{1}{|G_j|} \sum_{i=1}^{|G_j|} D_{ij}, \quad \sigma_{ij} = \sqrt{\frac{1}{|G_j| - 1} \sum_{i=1}^{|G_j|} (D_{ij} - \overline{D_{ij}})^2}, \quad j \in A \quad (5.2.2)$$

q_i^B 为会员 i 的信誉值，依据在线支付信誉评价标准，将附件二中各会员按其信誉度值划分为高信誉、中信誉和低信誉三个等级^[5]，得如图所示的信誉趋势，并量化得会员信誉分段函数 Q_i^B ：

$$Q_i^B = \begin{cases} 1.1, & q_i^B > 19.9231 \\ 1, & q_i^B = 19.9231 \\ 0.9, & q_i^B < 19.9231 \end{cases} \quad (5.2.3)$$

由此，构建包含任务会员时间距离、空间备选会员集和会员信誉度的时空效率定价模型：

$$P_{pj} = \frac{1}{2} K_p P_{oj} \left(H_1 \sum_{m=1}^{|G_j^{low}|} \frac{D'_{mj}}{Q_m^B} + H_2 \sum_{m=1}^{|G_j^{up}|} \frac{D'_{nj}}{Q_n^B} \right) \quad (5.2.4)$$

式中， p 为聚类中心编号， $p = 1, 2, 3, \dots, |C|$ ； j 为任务编号， $j = 1, 2, 3, \dots, |B|$ ； H_1 和 H_2 分别为时空高效范围 $r_1(j)$ 及时空可抵范围 $r_2(j)$ 的会员信息系数值，分别取值1.1和0.9。执行时空效率定价时，任务 j 先匹配 $r_1(j)$ 范围内的会员，再匹配 $r_1(j)$ 和 $r_2(j)$ 范围间的会员，故 $H_1 > H_2$ 。

根据问题一任务未完成情况归因可知，经济发展水平较高的地区（如广州、深圳）任务未完成的概率远大于经济发展水平较低的地区（如东莞）。因此式（5.2.4）中设置经济发展系数 K_p 量化宏观经济因素。由此，可以通过调节定价 K_p 提高经济发达地区任务基准定价，刺激当地会员的积极性，促使其更好地完成任务。据《2018年广东省各市GDP与人均GDP报告》^[8]赋予表5.1-1所示的系数值。

表5.1-1 各聚类中心经济发展系数赋值

聚类中心点	佛山市	广州市	东莞市	深圳市
K_p	1.02	1.13	0.97	1.19

5.2.1.2 任务匹配规划模型

基于时空效率定价模型重新定价后，参考竞争成功选择问题相关规律，构建以平台任务总定价提升比例最小和任务成功执行数最大为目标的会员任务匹配规划模型。

$$\min \left(\sum_{i=1}^{|B|} \sum_{j=1}^{|A|} z_{ij} \frac{P_{pj}}{P_{oj}} - \sum_{j=1}^m \sum_{i=1}^n z_{ij} \right) \quad (5.2.5)$$

$$\text{s. t.} \begin{cases} \sum_{i=1}^{|B|} z_{ij} \leq 1, j = 1, 2, \dots, |A| \\ \sum_{j=1}^{|A|} z_{ij} \leq c_i^B, i = 1, 2, \dots, |B| \\ \Pr(F_j = b_i) \geq \Pr(z_{ij} = 1), i = 1, 2, \dots, |B|, j = 1, 2, \dots, |A| \end{cases} \quad (5.2.6)$$

目标函数（5.2.5）分为两部分，前部为在 APP 平台原始定价基础上成功执行任务的平台总定价，后部为任务被成功执行次数和；约束（2）为一个任务至多被一个会员成功执行；约束（3）为被同一会员成功执行的任务数不得超过该会员的任务配额；约束（4）表示成功匹配的任务不一定被执行。

5.2.1.3 二维多阶段轮盘赌求解算法

针对时空效率定价模型的任务匹配问题特征，我们设计了基于时空高效距离 r_1 和时空可抵距离 r_2 的“二维多阶段轮盘赌”任务会员匹配算法。对于模型中涉及会员信誉度 q_i^B 的约束，算法中可通过题设条件“参考其信誉给出的任务开始预订时间和预订配额”中的任务开始预订时间 t_i^B 和预订配额 c_i^B 作归约处理。由于距离约束受 r_1 和 r_2 的限制，每个任务 j 备选会员集 G_j 的规模将有效减小，并划分为可抵会员集 G_j^{up} 与高效会员集 G_j^{low} ，其中 $G_j^{up} \cap G_j^{low} = \emptyset$ 且 $G_j^{up} \cup G_j^{low} = G_j$ ，与全局遍历算法相比，显著降低了计算的时间复杂性。由于 G_j 元素从所有会员中遍历获得，在多阶段轮盘赌过程中，

末段决策各会员被选概率之和为 1，因此算例规模下，“二维多阶段轮盘赌”算法属于精确解法。

（1）主体算法执行步骤

对包含任务编号、地理位置、任务定价的任务集 $A\{a_j, (x_j^A, y_j^A), P_j\}$ 和包涵会员编号、地理位置、配额、任务开始时间、信誉度的会员集 $B\{b_j, (x_j^B, y_j^B), c_i^B, t_i^B, q_i^B\}$ 进行匹配，获得每个任务对应的匹配解 $F_j\{a_j, b_i\}$ 。如某任务下 $b_i = NaN$ ，则决策变量 $z_{ji} = 0$ ；否则 $z_{ji} = 1$ 为使得会员备选集中至少有一个高信誉度会员，计算时空高效距离：

$$r_1(j) = \min\{ D_{ij} \}, i = 1, 2, \dots, |B| \text{ and } q_i^B > 19.9231 \quad (5.2.7)$$

为保证每个任务至少有一个备选会员，以悲观准则计算时空可抵距离：

算法具体步骤如下：

Step1: 初始化 $j = 1$;

Step2: 选中 a_j , 初始化 $i = 1$;

Step3: 选中 b_i , 若 $D_{ij} \leq \min\{r_1(j), r_2(j)\}$, 则将 b_i 存入 G_j^{low} , $i = i + 1$; 否则, $i = i + 1$;

Step4: 判断 $i > |B|$? 若是, 进入Step5; 否则, 返回Step3;

Step5: 调用BET得 F_j^{low} , 若 $z_{ji} = 1$ 则 $F_j = F_j^{low}$, $c_i^B = c_i^B - 1$, $j = j + 1$, 进入Step9; 否则Step6;

Step6: 选中 b_i , 若 $D_{ij} \leq \max\{r_1(j), r_2(j)\}$, 则将 b_i 存入 G_j^{up} , $i = i + 1$, 否则 $i = i + 1$;

Step7: 判断 $i > |B|$? 若是, 进入Step8; 否则, 返回Step8;

Step8: 调用BET得 F_j^{up} , $F_j = F_j^{up}$, $j = j + 1$;

Step9: 判断 $i > |A|$? 若是, 结束匹配并输出所有 F_j ; 否则, 返回Step2。

(2) 多阶段轮盘赌规则

本规则在充分考虑“具有同一任务开始预订时间的会员属于相同的优先批次”和“同批次内各会员选中任务概率依预订配额占比决定”两种情况下, 设计批次内完全信息共享任务匹配规则。

分析 G_j 各元素 $g_{jn} (b_{jn}, x_{b_{jn}}^B, y_{b_{jn}}^B, c_{b_{jn}}^B, t_{b_{jn}}^B, q_{b_{jn}}^B)$, $n = 1, 2, \dots, |G_j|$, $b_{jn} \in B$ 。将会员划分为

$$G_j^{T(1)}, G_j^{T(2)}, \dots, G_j^{T(m)}, \bigcup_{i=1}^{i=m} G_j^{T(i)} = G_j \quad (5.2.8)$$

其中相同集合内的会员具有相同的 t_i^B , 即 $\forall i \in G_j^{T(m)}, T(m) = t_i^B$ 。

基于提出的新定价标准, 每个会员选中任务的概率与新标准价格 P_{pj} 和平台原有价格 P_{oj} 的比值成正比。通过式 (5.2.9) 可计算会员 b_{jn} 在集合 G_j 中选中任务 j 的概率

$\Pr(b_{jn})$:

$$\Pr(b_{jn}) = \frac{c_{b_{jn}}^B}{\sum_{n=1}^{n=|G_j|} c_{b_{jn}}^B} \times \frac{P_{pj}}{P_{oj}} \quad (5.2.9)$$

由此, 针对每一批次 $T(m)$, 均有式 (5.2.10) 所示的未选中概率 $\Delta Pr(m)$:

$$\Delta Pr(m) = 1 - \sum_{k=1}^{k=m} \sum_{n=1}^{n=|T_m|} \Pr(b_{jn}^k) \quad (5.2.10)$$

5.2.2 模型的求解

用二维多阶段轮盘赌求解算法可以求得 835 个任务点的任务的时空效率定价及成功执行的匹配会员编号，并可以计算出新旧方案的相关数据如下：

表 5.2-2 新旧方案的相关数据

	未执行	完成率	总定价	总花费
旧方案	313	0.62515	57708	36076.14
新方案	239	0.713772	58363	41657.9

由上表可以读出旧方案未执行数量为 313 个，新方案的未执行数量为 239 个，减少 23.7%；而完成率由旧方案的 62.5%提高到 71.4%，说明时空效率定价模型相比原定价策略有 8.9%的优化效果，一定程度上提高了匹配成功的概率。

在平台总定价方面，旧方案的总定价为 57708 元，新方案的总定价为 58363 元，增长了 1.14%；旧方案已完成的任务花费为 36076.14 元，新方案为 41657.9 元，增长了 15.47%。

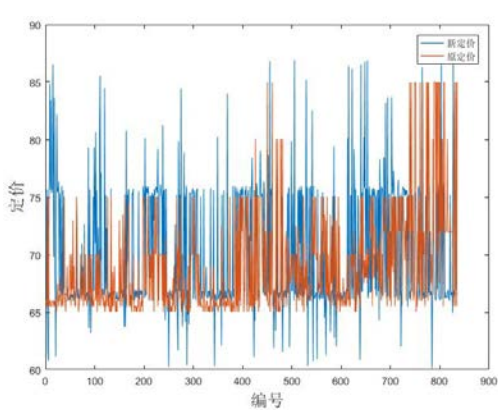


图 5.3-1 定价对比图

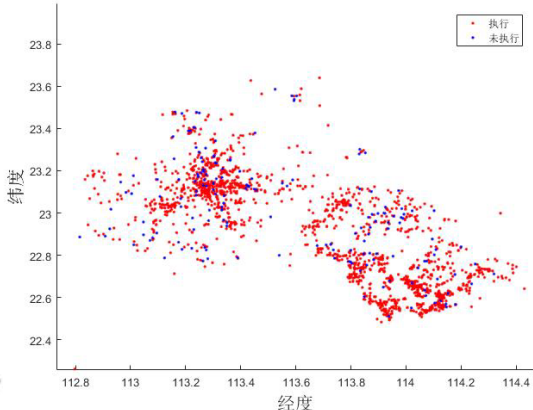


图 5.2-2 新方案任务执行图

此外，分析上图中的全局定价折线图，可发现时空效率定价波动性更大，这是由于新方案的考虑因素更多。而将问题一中的任务完成地理分布情况与新方案任务执行图（蓝点为未执行任务，红点为已执行任务）进行对比，可发现广州市区与深圳市区的未完成任务数显著减少，可能由新策略根据地区经济情况提升基准定价导致；佛山市原方案中未完成任务几乎都被新方案完成；东莞市新出现零星未完成任务，属合理概率事件。

综上所述，新设计的时空效率定价模型能在附件 2 会员规模的情况下，有效提高附件 1 中任务的成功执行概率，但需要付出小幅上升的总体定价，用于激励经济较发达的大城市会员执行任务。

5.2.3 灵敏度分析

将上述模型中信誉分级函数的临界指标变动2%，得到模型结果变化如下：

表5. 2-3 模型结果的变化

	未执行	完成率	平均定价	平均花费
新方案	239	0.713772	69.89	49.88
指标提升	283	0.6611	69.91	46.21
指标下降	22	0.9737	69.90	68.06

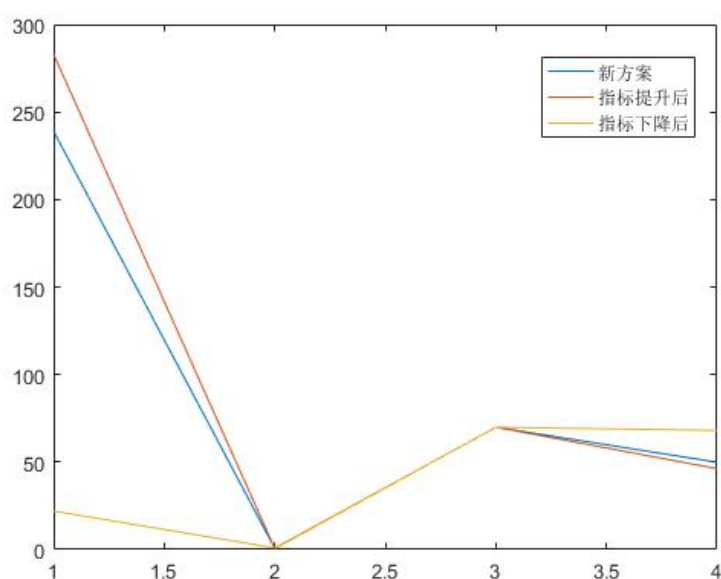


图5. 2-3 灵敏度分析

由上表可知，当指标限制提升时完成率下降，总花费也有所下降，这是由于提升信誉阈值后任务的可选用户数量减少，相反，当指标限制下降后，可选用户显著增加，完成率明显提高，带来的后果是平台支出也大幅提高，可见该模型的用户评级阈值起到很关键的分级作用，综合考虑了完成率和支出。

5. 2. 4 问题的结论

通过时空效率定价模型和任务匹配规划模型，设计了新的任务定价方案。与旧方案相比较，在任务完成率和任务完成额方面均有所提升，在任务未执行数量方面显著下降。其中，任务完成率由旧方案的62.5%提高到71.4%，任务完成额由旧方案的36076.14元提高到41657.9元，任务未执行数量从313个下降到239个，效果显著。

5. 3 问题三的模型建立和求解

对于任务打包，采用模糊聚类中的模糊 C-均值进行聚类，可以通过极小化目标函数求得全局最优解，而后设置距离阈值对打包后的数据进行清洗。由于任务打包后任务数增加，同时考虑平台效率和平台成本，引入基于成本和打包规模的定价模型，对原模型进行修正。为了寻求合理解，可以使用支持向量机的方法对模型结果进行预

测。

5.3.1 模型的建立

5.3.1.1 任务打包

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，从而降低所有任务的完成率。由此对于位置相对集中的任务，进行打包，作为整体任务，并给予每个整体相应的指标值，从而判断定价的合理性。

具体打包步骤如下：

步骤一：对任务点进行聚类，考虑到每个坐标没有明确的类别标记，界限不是很分明，而又需要对整个地图范围内的任务点进行聚类以求得最优解。因此，这里我们采用模糊聚类中的模糊 C-均值（FCM）进行聚类^[2]，相比于传统聚类方法只能在局部最好最优解，该方法能够通过极小化目标函数求得全局最优解。

其一般化形式为：

$$J(U, c_1, \dots, c_c) = \sum_{i=1}^c J_i = \sum_{i=1}^c \sum_{j=1}^n u_{ij}^m d_{ij}^2 \quad (5.3.1)$$

u_{ij} 介于0, 1间； c_i 为模糊组i的聚类中心， $d_{ij} = \|c_i - x_j\|$ 为第i个聚类中心与第j个数据点间的欧几里得距离；且 $m \in [1, \infty)$ 是一个加权指数。

构造如下新的目标函数，可求得使（1）式达到最小值的必要条件

$$\begin{aligned} \bar{J}(U, c_1, \dots, c_c, U, \lambda_1, \dots, \lambda_n) \\ &= J(U, c_1, \dots, c_c) + \sum_{j=1}^n \lambda_j \left(\sum_{i=1}^c u_{ij} - 1 \right) \\ &= \sum_{i=1}^c \sum_{j=1}^n u_{ij}^m d_{ij}^2 + \sum_{j=1}^n \lambda_j \left(\sum_{i=1}^c u_{ij} - 1 \right) \end{aligned} \quad (5.3.2)$$

由此将附件一种所有任务分为c类，每一类作为一个打包整体。

步骤二：对每一个类进行干预，遍历类中任务点j到聚类中心的距离，设置距离阈值 $R_1 \sim R_2$ ，只有所有任务点到中心点的距离在 $R_1 \sim R_2$ 之间的类，才最终确定为一个打包集合，因为当打包中任务间距离太远时，会员对于任务的选择会产生矛盾，降低任务选择率和完成率；当打包中任务距离中心太近时，很容易出现局部集聚现象，而使打包失去意义。对于不符合条件的点，即剔除出原先包裹，作为一个单独任务进行分配。

5.3.1.2 基于成本和打包规模的定价模型

1、模型准备

实际生活中，一般大包装（数量多）商品比小包装（数量少）商品的平均单价（售价与包装内商品数量的比）略低，即当商品数量按一定比例增加时，商品售价不是同比例增加的，相对单价也不是同比例降低的，这无疑是商家的一种价格策略。

因此，本文用数学方法定量计算任务打包后的平均价格和包裹中任务数量之间的函数关系，建立基于成本和打包规模的定价模型。定义任务包总定价 $F(x)$ ，满足以下两个条件：

- ① 包裹中任务平均定价 $\frac{F(x)}{x}$ 关于x严格单调下降，且 $\lim_{x \rightarrow \infty} \frac{F(x)}{x} = Cp$

② 商品利润函数 $R(x) = F(x) - Cxp$, 关于 x 严格单调上升。

其中, x 为任务包中聚类得到的任务数, C 为相对单独任务的相对销售成本系数, Cp 为相对单包装商品的销售成本, $R(x)$ 为包装内任务数量为 x 时的商品利润函数。

2、模型理论推导

根据模型满足的条件推测商品售价与相对数量 x 的函数关系^[3]为

$$F(x) = [C + (A + Bx)^{-\alpha}]px \quad (5.3.3)$$

其中, A, B, C, α 为待定系数。

当(5.3.3)满足条件①时, 有:

$$\left(\frac{F(x)}{x}\right)' = [Cp + (A + Bx)^{-\alpha}p]' = [p(A + Bx)^{-\alpha}]' = (-\alpha pB)[(A + Bx)^{-1-\alpha}] < 0$$

$$\lim_{x \rightarrow +\infty} \left(\frac{F(x)}{x} - Cp\right) = \lim_{x \rightarrow +\infty} [p(A + Bx)^{-\alpha}] = 0$$

而 $(A + Bx)^{-1-\alpha} > 0$, $p > 0$, $B > 0$, 从而 $\alpha > 0$

当(5.3.3)满足条件②时, 有:

$$R' = (F(x) - Cxp)' = [(A + Bx)^{-\alpha}xp]' = (A + Bx)^{-\alpha}p - \alpha px(A + Bx)^{-1-\alpha} = p(A + Bx)^{-\alpha}[1 - \alpha x(A + Bx)^{-1}] > 0$$

而 $p(A + Bx)^{-\alpha} > 0$, 从而 $1 - \alpha x(A + Bx)^{-1} > 0$, 故 $\alpha \leq 1$

综上 $0 < \alpha \leq 1$

3、参数确定

定义任务包内相对数量为 x_i , 相对单价为 y_i , 任务单独发布时实际数量为 x'_i , 实际售价为 y'_i , 由此可知:

$$\text{任务包内相对数量: } x_i = \frac{x'_i}{2}, \text{ 相对任务单价为: } y_i = \frac{y'_i}{px_i}.$$

统计数据, 得到相对数量 x_i , 相对单价 y_i 之间的数量值, 利用该数据对待定参数进行估计:

将相对单价变形为

$$(y - C)^{-\frac{1}{\alpha}} = A + Bx \quad (5.3.4)$$

令 $u = (y - C)^{-\frac{1}{\alpha}}$, 则式(5.3.4)为

$$u = A + Bx \quad (5.3.5)$$

显然, 成本 C 必小于最小的相对销售单价且大于0, 而 $0 < \alpha \leq 1$, 即对在范围内给定的 (C, α) 就可以得到一个确定的 u 。在MATLAB中, 利用最小二乘法, 进行 u 和 x 的线形拟合, 以 x_i , u_i 的相似度的最大值为目标函数, 在需要精度下搜索, 即可确定出参数值。

5.3.1.3 基于支持向量机^[7]的任务完成度预测

为了将打包后的定价方案与原定价方案进行比较，还需预测新定价方案的完成情况。选取指标变量分别为每个任务的位置（经度 x_1 、纬度 x_2 ）、任务价格 x_3 、该任务R范围内会员数目 x_4 、会员平均信誉值 x_5 ，对所有指标数据进行如下标准化处理：

$$\tilde{x}_{ij} = \frac{x_{ij} - \mu_j}{s_j}, i = 1, \dots, n, j = 1, 2, 3, 4, 5 \quad (5.3.6)$$

式中： $j = 1, 2, 3, 4, 5$ 分别代表五个指标； μ_j 为第 j 个指标的均值； s_j 为第 j 个指标的标准差。

记标准化后已分类的任务行向量为 $b_i = [\tilde{a}_{i1}, \dots, \tilde{a}_{i5}]$, $i = 1, \dots, n$ 。用线性内核函数的支持向量机模型进行分类，线性分类函数为：

$$c(\tilde{x}) = \sum_i \beta_i K(b_i, \tilde{x}) + b \quad (5.3.7)$$

当 $c(\tilde{x}) \geq 0$ ， \tilde{x} 属于第一类，即任务可以被完成；当 $c(\tilde{x}) \leq 0$ ， \tilde{x} 属于第二类，即任务无法完成。

利用附件一和附件二提供的实际观测数据集对支持向量机进行训练，随后得出任务包总定价，计算出平均任务定价，以平均任务定价作为预测集，用训练过的模型得到预测的完成情况。根据完成情况，计算出整体的完成率，和问题一、二中的整体完成率进行比较。

5.3.2 模型的求解

根据建立的模型，求解时按照任务打包、聚类筛选数据、模型建立、模型效果预测、任务完成情况评估的顺序依次展开：

Step1: 确定打包规模 N ，并给出打包规模的上界。通过供需平衡数学模型计算可得当 N 趋于无穷时， P_n 收敛于 4，说明当打包规模分布在 4 时，是最为合理的，并将上界定为 6， $N \in [3, 6]$ 。

Step2: 确定同一类中的所有点到聚类中心 p 的距离总和 d_p 。在会员执行任务时，过大的距离往往会超出会员的承受能力，造成任务执行率偏低。因此，我们将 d_p 的上界定为 2km。

Step3: 通过 N 和 d_p 的值确定打包方案。得到打包后的效果图如下（白点为包点，蓝点为原任务点，括号内为任务数量）：

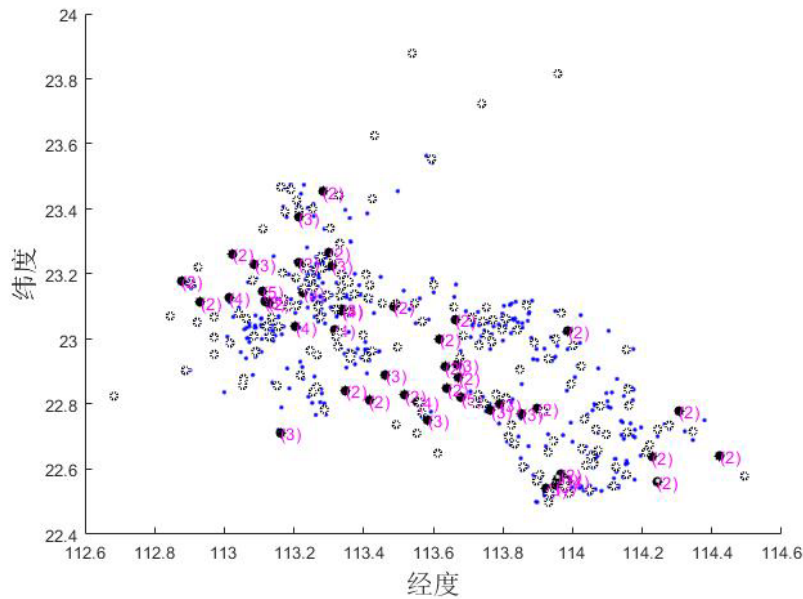


图 5.3-1 打包点的分布及任务数量

通过对打包方案图的分析，可以得到以下结论：

- 1、打包后的任务点分布有明显的集群效应；
- 2、未参与打包的任务点主要分布在偏远地区；
- 3、打包后的任务规模以 2-4 个任务为主。

Step4: 用基于成本和规模的函数模型计算出每个包的总价和包内任务的平均价格，并用第二问时空效率定价模型每个任务的位置（经度 x_1 、纬度 x_2 ）、任务价格 x_3 、该任务 R 范围内会员数目 x_4 、会员平均信誉值 x_5 生成训练集，用基于支持向量机的任务完成度预测打包后任务的完成情况。部分数据如下（详见附录）：

表 5.1-1 部分打包任务的数据

包点纬度	包点经度	包总价	包内均价	原价格	是否完成	包内的任务编号		
23.10	113.45	186.96	62.32	70.5	1	148	149	153
22.82	113.51	136.61	68.30	74	1	198	390	
23.43	113.42	212.15	70.71	80	1	169	401	426
22.75	113.58	185.63	61.87	70	0	178	207	209
23.08	113.96	190.93	63.64	72	1	619	773	774
23.26	113.02	138.46	69.23	75	1	544	737	

Step5: 如果任务在包内则用任务点和会员进行匹配，否则单独进行匹配。

一共 718 个任务中，匹配成功的共有 634 个，配对成功率为 88.3%。比问题二中 71.4% 的匹配成功率高出 16.9%。说明新的打包定价方案优于问题二中的定价方案。在平台总定价方面，问题二中的定价方案所有任务为 58363 元，新方案为 51592.89 减少了 11.6%；原方案为已完成的任務花费 41657.9 元，新方案为 39562.7 元，降低了 5.03%，新方案收益较原方案更好。

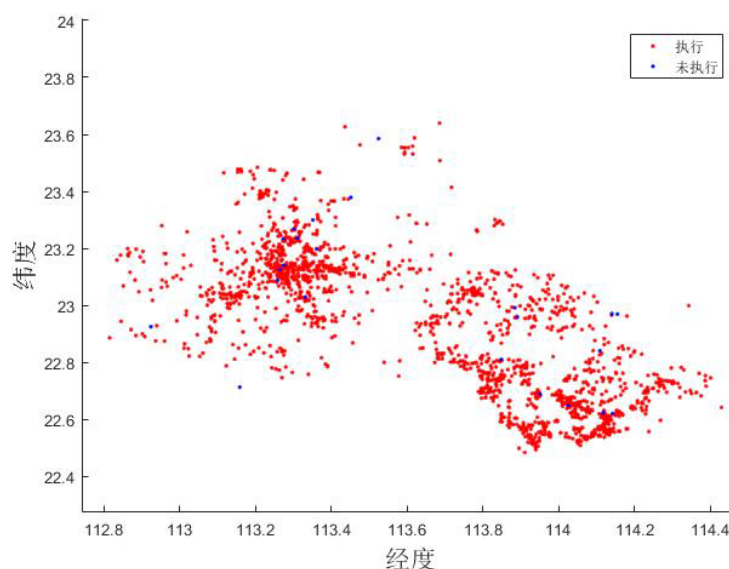


图 5.1-2 打包后的任务执行图

综上可以得出打包后任务比较密集的市区内的未执行任务数量明显减少，未执行的任务主要分布在偏远地区或者任务稀疏的地区。实施打包方案后匹配率提高而且降低了任务花费，主要是因为该方案降低了任务密集区域的未匹配度，且打包后的任务单价下降，一定程度上既提高了匹配度又减少了平台支出。

5.3.3 问题结论

当打包规模分布为4时，最为合理。通过进一步的模型求解，得到的任务完成情况与原方案相比，任务匹配成功率从71.4%提升到88.3%，已完成的任务成本从41657.9元降到39562.7元，说明该方案可以兼顾效率和成本，且更优于原方案。

5.4 问题四的模型建立和求解

在制定任务定价方案时，略高或略低的价格都会影响会员的积极性，从而影响整个任务的完成率，而仅仅按照实时任务完成程度进行统一定价的方法也是较为片面的，采用BP-神经网络动态对其进行决策，可以很好解决这个问题。首先使用附件一的数据，对任务的基本情况进行训练，预测出任务会员匹配情况下的合理定价，之后再将其应用于附件三的数据，得到最优定价方案。

5.4.1 模型建立

1、任务分布情况分析

问题四要求根据附件三中的新任务分布情况，设计出新任务定价方案。我们将附件三任务的位置坐标导入MATLAB，得到任务直观分布图，其中，红色为新任务分布，黑色为会员分布。

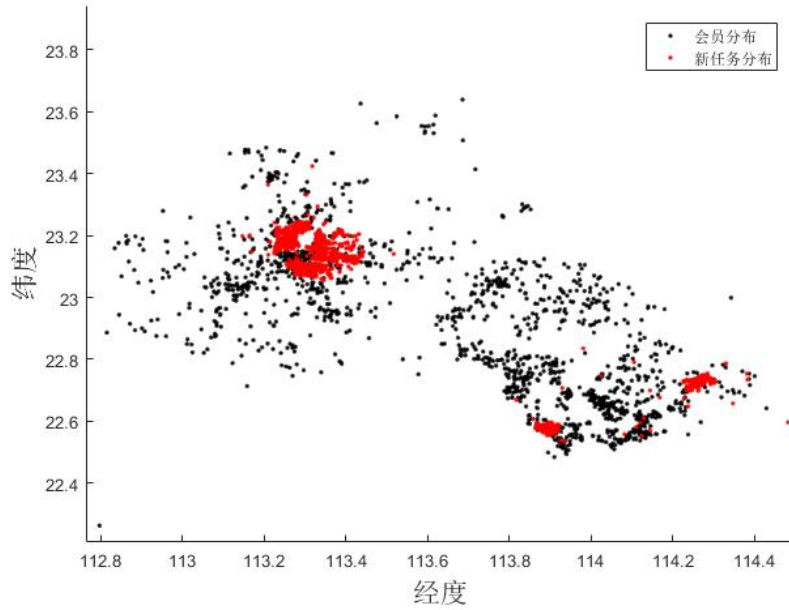


图5.4-1 任务直观分布图

观察分布图，得到如下特点：

- ① 任务主要呈现聚集状分布，由经纬度范围，主要位于广州和深圳两个城市。和附件一中任务分布情况有明显差别。
- ② 在任务聚集区，任务数量众多，与该区附近的会员数目明显不符，会出现供过于求的现象；在会员聚集区，会员分布面广，而周边没有任务，会出现供少于求的现象。

2、基于任务匹配和BP-神经网络的动态定价模型

根据以上特点，本文选择对任务聚集区和任务空闲区的会员实行动态定价的原则。通过问题二的任务匹配规划模型，借助二维多阶段轮盘赌求解算法，在任务位置、会员位置、会员信誉度值，得出所有任务与会员的匹配情况。

接着，本文使用BP-神经网络对问题二中得到的附件一任务的基本情况进行训练，用训练后的神经网络预测出此时任务会员匹配情况下的合理定价，具体方案如下：

在BP神经网络中，包括输入层、隐藏层以及输出层，其中：

输入层为模型二中得到的附件一任务中任务与会员匹配后，所得到的任务与会员间的距离 d_{ij} 和会员信誉度值 p_i 。

隐藏层为多层，其中，神经元的数学模型为：

$$a = f\left(\sum_{i=1}^q \omega_i + b_j\right) \quad (5.4.1)$$

在神经网络中，输入层与输出层的激活函数是由数据处理的不同需求决定的，本文隐藏层的激活函数选择为 S 型函数：

$$f(x) = \frac{1}{1 + e^{-x}} \quad (5.4.2)$$

目标层为任务与会员一一匹配之后，考虑到距离和效益情况下的任务定价 y_j 。

下图即为简化BP神经网络结构图：

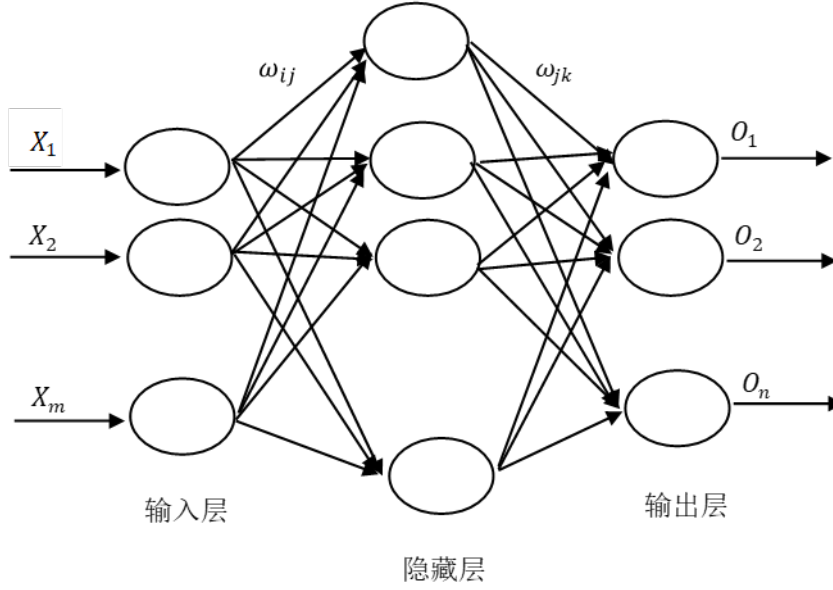


图5.4-2 简化BP神经网络结构图

BP网络^[4]是沿着误差函数减小最快的方向，即梯度的反方向改变权值和偏差。BP算法的迭代计算公式为：

$$x_{k+1} = x_k - a_k g_k \quad (5.4.3)$$

式中： x_k 为当前的权值和偏差， x_{k+1} 为迭代产生的权值和偏差， g_k 为当前误差函数的梯度， a_k 作为学习速率。

BP 网络前向传播计算，即：

$$\text{net}_{ij} = \sum_{k=1}^{N_{i-1}} O_{(i-1)j} \omega_{(i-1)jk} \quad (5.4.4)$$

输出为：

$$O_{ij} = f(\text{net}_{ij}) = \frac{1}{1 + \exp[-(\text{net}_{ij} - \theta_{ij})]} \quad (5.4.5)$$

式中：误差定义为：

$$E = \frac{1}{2} \sum_j e_j^2 = \frac{1}{2} \sum_j (d_j - y_j)^2 \quad (5.4.6)$$

通过附件一数据对模型进行训练，由此得出附件三新任务分配下的任务合理定价。

3、打包任务评价定价效果

采用问题三模型，通过模糊 C-均值（FCM）进行聚类，人工干预出一些特殊点，得到合理打包规模。再基于成本和打包规模的定价模型，对包裹任务进行二次定价，在训练得出的价格基础上进行计算。最后通过支持向量机预测出任务完成情况，和附件一中完成情况对比分析。

5.4.2 模型的求解

1、任务会员匹配

利用任务会员匹配规划模型，得到新任务的会员匹配情况，选取部分匹配见下表：

表 5.4.2-1 任务会员匹配表

	编号									
任务编号	1	2	3	4	5	6	7	8	9	10
会员编号	977	239	80	321	730	375	46	100	145	161

2、BP 神经网络预测初始定价方案

首先由定价关于距离和会员信誉的关系 BP 神经网络的框架，输入层为 2，输出层为 1，隐含层为 3。计算附录一和二中的数据，将任务和会员进行时空匹配和效率匹配，将距离和信誉作为输入量，价格作为输出量进行 BP 神经网络训练得到训练后的价格测试值，结果显示与原价格契合度非常高。

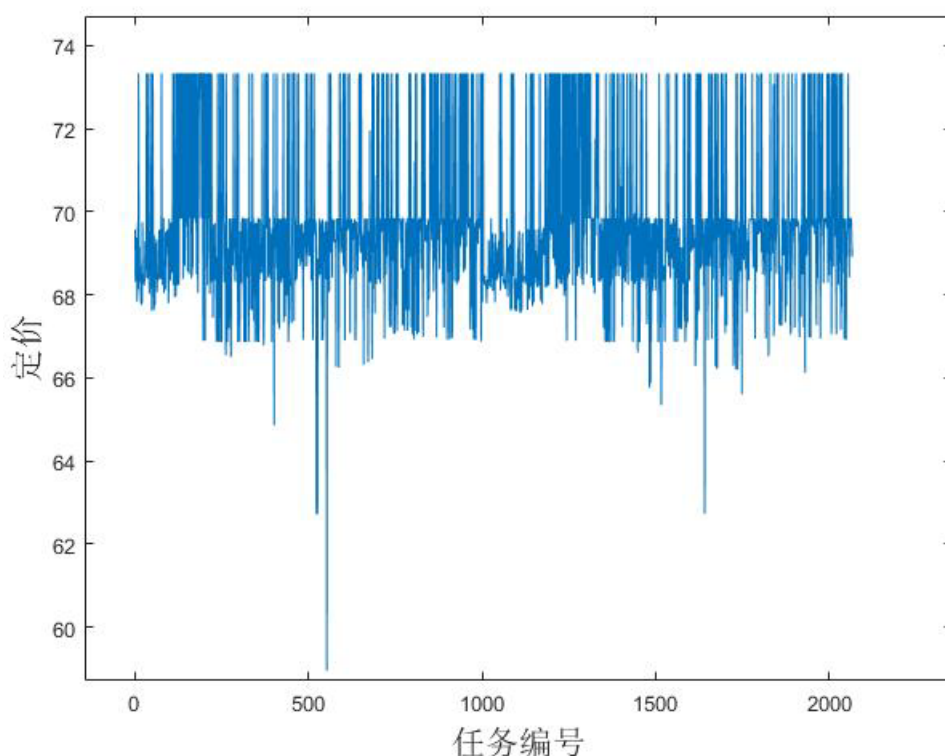


图 5.4.2-1 BP 神经网络预测任务价格

3、确定规模并打包任务点，提高效率

考虑到任务密集度较大，引用问题三建立的任务打包模型，设定距离阈值为 100m，将距离阈值以内的任务集“打包发布”。在打包过程中，将任务集数量过大（如 43、33、21 个任务聚集）的任务集随机分成若干个任务集，保证每个任务集的数量不超过 8 个。根据问题三中模型对任务包的处理方式重新分配打包任务的价格。打包点任务分布为（其中黑色星型为打包点，蓝色实心点为任务）：

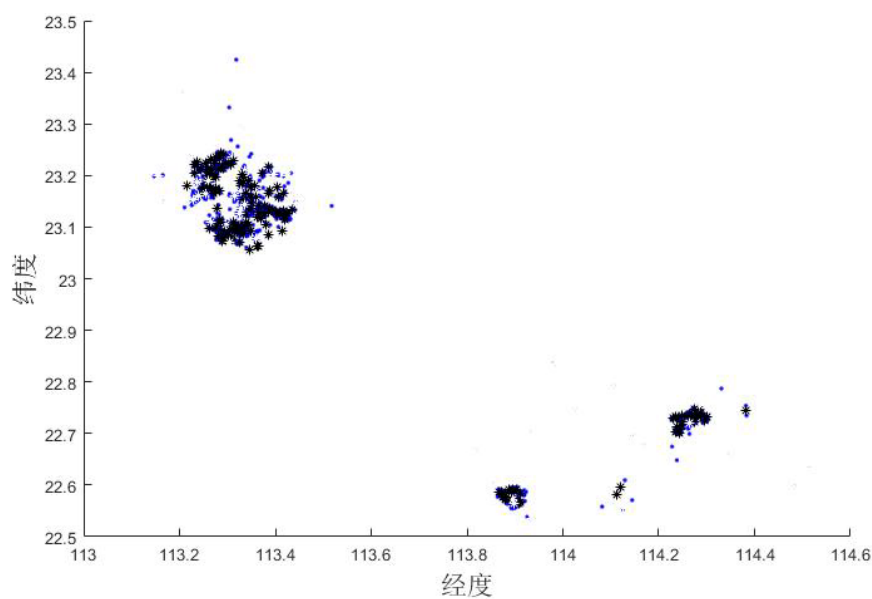


图 5.4.2-2 打包点任务分布

(3) 预测完成情况

利用问题二建立的支持向量机模型，预测此方案的效果（见附件），部分数据如下：

表 5.4.2-2 包内任务完成情况及包内任务编号

包点纬度	包点经度	包内均价	是否完成	包内任务编号
22.71887	114.2658	52.09359	0	5 10 42 47 51 54 84 1012
23.07718	113.3586	56.39699	1	476 478 513 517 525 539 628 630
23.21199	113.282	56.66895	1	874 875 879 880 882 883 895 896
23.2025	113.3294	55.49848	1	267 750 756 1828 1838 1876 1902
23.12688	113.4158	55.18078	1	292 340 385 1344 1402 1406 1493

问题四的任务点主要分布在深圳、广州两地，在此价格方案之下，由于任务较为密集，与之前两题相比有所下降，总体任务完成率为 60.5%，但比原方案的 47.9%高出 12.6%，一定程度上能提高匹配率。

六、模型评价

6.1 模型的优点

1、问题一中的 K-means 聚类算法避免了其他算法处理大数据集时的不准确性，得到了较为精准合理的聚类效果，按城市不同采取不同的定价方案，较符合实际情况下的定价规律；

2、本文创新的根据题目提出了“时空效率定价模型”，充分全面地考虑了定价影响因素考虑全面，而且定价稳定，适用于模型；

3、对于任务打包的情况，采用基于成本和打包规模的定价模型，能够平衡平台的效率和平台的成本；

6.2 模型的缺点

1、K-means 聚类算法中的 K 值是给定的，无法确定最优分类数；

2、“时空效率定价模型”在大规模下的应用有一定限制；

3、对于任务打包时，仅仅考虑了成本和规模的影响，没有考虑会员的信誉值，可能造成当打包发布的任务未被执行时，平台承受的损失会比较大；

6.3 模型的推广

本文建立的定价模型有效的提高了在原有定价规律下，任务被成功执行的概率，并创新性的“提出二维多阶段轮盘赌”算法，该算法运行效率高，准确性高，能够有效的解决移动互联网匹配效率平台，大大的降低了平台的实时性和运营成本。

参考文献

- [1] 马明德, 马学娟, 谢应忠, 马 甜, 宁夏生态足迹影响因子的偏最小二乘回归分析[J], 生态学报, 第 34 卷第 3 期, 2014. 2: 682-688。
- [2] 冯剑红, 李国良, 冯建华, 众包技术研究综述[J], 计算机学报, 第 38 卷第 9 期, 2015. 9: 1713-1725。
- [3] 付丽娜, 李宝毅, 张 驰, 商品售价与包装内商品数量的函数关系探讨[J], 天津师范大学学报(自然科学版), 第 31 卷第 3 期, 2011. 7: 17-21。
- [4] 庄育锋, 胡晓瑾, 翟 宇, 基于BP神经网络的微量药品动态称重系统非线性补偿[J], 仪器仪表学报, 第35卷第8期, 2014. 8: 1914-1920。
- [5] 王宝丽, 杨琦峰. 在线支付下个人信用等级评价指标体系[J], 中国水运:理论版, 2006. 4:170-171。
- [6] 乐琦, 樊治平. 基于累积前景理论的双边匹配决策方法[J]. 系统工程学报, 2013. 1:38-46。
- [7] Sarah M. Erfani, Sutharshan Rajasegarar, Shanika Karunasekera, Christopher Leckie High-Dimensional and Large-Scale Anomaly Detection using a Linear One-Class SVM with Deep Leaning[J]. Pattern Recognition, 2016
- [8] 《2016 年广东省各市GDP 与人均GDP 报告》
http://www.360doc.com/content/17/0419/14/502486_646832037.shtml

附录

```
%距离和价格
x=dis4(:,1);
y=dis4(:,4);
%plot(x,y);
%a=polyval(x,y,2)
p=polyfit(x,y,2) % 拟合出的二次函数的系数
ye=y-polyval(p,x); % 计算误差
ye2s=sum(ye.^2); % 误差的平方和
disp(sprintf('误差的平方和=%d',ye2s));
xx=linspace(min(x),max(x)); % 绘图用到的点的横坐标
yy=polyval(p,xx); % 拟合曲线的纵坐标
plot(x,y,'p',xx,yy); % 绘图，原始数据+拟合曲线
legend('原始数据','拟合曲线'); % 图示
%s=char(vpa(poly2sym(p,'x'),5)); % 二次函数式转换为字符串，vpa 转换小数，保留 5
位有效数字
%title(['y=' s]);
xlabel('距离') %x 坐标的描述
ylabel('价格') %y 坐标的描述
```

%findcustomer 用于在问题 3 中对会员分组%

```
function group=findcustomer(task,cust)
[a,~]=size(cust);
index=1:a;
index=index';
cust=[index,cust];%会员矩阵加编号
customer=sortrows(cust,3);%按经度排列
[n,~]=size(task);
group=zeros(n,6);
for i=1:n
SSS=find(customer(:,3)<task(i,2));
p1=max(find(customer(:,3)<task(i,2)));
p2=min(find(customer(:,3)>task(i,2)));
%返回行号
for j=1:3
%存入标号
if ~isempty(p1)
group(i,4-j)=customer(p1+1-j,1);
end
if ~isempty(p2)
group(i,3+j)=customer(p2-1+j,1);
end
end
```

```
end
end
end
```

%doublesystem 用于在问题 3 中在每组会员中匹配一个任务%

```
function man=doublesystem(group,customer)
```

```
 [~,b]=size(group);
```

```
 sum=0;
```

```
 g=zeros();
```

```
 range=zeros(2,1);
```

```
 if(b==1)
```

```
   man=group(1);
```

```
 else
```

```
   %尝试所有的选择
```

```
   for i=1:b
```

```
     if(customer(group(i),3)<=405)
```

```
       g(1,i)=group(i);
```

```
     elseif(customer(group(i),3)<=420)
```

```
       g(2,i)=group(i);
```

```
     elseif(customer(group(i),3)<=435)
```

```
       g(3,i)=group(i);
```

```
     elseif(customer(group(i),3)<=450)
```

```
       g(4,i)=group(i);
```

```
     elseif(customer(group(i),3)<=465)
```

```
       g(5,i)=group(i);
```

```
     else
```

```
       g(6,i)=group(i);
```

```
     end
```

```
     sum=customer(group(i),4)+sum;%求和
```

```
   end %分组
```

```
 [c,~]=size(g);
```

```
 for i=1:c%分组进盘
```

```
   temp=g(i,:);%第 i 组
```

```
   temp((temp==0))=[];%去 0;
```

```
   [~,m]=size(temp);
```

```
   for j=1:m%i 组进盘
```

```
     if(range(1,1)==0)
```

```
       range(1,1)=customer(temp(j),4)/sum;
```

```
       range(2,1)=temp(j);
```

```
     else
```

```
       endnum=range(1,end);
```

```
       per=(customer(temp(j),4)/sum);
```

```

nu=per+endnum;
range=[range,[nu;temp(j)]];
end
end
ra=rand(1,1);
if(ra<range(1,end))
[~,index]=size(range);
while(index>=1&&ra<range(1,index))
index=index-1;
end
man=range(2,index+1);
return;%找到并退出
end
end
end
end

```

%求距离

```

function res=distance1(x,y,L)
[m n]=size(L);
for i=1:m
    res(i,1)=sqrt((x-L(i,1))^2+(y-L(i,2))^2);
end
end

```

%计算新定价

%遍历商品

```

Gao=[];
Di=[];
Dis=[];
for i=1:835
    for j=1:6
        %区分高低信誉
        if customer(groups2(i,j),4)>=19.9231
            Gao(end+1)=groups2(i,j);
        else
            Di(end+1)=groups2(i,j);
        end
    end
    %区位系数

```



```

        if task(i,2)<113.2148
            K=1.02;
        elseif task(i,2)>=113.2148&&task(i,2)<113.5454
            K=1.13;
        elseif task(i,2)>=113.54548&&task(i,2)<=113.9064
            K=0.97;
        else
            K=1.19;
        end
        Dis(end+1)=distance1(task(i,1),task(i,2),customer(groups2(i,j),:));
    end
    dd(i,3)=0.5*K*task(i,3)*sum(Dis)*1.8;
    if dd(i,3)>87
        dd(i,3)=75+rand();
    elseif dd(i,3)<60
        dd(i,3)=66+rand();
    end
    Dis=[];
end
ss=sum(dd(:,3))/835
ss1=sum(dd(:,3));
ss2=sum(A(:,3));

```

%求出任务到任务中心点的距离

```

function length=tasklens(data,center,index)
index(index==0)=[];
m=size(index);
length=zeros(1,m(2));
for i=1:m(2)
    length(i)=sqrt((data(index(i),1)-center(1))^2+(data(index(i),2)-center(2))^2);
end

```

%绘出打包点

```

function [result,sum,count,index,center]=sumtopoint(newtask,n)
[index,center]=fcmanddraw(newtask,n);
sum=zeros(1,n);
count=zeros(1,n);
result=[];
for i=1:n;

```

```

temp=tasklens(newtask,center(i,:),index(i,:));
[~,b]=size(temp);
count(i)=b;% 个数
if(b>1)
for j=1:b
sum(i)=sum(i)+temp(j);
end
if(sum(i)>0.022)% 还原点
plot(center(i,2),center(i,1),'*','color','w');
dd1=index(i,:);
dd1(dd1==0)=[];
dd2=newtask(dd1,1);
dd3=newtask(dd1,2);
result=[result,[dd2';dd3']];
else% 取中心
result=[result,[center(i,1);center(i,2)]];
plot(center(i,2),center(i,1),'*','color','k');
text(center(i,2)+0.0005,center(i,1)+0.0005,['(',num2str(b),')'],'color','m');
plot(newtask(index(i,:)~=0,2),newtask(index(i,:)~=0,1),'w');
end
else
plot(center(i,2),center(i,1),'*','color','w');
result=[result,[center(i,1);center(i,2)]];
end
end
end
end

```

%kmeans 聚类

```

clc;clear;
A=xlsread('1.xls',1);
B=xlsread('2.xlsx',1);
[m,n]=size(A);
[m0,n0]=size(B);
X=A(:,1:4);
Y=B(:,1:2);
opts = statset('Display','final');
[idx,ctr] = kmeans(X(:,1:2),90,...
'Distance','city',...
'Replicates',5);
plot(X(idx==1,1),X(idx==1,2),'r','MarkerSize',12)
hold on
plot(X(idx==2,1),X(idx==2,2),'m','MarkerSize',12)

```

```

plot(X(idx==3,1),X(idx==3,2),'c.','MarkerSize',12)
plot(X(idx==4,1),X(idx==4,2),'g.','MarkerSize',12)
%plot(X(idx==5,1),X(idx==5,2),'y.','MarkerSize',12)
plot(ctrs(:,2),ctrs(:,1),'ko',...
'MarkerSize',4,'LineWidth',1.5)
legend('Cluster 1','Cluster 2','Cluster 3','Cluster 4','Cluster 5','Centroids',...
'Location','NW')
% 求出每一个点到中心点的距离
L1=[X(idx==1,1),X(idx==1,2),X(idx==1,3),X(idx==1,4)];
L2=[X(idx==2,1),X(idx==2,2),X(idx==2,3),X(idx==2,4)];
L3=[X(idx==3,1),X(idx==3,2),X(idx==3,3),X(idx==3,4)];
L4=[X(idx==4,1),X(idx==4,2),X(idx==4,3),X(idx==4,4)];
[m1,n1]=size(L1);
[m2,n2]=size(L2);
[m3,n3]=size(L3);
[m4,n4]=size(L4);
dis1=distance(ctrs(1,1),ctrs(1,2),L1);
dis2=distance(ctrs(2,1),ctrs(2,2),L2);
dis3=distance(ctrs(3,1),ctrs(3,2),L3);
dis4=distance(ctrs(4,1),ctrs(4,2),L4);
% 求最大距离
dis1max=max(dis1(:,1));
dis2max=max(dis2(:,1));
dis3max=max(dis3(:,1));
dis4max=max(dis4(:,1));
% 分类的同时计算平均价格和完成率
count=zeros(4,12);
count(1,1:9)=fenlei(dis1max,dis1,m1);
count(2,1:9)=fenlei(dis2max,dis2,m2);
count(3,1:9)=fenlei(dis3max,dis3,m3);
count(4,1:9)=fenlei(dis4max,dis4,m4);
% 第一次分类，分出城市
lei1=fenlei1(dis1,B,m0);
lei2=fenlei1(dis2,B,m0);
lei3=fenlei1(dis3,B,m0);
lei4=fenlei1(dis4,B,m0);
% 第二次分类，分出三种区域
count(1,10:12)=fenlei2(dis1max,lei1,ctrs(1,1),ctrs(1,2));
count(2,10:12)=fenlei2(dis2max,lei2,ctrs(2,1),ctrs(2,2));
count(3,10:12)=fenlei2(dis3max,lei3,ctrs(3,1),ctrs(3,2));
count(4,10:12)=fenlei2(dis4max,lei4,ctrs(4,1),ctrs(4,2));

```

```

%开始匹配%
clear;clc;
A=xlsread('1.xls',1);
B=xlsread('2.xlsx',1);
customer=B;
task=A;
groups2=findcustomer(task,customer);
[a,~]=size(groups2);
mans=zeros();
for i=1:a
mans(i)=doublesystem(groups2(i,:),customer);
end
%运行代码
%findtest;
%找出信誉太小的点
mans1=mans;
for i=1:835
if customer(mans1(i),4)<=19.9231
    mans1(1,i)=nan;
end
end
xx=isnan(mans1);
aa=mans(find(xx==1));
hold on;
plot(customer(:,2),customer(:,1),'r.')
plot(customer(aa,2),customer(aa,1),'b.')
mans22=mans1';
[n,m]=size(mans22);
dd=1:n;
dd=dd';
dd=[dd,mans'];%分配结果
for i=1:835
    dd(i,3)=distance1(A(dd(i,1),1),A(dd(i,1),2),B(dd(i,2),:));
    dd(i,4)=B(dd(i,2),4);
    dd(i,5)=A(dd(i,1),3);
end

```

```

%BP, 要注意矩阵方向
A=load('训练.txt');
B=load('待训练.txt');
p=A(:,3:4)';%读取距离，信誉，价格后转置

```

```

t=A(:,5)';
p1=B(:,3:4)';
% Y=B(:,3:4)';% 读取位置， 价格后转置
[pn1,input_str1]=mapminmax(p1);
[pn,input_str]=mapminmax(p);
[tn,output_str]=mapminmax(t);
net=newff(pn,tn,[2 3 1],{'purelin','logsig','purelin'});
net.trainParam.show=10;
net.trainParam.lr=0.05;
net.trainParam.epochs=5000;
net.trainParam.goal=0.65*10^(-3);
net.divideFcn="";
net=train(net,pn,tn);
an=sim(net,pn1);
a=mapminmax('reverse',an,output_str);

```

%绘图及分类

```

function [index,center]=fcmanddraw(newtask,n)
[a,~]=size(newtask);
%index1=1:a;
%index1=index1';
%newtask1=[index1,newtask];% 会员矩阵加编号
opts = statset('Display','final');
[idx,ctr]= kmeans(newtask(:,1:2),n,...
'Distance','city',...
'Replicates',5);
scatter(newtask(:,2),newtask(:,1),'b. ');
hold on
%plot(X(idx==5,1),X(idx==5,2),'y.','MarkerSize',12)
%plot(ctr(:,2),ctr(:,1),'ko',...
'MarkerSize',4,'LineWidth',1.5)
center=ctr;
index=zeros(n,8);
for j=1:n
for i=1:a
if idx(i,1)==j
for s=1:8
if index(j,s)==0
index(j,s)=i;
break;
end
end
end

```

```

        end
    end
end
end
end

```

%聚类的平均定价

```

newprice=zeros(200,3);
for i=1:200
    tempindex=index(i,:);
    tempindex(tempindex==0)=[];
    [~,m]=size(tempindex);
    sum1=0;
    ave=0;
    for j=1:m
        sum1=sum1+a(1,tempindex(m));
    end
    ave=sum1/m;
    newprice(i,1)=ave*m*(0.64371+(2.63171+0.76245*m)^(-0.8951));
    newprice(i,2)=newprice(i,1)/m;
    newprice(i,3)=ave;
end
%连接数组
bao=[center,newprice];

```

%支持向量机, 要注意矩阵方向

```

A=xlsread('1.xls',1);
B=xlsread('C:\Users\F Vadim\Desktop\第四题数据.xlsx',1);
X=A(:,1:3)';%读取位置, 价格后转置
Y=[B(:,1:2),B(:,5)]';%读取位置, 价格后转置
[b,ps]=mapstd(X);
dd=mapstd('apply',Y,ps);
group=A(:,4);%列排布训练集
s=svmtrain(b',group);
sv_index=s.SupportVectorIndices;
beta=s.Alpha;
bb=s.Bias;
mean_and_std_train=s.ScaleData;

```

```
check=svmclassify(s,b');  
err_rate=1-sum(group==check)/length(group);  
solution=svmclassify(s,dd');
```