“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，建立了二元线性回归模型，解决了任务定价规律的研究和任务未完成原因的分析问题；建立了基于遗传算法的多目标规划模型和支持向量机模型，解决了新定价方案的设计问题；建立了规模效应函数和费用补贴函数模型，解决了任务打包后的定价问题；运用问题二和问题三所建模型，解决了新项目的定价问题。

针对问题一，建立了二元线性回归模型，解决了任务定价规律的研究和任务未完成原因的分析问题。用皮尔逊相关系数法分别分析任务周围 3km 内会员数、任务数、平均信誉值与标价的相关程度，为提高模型精度，舍去相关性弱的平均信誉值变量，建立二元线性回归模型，用最小二乘法拟合得到回归方程为 $y=72.3531-0.1276x_1-0.2063x_2$，且该回归模型分别通过了 DW 检验、t 检验和 F 检验，验证了模型适用性；最终通过数据可视化分析任务未完成原因。得出平台定价规律为：任务标价与 3km 内会员数、任务数呈负相关；任务未完成原因有两个：一是任务标价低，二是周围高信誉会员多。

针对问题二，建立了基于遗传算法的多目标规划模型和支持向量机模型，解决了新定价方案的设计问题。在问题一回归模型的基础上引入信誉值和距离指标建立新定价模型，以最小平均任务成本和最大任务选择率为目标函数建立多目标规划模型；用遗传算法求解出定价方程为 $y=99.6850-0.3689x_1+0.8593x_2-0.0128x_3+0.8561x_4$，据此制定新定价方案；用已知数据训练支持向量机，预测新定价方案下任务的完成情况，以任务完成率为评价指标，量化定价方案的实施效果。得出结论：在新方案下深圳市、广州市、东莞市、佛山市的任务完成率分别为 54.04%、68.59%、93.92%和 93.35%，相比于原方案有所提升，且总完成率从 62.51%提升至 76.59%。

针对问题三，建立了规模效应函数和费用补贴函数模型，解决了任务打包后的定价问题。首先，确定距离阈值 $r=500\,\mathrm{m}$，将阈值范围内的任务集打包发布；其次，将任务包分成定价过高与定价过低的两类，并分别建立规模效应函数和费用补贴函数，由此制定打包后的定价模型并求解定价方案；运用问题二中支持向量机模型预测完成情况，得出结论：修改定价方案后，深圳市、广州市、东莞市和佛山市的任务完成率分别为 60.87%、68.89%、96.69%和 90.46%，总任务完成率为 77.90%，相比于问题二中新定价方案完成率略有提升，说明将任务打包能够进一步提升任务完成率。

针对问题四，运用问题二中基于遗传算法的多目标优化模型和支持向量机模型，解决了新项目定价方案的求解问题。观测数据分布图，发现新项目任务密度较大，因此引入空间修正系数建立新项目定价模型，利用问题二中的多目标规划模型求解新项目定价方案，以及支持向量机模型预测任务完成情况，计算出新项目任务定价方案的完成率达到 91.49%。

关键词：多元线性回归　皮尔逊相关系数　遗传算法　多目标规划　支持向量机

一、问题重述

“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。

附件一是一个已结束项目的任务数据，包含了每个任务的位置、定价和完成情况（“1”表示完成，“0”表示未完成）；附件二是会员信息数据，包含了会员的位置、信誉值、参考其信誉给出的任务开始预订时间和预订限额，原则上会员信誉越高，越优先开始挑选任务，其配额也就越大（任务分配时实际上是根据预订限额所占比例进行配发）；附件三是一个新的检查项目任务数据，只有任务的位置信息。请完成下面的问题：

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

2.1 问题一的分析

由于平台在为现有任务定价时是从任务的角度出发的，而导致任务未完成的原因需从用户的角度来考虑，由此本题可分成两部分来解答。

针对第一部分任务定价规律的研究，主要目的是研究标价与相关数据之间的关系，是一个典型的统计分析问题。多元统计分析中的一个重要方法，即多元线性回归，它的主要思想是在确定了具有强相关性的自变量和因变量后，利用最小二乘法的原理拟合出多元线性回归方程，这一方程即可用来表示标价与自变量之间的关系，由此可分析出任务定价规律。进行回归分析之前，需对数据进行相关分析。从已知数据中可提取出任务位置、会员位置、信誉值、开始预定时间和预定限额这些可能影响定价的因素。考虑先用皮尔逊相关系数法分析这些影响因素与标价间的线性相关程度。为提高模型精度，只对相关性强的主要因素作后续分析。

针对第二部分任务未完成原因的分析，影响一个任务是否完成的因素有很多，诸如任务定价、任务难易度、任务所在地段会员密度、会员信誉度、任务发布时间、所在地经济发展水平、就业状况、气候状况（如台风、暴雨会影响某地区任务完成率）等。由于目前掌握数据有限，可通过数据可视化和数据统计，从以下几个方面分析任务未完成的原因：任务定价、会员密度、会员信誉度均值。

2.2 问题二的分析

问题二要求设计新的定价方案，并与原方案进行比较。问题二的建立是在问题一的基础上，对原定价方案的改善。可结合任务未完成的原因采取对原定价方案进行优化的

方法，以此提高任务的完成率。考虑在问题一回归模型上引入会员信誉值和任务与会员之间的距离这两个影响因素建立新的任务定价模型，并将其作为约束条件，除此之外会员的积极性、任务对会员的吸引程度和会员选择任务的优先权都可纳入考虑范围内，最终以任务平均成本和任务选择率为目标函数建立多目标优化模型。遗传算法是一种自适应的随机化搜索方法，已被广泛应用于处理多目标优化问题[1]，所以本题采取具有良好的稳定性、内在的隐并性和全局寻优能力的遗传算法求解新定价方案。为了将其与原方案进行比较，还需建立支持向量机模型预测任务完成情况，由此与附件一中任务完成情况进行对比分析。

2.3 问题三的分析

问题三的本质是在任务打包的前提下对问题二的定价方案作进一步改善。实际情况下，多个任务因为相对集中，会导致用户争相选择对自己有利的任务。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。

首先可以设置算法将位置集中的任务打包，由于每一任务包内的任务完成情况参差不齐，考虑先将任务包分为完成情况好和不好的两类；其次分别对两类任务包建立定价模型，求解出定价方案后仍可建立支持向量机模型预测任务完成情况。

2.4 问题四的分析

首先总结各个问题之间的关系，问题一可分析出平台定价模型的不足而为后文作铺垫，问题二是对这些不足的改善，问题三是针对打包问题作进一步优化，问题四是问题二和问题三所用模型的综合应用。结合问题二和问题三所建模型的优点，首先建立多目标优化模型求解每一任务的定价， 其次运用问题三模型的思想将位置集中的任务打包，由此求解任务打包后的定价方案，最后仍然建立支持向量机预测完成情况。

三、模型假设

1. 假设所提供数据都真实可信；
2. 假设每项任务仅由一位会员完成，但是一个会员可以完成多项任务；
3. 假设会员预定任务时相互独立；
4. 假设会员以追求利益最大化为目标。

四、符号说明

y	任务标价
n	任务总数
m	会员总数
R	半径阈值，单位：km
R'	距离阈值，单位：m
x_1	R 内会员数
x_2	R 内任务数
x_3	R 内会员平均信誉值
x_4	R 内任务平均距离，单位：km

r	皮尔逊相关系数
β	多元线性回归方程的回归系数
ε	多元线性回归方程的残差
d	任务之间的实际距离，单位：km
f	任务对会员的吸引程度
L_j	第 j 个会员的预定限额， $j=1,2,\cdots,m$
c_0	会员去到任务所在地的路程中的单位距离花费成本，单位：元
S	会员积极性指标
γ	空间修正系数

五、模型的建立与求解

5.1 问题一的模型建立与求解

为研究项目的任务定价规律，分别分析任务标价与其周围一定范围内的任务数、会员数和会员平均信誉值的相关程度，并建立多元线性回归模型；依据实际观测值求解回归方程，由此分析该结束项目的定价规律；通过相关数据的可视化，结合任务定价规律，分析任务未完成的原因。

问题一的第一部分：研究现有任务定价规律

5.1.1 数据预处理

观察附件二中会员信息数据，发现某位会员的位置数据异常，该会员编号为 B1175，位置为（113.131483 °N，23.031824 °E），除此之外其他所有会员位置的经度范围为 106.239083 °E ~116.97047 °E，纬度范围为 20.335061 °N~33.65205 °N，可以看出该异常会员的位置远远偏离于其他会员。由于所有数据都是真是可信的，推测该异常数据只是在录取过程中经纬度信息填反导致的差错，对其作如下修正即可：（113.131483 °N，23.031824 °E）→（23.031824 °N，113.131483 °E）。

5.1.2 模型建立

多元回归分析是研究自变量与因变量之间关系的统计方法，其主要步骤如下：

（1）自变量的选取

由附件一和附件二中所提供的数据可知影响平台定价的因素包括会员的位置、信誉值、开始预订时间和预订限额，以及任务的位置。由于会员的开始预定时间和预定限额是参考其信誉值给出的，所以可直接用信誉值代表这两个因素。

定义半径 $R=3\text{km}$，以每一任务为对象，结合主要影响因素可初步定义如下三个自变量：

①R 内会员数

以半径 R 作圆，计算圆内包含的所有会员数量，用 x_1 表示。

②R 内任务数

以半径 R 作圆，计算圆内包含的所有任务数量，用 x_2 表示。

③R 内会员平均信誉值

以半径 R 作圆，计算圆内包含的会员的平均信誉值，可表示为

$$x_3=\frac{1}{M}\sum_{k=1}^{M}I_k \tag{1}$$

式中：x_3为会员平均信誉值，M为圆内会员总数，I_k为圆内第k个会员的信誉值。

以上选取的三个自变量是从所提供数据中提取的，而平台现有的定价规律是否与上述变量有较强的因果关系还需作进一步的相关分析，与因变量相关性强的自变量保留，与因变量相关性弱的自变量舍去。

（2）相关分析

皮尔逊相关系数可准确地反映两变量之间的线性相关程度，首先，用皮尔逊相关系数检验各自变量与标价的相关程度，由此对自变量进行初步筛选。

a、理论准备：皮尔逊相关系数法的主要步骤如下[2]：

步骤一：提出假设：H_0：两组定量变量之间无显著线性关系，存在零相关；

H_1：两组定量变量之间有显著线性关系。

步骤二：选取显著性水平为 0.01。

步骤三：计算两组定量变量间的相关系数。

用相关系数$r\in[-1,1]$来表示相关程度的大小：

$$r=\frac{1}{n}\sum_{i=1}^{n}\left(\frac{x_i-\bar{x}}{S_x}\right)\left(\frac{y_i-\bar{y}}{S_y}\right) \tag{2}$$

式中：n为总任务数，有$n=835$，x_i为自变量值，y_i为因变量值。$r>0$表示正线性相关，$r>0$表示正线性相关，$r=0$表示不线性相关。

步骤四：确定检验统计量。皮尔逊相关系数法的检验统计量为：

$$t=\frac{r\sqrt{n-2}}{\sqrt{1-r^2}} \tag{3}$$

式中：t统计量服从$n-2$个自由度的t分布。

步骤五：计算检验统计量的观测值和对应的概率P值。

步骤六：决策。若所得概率P值小于显著性水平α，则拒绝原假设，认为两组变量间存在显著的线性关系；反之，若所得概率P值大于显著性水平α，则接受原假设，认为两组变量间存在零相关。

b、皮尔逊相关系数检验

利用皮尔逊相关系数检验法分别对R内会员数、任务数、会员平均信誉值与任务标价进行相关分析，得到结果如下：

①R内会员数与任务标价的相关分析

表 5.1-1 第一组数据的皮尔逊相关系数检验结果

		任务标价	R 内会员数
任务标价	Pearson 相关性 显著性（双尾） N	1 835	-0.486** 0.000 835
R 以内会员数	Pearson 相关性 显著性（双尾） N	-0.486** 0.000 835	1 835

注：**.在置信度（双侧）为 0.01 时，相关性是显著的。

由表 5.1-1 可知，R 内会员数与任务标价的相关系数为 0.486，且两者之间存在负的强相关性。相关系数检验的概率 P 值近似为 0，所以当显著性水平 α 为 0.01 时，应拒绝原假设，认为 R 内会员数与任务标价具有显著的线性关系。

②R 内任务数与任务标价的相关分析

表 5.1-2 第二组数据的皮尔逊相关系数检验结果

		任务标价	R 内任务数
任务标价	Pearson 相关性 显著性（双尾） N	1 835	-0.432** 0.000 835
R 以内任务数	Pearson 相关性 显著性（双尾） N	-0.432** 0.000 835	1 835

由表 5.1-2 可知，R 内任务数与任务标价之间呈现负相关性，其相关系数为 0.432，相关系数检验的概率 P 值近似为 0。因此，当显著性水平 α 为 0.01 时，拒绝原假设，认为 R 内任务数与任务标价有显著的线性关系。

③R 内会员平均信誉值与任务标价的相关分析

表 5.1-3 第三组数据的皮尔逊相关系数检验结果

		任务标价	R 内会员平均信誉值
任务标价	Pearson 相关性 显著性（双尾） N	1 835	0.43 0.215 835
R 以内任务数	Pearson 相关性 显著性（双尾） N	0.43 0.215 835	1 835

由表 5.1-3 可以看出，R 内会员平均信誉值与任务标价的相关系数检验的概率 P 值为 0.215，大于显著性水平 0.01，因此接收原假设，认为 R 内会员平均信誉值与任务标价无显著线性关系。

c、自变量筛选

由于 R 内会员数、R 内任务数两个自变量通过了皮尔逊相关系数检验（P <0.01），R 内会员平均信誉值未通过皮尔逊相关系数检验，为了提高模型精度，剔除 R 内会员平均信誉值进行后续分析。

（3）建立多元线性回归模型

综上所述，随着 R 内会员数的增加，任务标价降低，两者有较强的相关性，R 内任务数与标价的关系也类似，因此建立二元线性回归模型[3]：

$$y=\beta_0+\beta_1x_1+\beta_2x_2+\varepsilon \qquad (4)$$

式中：β_0,β_1,β_2 为回归系数；ε 为随机误差项，假设 ε 相对独立，且服从均值为 0 的正态分布。

记 $i=1,2,\cdots,n$ 代表每一任务的编号，n 为任务总数，自变量与因变量的 n 组实际观测值可用矩阵来表示：

$$Y=\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}, X=\begin{pmatrix} 1 & x_{11} & x_{21} \\ 1 & x_{12} & x_{22} \\ \vdots & \vdots & \vdots \\ 1 & x_{1n} & x_{2n} \end{pmatrix} \quad (5)$$

式中：y_i 代表第 i 个任务的标价；x_{1i}, x_{2i} 分别代表第 i 个任务 R 内会员数和 R 内任务数。

以上二元线性回归模型可写为矩阵形式：

$$Y = X\beta + \varepsilon \quad (6)$$

5.1.3 模型求解

多元线性回归分析的主要思想就是利用最小二乘法来拟合模型，最小二乘法的原理是令残差平方和

$$ESS = (Y - X\hat{\beta})'(Y - X\hat{\beta}) \quad (7)$$

最小，得到

$$\hat{\beta} = (X'X)^{-1}(X'Y) \quad (8)$$

$\hat{\beta}$ 即为 β 的最佳线性无偏估计量。根据附件一和附件二的数据，利用 Eviews 软件进行最小二乘法拟合，拟合 R 方为 0.2750，拟合结果图见附录一中图 1-1。

实际上，在做回归分析时，模型的随机误差项 ε 可能存在相关性，违背之前关于 ε 相互独立的假设。D-W 检验是对 ε 进行自相关性判断的一种统计方法[4]，若 DW 检验值接近 2，表明 ε 的自相关性弱或不存在自相关；若 DW 检验值接近 0 或 4，表明 ε 的自相关性强。从附录一的图 1-2 中可知该模型的 DW 检验值为 1.5051，说明 ε 存在自相关。因此，对模型进行一阶差分，以消除随机误差项的自相关性，最终求得各回归系数的估计值如表 5.1-4 所示。

表 5.1-4 各回归系数的最佳线性无偏估计量

回归系数	β_0	β_1	β_2
估计值	72.3531	-0.1276	-0.2063

一阶差分后的拟合结果图见附录一中图 1-3，从图中看出一阶差分后模型的拟合 R 方有所提高，为 0.3170；DW 检验值为 2.1001，说明 ε 不存在自相关，一阶差分后得到的模型是适用的。

最终得到描述平台现有定价规律的二元非线性回归模型如下：

$$y = 72.3531 - 0.1276x_1 - 0.2063x_2 \quad (9)$$

分析该二元非线性回归模型，任务周围会员数每增加一个，任务定价减少 0.1276 元；任务周围任务数每增加一个，任务定价减少 0.2063 元，分析其原因是平台为了吸引更多的用户量，所以对于偏远地区的定价相对更高，而会员集中地区的定价则相对较低，并且为了使各地区的任务均价保持稳定，所以任务越多的地方定价越低，而任务少的地方对每一任务则可以标更高的价格。从总体上看，R 内会员数、R 内任务数与任务标价呈现负相关性，与相关分析的结论一致，验证了模型的准确性。

问题一的第二部分：分析任务未完成原因

由分析可知，本文通过数据可视化和数据统计分析，依次从以下几个方面分析任务

未完成的原因：会员密度、任务定价、会员信誉度均值。

（1）未完成任务与会员密度的关系

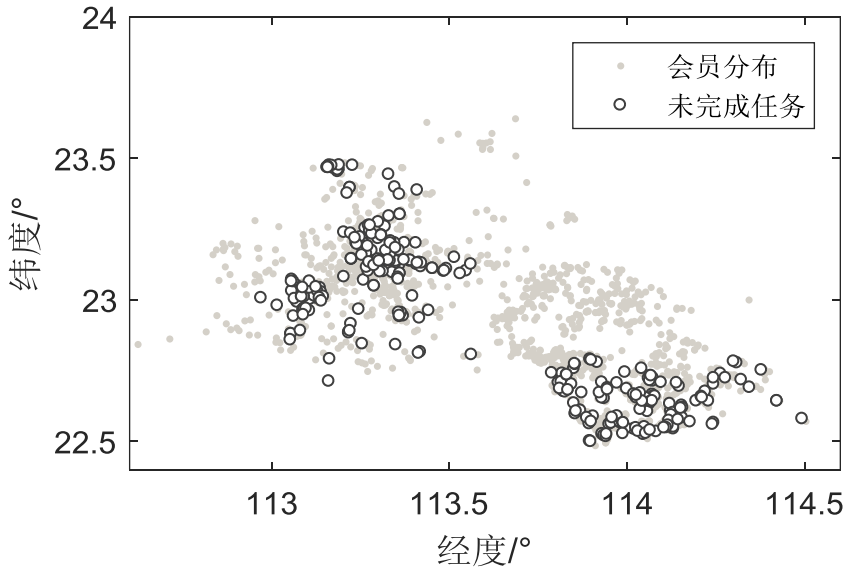

图 5.1-2 未完成任务与会员分布的关系图

从图中看出未完成任务主要集中在会员数量多的地方，而会员稀疏的地区未完成任务少或没有。由平台的定价规律知任务周围会员数越多，其标价越低。可推断出现图中现象的原因归根结底是由于该处任务价格低的原因，会员们对低价任务的积极性相对较小，导致该处大多任务无人问津。下面对未完成任务与价格的关系作进一步分析。

（2）未完成任务与任务标价的关系

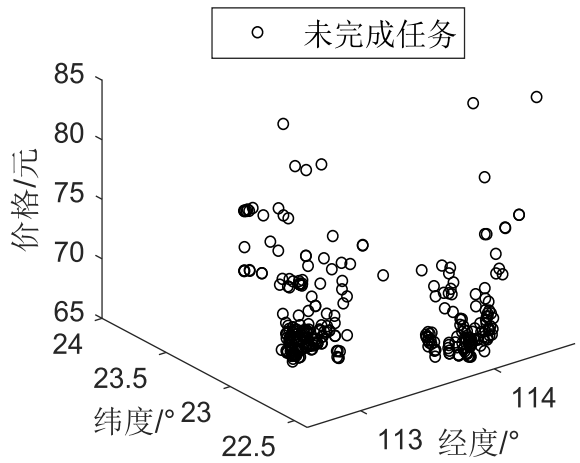

图 5.1-1 未完成任务与价格的关系图

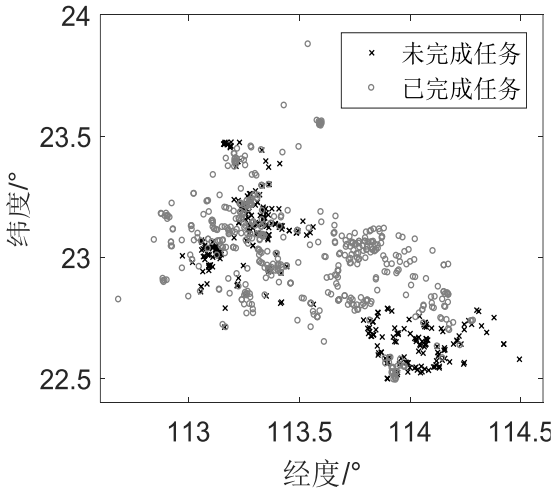

图 5.1-2 任务分布图

未完成任务与任务标价的数据可视化如图 5.1-1 所示，从图中看出未完成任务随着价格的降低而增多，可初步推断未完成成任务与任务标价存在负相关性。

接下来，利用数据定量分析未完成任务与标价的关系。从图 5.1-2 可以看出任务分布的大致情况。通过对附件一、附件二中所给数据的经纬度的检索，发现数据点主要分布在广东省的深圳市、广州市、东莞市、佛山市，而未完成任务的点主要集中在深圳市和广州市。

经过计算得出，四个城市的任务完成率（即完成任务数与总任务数的比值）分别为21.12%、60.94%、98.34%、65.90%，完成率从大到小依次为：东莞市、佛山市、广州市、深圳市；四个城市的任务平均定价为 67.1926、68.0672、70.2790、71.4624（元），平均定价从大到小依次为：佛山市、东莞市、广州市、深圳市。任务完成率排序与均价排序大致对应，因而可以初步得出结论：一个地区的任务完成率与任务均价存在一定的正相关性。

为了进一步验证任务完成率与定价的正相关性，将附件一中所有任务按定价统计，去除其中样本数少于 10 的点（即定价为 69.5、71、71.5、73.5、74、74.5（元）的点），得出如下表格。

表 5.1-5 任务标价与任务完成率

定价/元	65	65.5	66	66.5	67	67.5
任务完成率	53.85%	50.67%	37.36%	48.15%	47.37%	73.91%
定价/元	68	68.5	69	70	70.5	72
任务完成率	83.33%	54.55%	63.16%	80.21%	81.82%	70.00%
定价/元	72.5	73	75	80	85	
任务完成率	80.00%	70.00%	75.64%	69.23%	88.89%	

横向观察该表，可以看出任务完成率与任务定价有较大的正相关性，因此可以得出结论：在其他情况相近的条件下，标价越低，任务有更大的概率不被完成。从而推断任务未被完成的原因是这些任务的标价太低，而平台用户对价格敏感度高，用户争先完成价格高的任务，对于价格越低的任务积极性较小，所以大量低价任务被用户们忽略。

（3）未完成任务与会员信誉值的关系

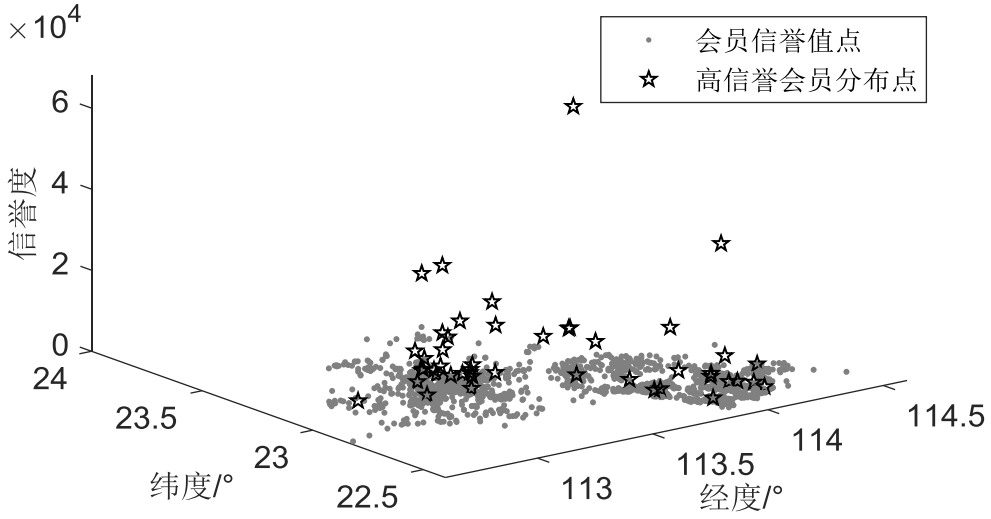

图 5.1-3 未完成任务与会员信誉值的关系图

观察附件二中会员信誉值的数据，最高信誉值为 67997.39，而最低信誉值只有0.0,01，发现其存在极差大的特点，若直接用该数据作图不能很好地体现不同信誉值的分布。所以将会员信誉值点分为两部分，信誉值大于 1000 的会员统称为高信誉会员，用五角星符号表示，信誉值小于等于 1000 的会员用点表示。从图中看出会员数量越多

的地方会员信誉值越高。结合前文的分析可知，未完成任务主要集中在会员信誉值高的区域且多为低价任务，分析其原因是高信誉会员具有优先预订任务的权利，所以他们往往趋向于标价高的任务，由此而忽略了标价低的任务。

5.1.4 问题回答

综上所述，平台现有的任务定价模型为 $y=74.05876-0.275331x_1-0.478321x_2$，总结项目的任务定价规律和任务未完成原因如下：

（1）项目任务定价规律：任务标价与 3km 内会员数、3km 内任务数呈负的线性关系；

（2）任务未完成原因：①未完成任务主要是定价低的任务，而平台用户对价格敏感度高，用户争先完成价格高的任务，对于价格越低的任务积极性较小，所以大量低价任务不被用户选择；②未完成任务主要集中在会员信誉值高的区域，而高信誉会员具有优先预订任务的权利，所以他们往往趋向于标价高的任务，由此而忽略了标价低的任务。

5.1.5 模型检验与分析

（1）回归参数的显著性检验

多元回归分析中对各个回归系数的显著性检验，目的在于分别检验当其他自变量不变时，该回归系数对应的自变量是否对因变量有显著影响。用 t 检验对各个自变量作显著性检验，检验结果如下表。

表 5.1-5 回归参数的显著性检验结果

	t 检验值	概率 P 值
R 内会员数	-11.48493	0.0000
R 内任务数	-9.83201	0.0000

表 5.1-5 给出了模型对两个自变量的偏回归系数是否等于零的 t 检验结果。t 值分别等于-11.48493 和-9.83201，概率 p 值都小于显著性水平 0.05，说明当在其他变量不变的情况下，R 内会员数、R 内任务数分别对任务标价有显著的影响。

（2）回归方程的显著性检验

验证了每个自变量对因变量都有显著影响后，还需对因变量与所有自变量之间的线性关系在整体上是否显著作出检验，这种检验是在方差分析的基础上利用 F 检验进行的，检验结果为：F 检验值为 116.0980，显著性概率近似于 0，说明回归方程整体显著，即认为 R 内会员数、R 内任务数联合起来对任务标价有显著影响。

综上所述，该二元线性回归模型通过了自相关检验、回归参数和回归方程的显著性检验，用它来描述项目的任务定价规律是合理且准确的。

5.2 问题二的模型建立与求解

在问题一回归模型的基础上引入信誉值和距离指标建立新定价模型，并将其作为约束条件；以平均成本最低和任务选择率最高为目标函数建立多目标优化模型；利用遗传算法求解得到最佳定价方案；建立支持向量机模型预测新定价方案的完成情况，计算完成率并将其与原方案的完成率作比较。

5.2.1 模型建立

5.2.1.1 目标分析

（1）平均成本最低

设某城市共有 m 个会员，n 个任务，令 $i=1,2,\cdots,m$ 代表会员的编号，$j=1,2,\cdots,n$ 代表任务的编号。根据第一问结果的分析，任务未被完成主要是由现有定价规律考虑的因素不足造成的，且未完成任务与会员信誉值有很大关系，所以建立新定价模型如下：

$$y_i=\alpha_0+\alpha_1x_{1i}+\alpha_2x_{2i}+\alpha_3x_{3i}+\alpha_4x_{4i} \tag{10}$$

式中：y_i 为任务定价；$\alpha_0,\alpha_1,\alpha_2,\alpha_3,\alpha_4$ 为待优化系数；x_{1i}，x_{2i},x_{3i} 分别为 R 内会员数、R 内任务数和 R 内会员平均信誉值，在问题一模型的建立中已给出定义；x_{4i} 为 R 内任务平均距离（千米），定义如下：

$$x_{4i}=\frac{1}{N}\sum_{k=1}^{N}d_{ik} \tag{11}$$

式中：N 为圆内任务总数；d_{ik} 为第 i 个任务至其领域（以 R 为半径的圆）内第 k 个任务的实际距离（千米）。

要使得平均成本最低，则有

$$\min\frac{1}{n}\sum_{i=1}^{n}y_i \tag{12}$$

（2）任务选择率最高

有如下定义：

$$w_{ij}=\begin{cases}0 & ,第j个会员预订了第i个任务\\ 1 & ,第j个会员预订了第i个任务\end{cases} \tag{13}$$

为使任务选择率最高，有

$$\max\frac{1}{n}\sum_{j=1}^{m}\sum_{i=1}^{n}w_{ij} \tag{14}$$

5.2.1.2 约束条件

（1）一个任务要么没有会员完成，要么仅由一位会员完成，所以有如下约束：

$$\sum_{j=1}^{m}x_{ij}=\{0,1\} \tag{15}$$

当该值取 0 时代表没有会员完成此任务，当该值取 1 时代表有一个会员完成此任务。

（2）一位会员完成的任务数不能超过预定任务限额，即

$$\sum_{i=1}^{n}x_{ij}\le L_j \tag{16}$$

（3）会员选择任务时主要考虑任务标价和任务与会员之间的距离这两个因素，定义任务对会员的吸引程度为

$$f=f(y_i,x_{i4})=y_i-2\cdot x_{i4}\cdot c_0 \tag{17}$$

式中：c_0 为会员去到任务所在地的路程中的单位距离花费成本（元）。

（4）定义会员积极性指标 S，取决于当地经济发展水平。

（5）定义会员挑选任务的优先权：

a、若预定任务开始时间越早，则优先权越大。

b、预定任务开始时间一样，若预定任务限额越大，则优先权越大。

（6）分配规则的确定：

a、会员优先权越大，则会员将优先挑选任务；

b、任务按照得分由高到低排序，会员优先挑选得分最高的任务；

c、若现有任务得分均小于会员积极性指标，会员将退出队列，下一个优先权次之的会员将开始挑选任务；

d、若会员现有任务达到预定任务限额，会员将退出队列，下一个优先权次之的会员将开始挑选任务。

5.2.1.3 建立多目标优化模型

基于以上分析，以（11）、（12）式为目标，以（15）、（16）式为约束条件，建立模型如下：

$$\min \sum_{i=1}^{n} y_i$$

$$\max \frac{1}{n} \sum_{j=1}^{m} \sum_{i=1}^{n} w_{ij}$$

$$\text{s.t.} \begin{cases} \sum_{j=1}^{m} x_{ij} = 1 \\ \sum_{i=1}^{n} x_{ij} = L_j \\ y_i = \alpha_0 + \alpha_1 x_{1i} + \alpha_2 x_{2i} + \alpha_3 x_{3i} + \alpha_4 x_{4i} \\ x_{4i} = \frac{1}{N} \sum_{k=1}^{N} d_{ik} \end{cases} \tag{18}$$

其中，待优化系数 $\alpha_0, \alpha_1, \alpha_2, \alpha_3, \alpha_4$ 为决策变量。

5.2.2 模型求解

（1）遗传算法求解多目标优化模型

遗传算法是模拟达尔文遗传选择和自然淘汰的生物进化过程的计算模型，其主要步骤如下[5-6]：

步骤一：本文采用实数编码方法编码，每个染色体为一个实数向量。

步骤二：随机生成初始化种群。

步骤三：确定适应函数。可用多目标优化模型的目标函数代替种群的适应度计算函数：

$$F = f(x) \tag{19}$$

步骤四：依据轮盘赌法筛选优秀个体，个体适应度越高，被选择进行遗传的概率越大；个体适应度越低，被选择的概率就小。这样计算的结果和目的是：最优秀的染色体被复制成多分遗传给下一代，中等的染色体维持原有水平，较差的染色体则不被选择最终淘汰。个体 i 被选中的概率为

$$p_i = \frac{F_i}{\sum_{j=1}^{N} F_j} \tag{20}$$

式中： F_i 为个体 i 的适应度值； N 为种群个体的数目。

步骤五：遗传算子的设计：

①交叉操作：由于本文中采取实数编码。所以交叉操作采用实数交叉法，第 k 个染色体 a_k 和第 l 个染色体 a_l 在 j 位的交叉操作方法为

$$\begin{aligned} a_{kj} &= a_{ij}(1-b)+a_{lj}b \\ a_{lj} &= a_{lj}(1-b)+a_{kj}b \end{aligned} \qquad (21)$$

其中，b 是区间 $[0,1]$ 的随机数。

②变异操作：其主要目的是维持种群多样性。第 i 个体的第 j 个基因进行变异的操作方法为

$$a_{ij} = \begin{cases} a_{ij}+(a_{ij}-a_{max})*f(g), r \geq 0.5 \\ a_{ij}+(a_{\min}-a_{ij})*f(g), r < 0.5 \end{cases} \qquad (22)$$

式中：$a_{\max}$ 是基因 a_{ij} 的上界；$a_{\min}$ 是基因 a_{ij} 的下界；$f(g)=r_2(1-g/G_{\max})^2, r_2$ 为随机数，g 是当前迭代次数，$G_{\max}$ 是最大进化次数，r 为区间 $[0,1]$ 的随机数。

步骤六：设定遗传算法相关参数：本文种群大小因为群体规模越大，越容易找到最优解。交叉概率为 0.6 能够保证种群的充分进化。一般而言，变异发生的可能性较小，变异概率为 0.01 更加符合自然规律。考虑到时间复杂度，设定最大遗传代数为 10。

利用上述遗传算法的步骤求解多目标优化模型，下图反映了目标函数随着迭代代数的平均寻优情况。

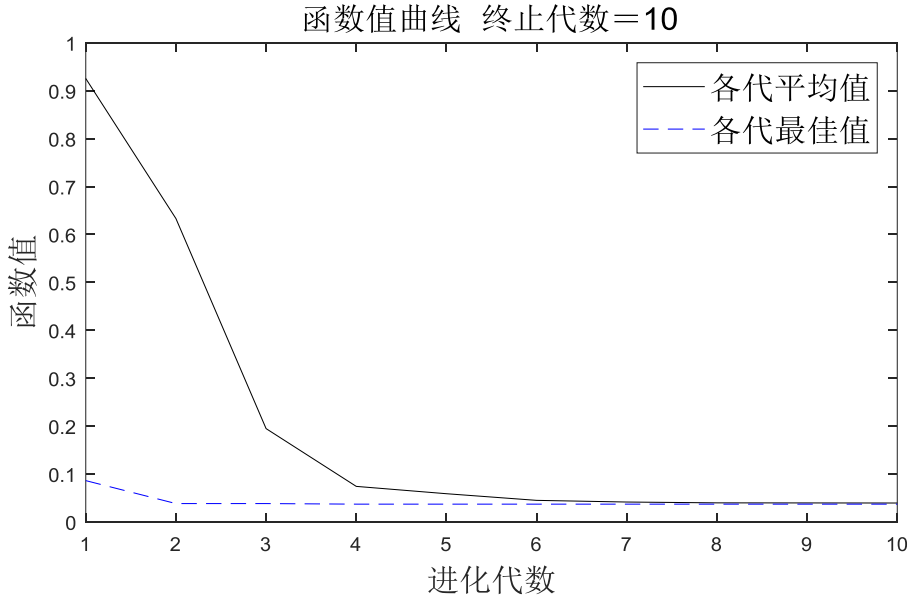

图 5.2-1 遗传算法函数值曲线

从图中可以看出，遗传算法可以非常稳健地向着全局最优的方向收敛，当进化到第 7 代时，寻找到了新定价模型各系数的最优值如表 5.1-4 所示，寻优速度较快。

表 5.1-4 各系数最优解

系数	α_0	α_1	α_2	α_3	α_4
最优解	99.6850	-0.3689	0.8593	-0.0128	0.8561

将所有系数值代入式（10）中，最终得到新定价模型为：

$$y = 99.6850 - 0.3689x_1 + 0.8593x_2 - 0.0128x_3 + 0.8561x_4 \qquad (23)$$

依据新定价模型计算新任务定价方案（具体表格见支撑材料）。

（2）支持向量机预测新定价方案的完成情况

为了将新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 R 内会员数、R 内任务数、R 内会员平均信誉值、R 内任务平均距离和定价，对所有指标数据进行如下标准化处理：

$$\tilde{x}_{ij}=\frac{x_{ij}-\mu_j}{s_j},\quad i=1,2,\cdots,n,\ j=1,2,3,4,5 \tag{24}$$

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。

记标准化后已分类任务行向量为 $b_i=[\tilde{a}_{i1},\cdots,\tilde{a}_{i5}],i=1,2,\cdots,n$。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[7-8]

$$c(\tilde{\boldsymbol{x}})=\sum_i \beta_i K(\boldsymbol{b}_i,\tilde{\boldsymbol{x}})+b \tag{25}$$

当 $c(\tilde{\boldsymbol{x}})\geq 0$，$\tilde{\boldsymbol{x}}$ 属于第 1 类；当 $c(\tilde{\boldsymbol{x}})\leq 0$，$\tilde{\boldsymbol{x}}$ 属于第 2 类。

用附件一、附件二所提供的实际观测数据集对支持向量机进行训练，随后以新定价方案求解的结果作为预测集，用训练过的模型得到预测的完成情况。根据预测的完成情况，计算新定价方案在不同城市的完成率（即完成任务数与总任务数之比），将其与原定价方案的完成率进行比较，如表 5.2-1 所示。

表 5.2-1 原定价方案和新定价方案在不同城市的完成率

	深圳市	广州市	东莞市	佛山市	总完成率
原定价方案	21.12%	60.94%	98.34%	65.90%	62.51%
新定价方案	54.04%	68.59%	93.92%	93.35%	76.59%

新的任务定价方案综合考虑了 R 内会员数、R 内任务数、R 内会员平均信誉值、R 内任务平均距离这四个方面的因素对定价的影响，比原方案相对来说考虑得更加全面。分析结果：采用新的定价模型后，深圳市任务完成率显著提高，广州市保持在中上水平，东莞市保持在较高水平，佛山市有了大幅提升；得出结论：使用新的任务定价方案后，总完成率从 62.51%提高到 76.59%，且四个地区的任务完成率都达到了较高的水平，效果较为显著的为深圳市与佛山市，任务完成率分别提升了 15.87%和 41.65%，说明新的定价方案更合理。

5.2.3 问题回答

引入会员信誉值和任务与会员之间的距离这两个影响因素建立的新定价模型为 $y=99.6850-0.3689x_1+0.8593x_2-0.0128x_3+0.8561x_4$，新的任务定价方案表格见支撑材料；在新方案下深圳市、广州市、东莞市、佛山市的任务完成率分别为 54.04%、68.59%、93.92%和 93.35%，相比于原方案都有所提升，且总完成率从 62.51%提升至 76.59%。

5.3 问题三的模型建立与求解

为建立任务打包后的定价模型，首先确定距离阈值，设计算法将阈值范围内任务集打包，其次将任务包分成定价过高和过低的两类，对于第一类任务包建立规模效应函数模型，对于第二类任务包建立费用补贴函数模型，再次将问题二求解的任务定价数据代入模型中求解任务打包后的定价方案，最终利用支持向量机模型预测完成情况。

5.3.1 模型建立

（1）打包位置集中的任务

设计算法将位置集中的任务打包，算法的主要步骤如下：

步骤一：定义距离阈值 R'，用 $i=1,2,\cdots,835$ 分别表示编号为 A0001~A0835 的 835 个已结束任务，计算任意两个任务之间的距离，其表达式为：

$$d_{ij}=\sqrt{(x_i-x_j)^2+(y_i-y_j)^2} \tag{26}$$

式中：d_{ij} 表示第 i 个任务与第 j 个任务的实际距离（km）；(x_i,y_i) 表示第 i 个任务的经纬度坐标，$i=1,2,\cdots,835$；(x_j,y_j) 表示第 j 个任务的经纬度坐标，$j=1,2,\cdots,835$，$j\neq i$。

步骤二：对每一个已结束任务，遍历它与其他所有任务的实际距离，存在判断条件：

$$d\leq R'$$

记录下所有满足此条件的任务编号，说明这两个任务的距离小于 r，需要将它们打包。

步骤三：对记录结果进行人工筛选。例如：对于编号 A0001 的任务，其记录结果中包括编号为 A0002 的任务，说明两任务的实际距离小于 500m，那么观察 A0002 的记录结果中一定会存在 A0001，由此将两任务打包。按照举例的方法对所有记录结果进行处理。

依次执行算法的每一步骤，最终得到的任务打包结果放在支撑材料中，总共有 338 个任务进行了打包处理，共打包成 134 份。

（2）任务包分类

根据支持向量机对任务包内任务完成率的预测，将任务包分为两类：

A 类：任务包内任务全部完成

B 类：任务包内任务没有全部完成

对于 A 类任务包，其中任务定价过高，所以包内任务全部完成，对于此类任务包可适当降低价格发布；对于 B 类任务包，其中任务定价过低，所以包内任务未全部完成，对于此类任务包可适当调高价格发布。

（3）建立定价模型

a、对于 A 类任务包建立规模效应函数模型：定义任务包总定价 $F(x)$ 满足以下两个规则：

①平均定价 $\dfrac{F(x)}{x}$ 关于 x 单调下降，且满足

$$\lim_{x\to\infty}F(x)=C\sum_{i=1}^{x}y_i \tag{27}$$

式中：x 表示任务包内任务个数，C 表示成本系数，y_i 表示任务包内第 i 个任务单独发布时的定价。

②效益函数 $R(x)=F(x)-C\sum_{i=1}^{x}y_i$ 关于 x 单调上升。

可以得到满足上述规则的模型为

$$F(x)=[C+(A+Bx)^{-\alpha}]\sum_{i=1}^{x}y_i \tag{28}$$

式中：A、B、C、α 为待定参数。

b、对于 B 类任务包建立规模效益函数模型：

由于任务之间存在定价、位置等因素的差异，部分会员有选择地选择“好”任务而不愿接受“坏任务”。会员“挑单”的行为造成部分经济效益较差的任务难以完成。如果能把所谓的“好”任务与“坏”任务打包起来，并给予一定的提价，使所有任务对会员来说经济效益基本相同，则会员将不会再“挑单”，此时就能有效提升任务的完成率。

定义任务不平衡指标Q满足

$$Q=\frac{\sqrt{\sum_{i=1}^{x}(y_i-\mu)\Big/x}}{\sum_{i=1}^{x}y_i} \tag{29}$$

式中：x表示任务包内任务个数，y_i表示任务包内第i个任务单独发布时的定价，μ表示任务包内的平均定价。

则任务包总定价$F(x)$满足

$$F(x)=\sum_{i=1}^{x}y_i\cdot(1+Q) \tag{30}$$

5.3.2 模型求解

选取距离阈值$R'=500m$，查阅文献[8]，对于（28）式中的参数分别取$A=2.6317$，$B=0.7624$，$C=0.6437$，$\alpha=0.8952$，结合（28）、（30）式求得修改后的定价模型为

$$F(x)=\begin{cases}[0.6437+(2.6317+0.7624x)^{-0.8952}]\sum_{i=1}^{x}y_i & ,\text{对于}A\text{类包}\\ \sum_{i=1}^{x}y_i\cdot(1+Q) & ,\text{对于}B\text{类包}\end{cases} \tag{31}$$

将问题二求解的定价代入（31）式中，得到打包后定价方案（见支撑材料）。

仍然运用问题二中支持向量机模型预测完成情况，并计算不同城市的完成率，结合前两种方案的完成率，制得汇总表如下。

表 5.3-1 三种定价方案在不同城市的完成率

	深圳市	广州市	东莞市	佛山市	总完成率
原定价方案	21.12%	60.94%	98.34%	65.90%	62.51%
新定价方案	54.04%	68.59%	93.92%	93.35%	76.59%
打包后定价方案	60.87%	68.69%	96.69%	90.46%	77.90%

纵观表中数据，比较打包后定价方案的完成率与新方案的完成率，深圳市和广州市的任务完成率有所提高，佛山市和东莞市的完成率保持在较高水平，总完成率略有提高。得出结论：任务打包后的定价方案优于新定价方案，且两方案都优于原定价方案，总完成率提升到 77.90%，且四个地区的任务完成率略有提升，说明打包后的定价方案最合理。

5.3.3 问题回答

选取距离阈值$r=500\,\text{m}$，筛选出 338 个需打包的任务，总共打包成 134 份。任务打包后对定价方案进行修改，最终任务完成情况为深圳市、广州市、东莞市和佛山市的任务完成率分别为 60.87%、68.89%、96.69%和 90.46%，总任务完成率为 77.90%，相比于问题二中新定价方案完成率略有提升，说明将任务打包能够进一步改善任务完成率。

5.4 问题四模型的建立与求解

为求解新项目的定价方案，结合问题二和问题三中定价模型的优点，引入空间修正系数，在问题二模型的基础上建立新项目定价模型；用问题三中同样方法将任务打包，并求解任务打包后的定价方案，仍然采用支持向量机模型预测完成情况。

5.4.1 数据的预处理

附件三中的 2066 个任务点有 533 个分布在深圳、1505 个分布在广州，以广州市为例，通过数据可视化技术，在地图中绘制出任务的空间分布图如下。

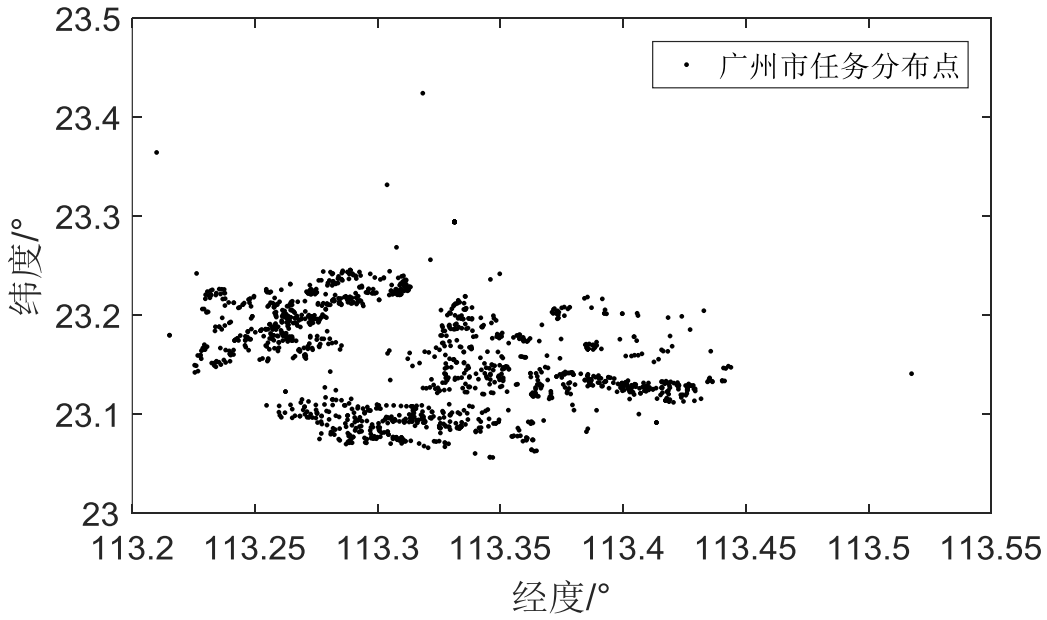

图 5.4-1 新项目任务分布图

分析图 5.4-1 可知，相对于附件一中数据而言，任务点分布密度明显增大。

5.4.2 模型的建立

由于任务点密度增大，因此在制定定价方案时，引入空间修正系数γ，在问题二模型的基础上建立新项目定价模型，满足

$$y_i = \alpha_0\gamma + \alpha_1\gamma x_{1i} + \alpha_2\gamma x_{2i} + \alpha_3\gamma x_{3i} + \alpha_4\gamma x_{4i} \tag{32}$$

5.4.3 模型求解

（1）求解定价方案

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 A 类任务包的处理方式重新分配打包任务的价格。最终得到新定价模型为：

$$y = 45.8551 - 0.1697x_1 + 0.3953x_2 - 0.0059x_3 + 0.3938x_4 \tag{33}$$

依据新定价模型计算新任务定价方案（具体表格见支撑材料）。

（2）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），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，两地总体任务完成率达到了 91.49%。

六、模型评价与改进

6.1 模型的优点

1. 对于问题一，建立多元线性回归模型之前，利用相关系数检验筛选了变量，避免了无关变量对模型精度的影响。并且对模型进行显著性检验，再一次验证了变量选取的合理性。

2. 对于问题二，利用数据可视化技术挖掘出原有定价模型没有考虑到的因素，建立多目标规划模型，求解得到了新的定价方案，并且进行了检验，验证了新的定价方案的合理性。综合运用数据挖掘与处理技术，最大化地挖掘了所给数据的信息。

3. 对于问题三，分别针对定价过高与定价过低的任务包，制定了不同的定价方案。针对定价过高任务包，建立规模效应函数，在保证任务完成率的情况下，适当减少了费用支出；针对定价过低任务包，建立费用补贴函数，提高了会员完成任务的积极性。并且进行了模型检验，发现这种方案有效提高了任务完成率，验证了模型的有效性。

6.2 模型的缺点

1. 由于时间限制，未能进一步充分挖掘数据的特征与相关性指标，模型的精度有待进一步的提高。这也可以作为本文下一步的研究方向。

6.3 模型的改进

1. 可优化支持向量机，选择更加合适的核函数，以提高模型分类预测的精度。

2. 本模型经适当简化与改进后可以应用于共享单车的调度、网约车调度与安排等问题的解决。

参考文献

[1]韩煜东,董双飞,谭柏川. 基于改进遗传算法的混装线多目标优化[J/OL]. 计算机集成制造系统,2015,(06):1476-1485.

[2]薛薇. 统计分析与 SPSS 的应用. 北京:中国人民大学出版社,2014.

[3]周晨,冯宇东,肖匡心,韩珊,董颖. 基于多元线性回归模型的东北地区需水量分析[J]. 数学的实践与认识,2014,(01):118-123.

[4]赵卫亚. DW 检验的局限性与模型的高阶自相关检验[J]. 统计与决策,2004,(01):18-19.

[5]金敏,鲁华祥. 一种遗传算法与粒子群优化的多子群分层混合算法[J]. 控制理论与应用,2013,(10):1231-1238.

[6]Sandro Chiappone,Orazio Giuffrè,Anna Granà,Raffaele Mauro,Antonino Sferlazza. Traffic simulation models calibration using speed–density relationship: An automated procedure based on genetic algorithm[J]. Expert Systems With Applications,2016,44:.

[7]司守奎,孙玺菁. 数学建模算法与应用. 北京:国防工业出版社,2016.

[8]Sarah M. Erfani,Sutharshan Rajasegarar,Shanika Karunasekera,Christopher Leckie. High-Dimensional and Large-Scale Anomaly Detection using a Linear One-Class SVM with Deep Learning[J]. Pattern Recognition,2016,:.

[9]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),2011,(03):17-21.

附录

附录一　问题一代码及结果图

1.1 问题一代码

```
%%Q1_1.m
clc
clear all
load data
%%
R=5;
for i=1:length(F1)
      Thing(i).IDnumber=find_people(F1(i,1),F1(i,2),R);
end
R2=R;
for i=1:length(F1)
      Thing(i).IDofMission=find_mission(F1(i,1),F1(i,2),R2 );
end
for i=1:length(Thing)

NIDnumber(i)=length(Thing(i).IDnumber);
NIDofMission(i)=length(Thing(i).IDofMission);
end
NIDnumber=NIDnumber';
NIDofMission=NIDofMission';
n=[NIDnumber NIDofMission F1(:,4)];

result=[[1:length(F1)]' n F1(:,3)];
%{
%%
for i=1:length(F22)
      Vip(i).IDnumber=find_people(F22(i,1),F22(i,2),R );
end

for i=1:length(F22)
      Vip(i).IDofMission=find_mission(F22(i,1),F22(i,2),R2 );
end
for i=1:length(Vip)

NIDnumber2(i)=length(Vip(i).IDnumber);
NIDofMission2(i)=length(Vip(i).IDofMission);
end
NIDnumber2=NIDnumber2';
NIDofMission2=NIDofMission2';
n2=[NIDnumber2 NIDofMission2 ];

%}

for i=1:length(Thing)
       ave_cred(i)=sum(Thing(i).IDnumber);
end
```

```
ave_cred=ave_cred';

%%find_people.m
function [ IDnumber ] = find_people( wei,jing,R )

J=jing/180*pi;
W=wei/180*pi;
load fun_data
IDnumber=[];
k=1;

for i=1:length(F22)

    C = sin(F22(i,1)*pi/180)*sin(W) + cos(F22(i,1)*pi/180)*cos(W)*cos(J-F22(i,2)*pi/180);
    Distance =6371.004*acos(C);%单位为 km
    if Distance<=R
        IDnumber(k)=i;
        k=k+1;
    end
end
end

%%find_mission.m
function [ IDofMission] = find_mission( wei,jing,R )
J=jing/180*pi;
W=wei/180*pi;
load fun_data
IDofMission=[];
k=1;
for i=1:length(F1)

    C = sin(F1(i,1)*pi/180)*sin(W) + cos(F1(i,1)*pi/180)*cos(W)*cos(J-F1(i,2)*pi/180);
    Distance =6371.004*acos(C);%单位为 km
    if Distance<=R
        IDofMission(k)=i;
        k=k+1;
    end
end
end
```

1.2 最小二乘法拟合结果图

Dependent Variable: Y
Method: Least Squares
Date: 09/17/17 Time: 18:05
Sample: 1 835
Included observations: 835

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	72.45477	0.242141	299.2257	0.0000
X1	-0.132133	0.013118	-10.07235	0.0000
X2	-0.212854	0.031877	-6.677427	0.0000

R-squared	0.275037	Mean dependent var	69.11078
Adjusted R-squared	0.273295	S.D. dependent var	4.512772
S.E. of regression	3.847003	Akaike info criterion	5.536052
Sum squared resid	12313.13	Schwarz criterion	5.553037
Log likelihood	-2308.302	Hannan-Quinn criter.	5.542564
F-statistic	157.8227	Durbin-Watson stat	1.505133
Prob(F-statistic)	0.000000		

图 1-1 最小二乘法拟合结果图

1.3 一阶差分后拟合结果图

Date: 09/17/17 Time: 18:05
Sample (adjusted): 2 835
Included observations: 834 after adjustments
Convergence achieved after 5 iterations

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	72.35308	0.268641	269.3302	0.0000
X1	-0.127552	0.013557	-9.408541	0.0000
X2	-0.206279	0.032706	-6.307145	0.0000
AR(1)	0.244293	0.034006	7.183773	0.0000

R-squared	0.316987	Mean dependent var	69.11451
Adjusted R-squared	0.314518	S.D. dependent var	4.514192
S.E. of regression	3.737473	Akaike info criterion	5.479481
Sum squared resid	11594.02	Schwarz criterion	5.502149
Log likelihood	-2280.944	Hannan-Quinn criter.	5.488172
F-statistic	128.4011	Durbin-Watson stat	2.100111
Prob(F-statistic)	0.000000		

图 1-2 一阶差分后模型的拟合结果图

附录二：问题二遗传算法代码主要部分

```
%%main.m 主函数
%% 清空环境
clc
clear
warning off

%% 遗传算法参数
maxgen=10;                          %进化代数
sizepop=100;                        %种群规模
pcross=[0.6];                       %交叉概率
pmutation=[0.01];                   %变异概率
```

```
lenchrom=[1 1 1 1 1];
bound=[-1 0;0 1;-1 0;0 1;0 100];    %变量范围
%%  个体初始化
individuals=struct('fitness',zeros(1,sizepop), 'chrom',[]);    %种群结构体
avgfitness=[];                                                  %种群平均适应度
bestfitness=[];                                                 %种群最佳适应度
bestchrom=[];                                                   %适应度最好染色
体
%  初始化种群
for i=1:sizepop
    individuals.chrom(i,:)=Code(lenchrom,bound);            %随机产生个体
    x=individuals.chrom(i,:);
    individuals.fitness(i)=Fun(x);                          %个体适应度
end

%找最好的染色体
[bestfitness bestindex]=min(individuals.fitness);
bestchrom=individuals.chrom(bestindex,:);    %最好的染色体
avgfitness=sum(individuals.fitness)/sizepop; %染色体的平均适应度
%  记录每一代进化中最好的适应度和平均适应度
trace=[];

%%  进化开始
for i=1:maxgen

    %  选择操作
    individuals=Select(individuals,sizepop);
    avgfitness=sum(individuals.fitness)/sizepop;
    %  交叉操作
    individuals.chrom=Cross(pcross,lenchrom,individuals.chrom,sizepop,bound);
    %  变异操作
    individuals.chrom=Mutation(pmutation,lenchrom,individuals.chrom,sizepop,[i
maxgen],bound);


    %  计算适应度
    for j=1:sizepop
        x=individuals.chrom(j,:);
        individuals.fitness(j)=Fun(x);
    end

    %找到最小和最大适应度的染色体及它们在种群中的位置
    [newbestfitness,newbestindex]=min(individuals.fitness);
    [worestfitness,worestindex]=max(individuals.fitness);
    %  代替上一次进化中最好的染色体
    if bestfitness>newbestfitness
        bestfitness=newbestfitness;
        bestchrom=individuals.chrom(newbestindex,:);
```

```
    end
    individuals.chrom(worestindex,:)=bestchrom;
    individuals.fitness(worestindex)=bestfitness;

    avgfitness=sum(individuals.fitness)/sizepop;

    trace=[trace;avgfitness bestfitness]; %记录每一代进化中最好的适应度和平均适应度
end
%进化结束

%% 结果显示
figure
[r c]=size(trace);
plot([1:r]',trace(:,1),'r-',[1:r]',trace(:,2),'b--');
title(['函数值曲线  ''终止代数＝' num2str(maxgen)],'fontsize',12);
xlabel('进化代数','fontsize',12);ylabel('函数值','fontsize',12);
legend('各代平均值','各代最佳值','fontsize',12);
ylim([0 30])
disp('函数值                    变量');
% 窗口显示
snapnow;
disp([bestfitness x]);
grid on
```

附录三：问题二目标规划代码主要部分

```
%% Fun.m
function FF=Fun(XX)
load data_Q2_2
m=length(Ren);%会员数
n=length(All);%任务数
%会员积极性指标，是一个常数，
S=60;
%单位距离花费成本 Cx，单位：元/千米，是一个常数
Cx=5;
Price=fun1(XX);
F1=sum(Price)/835;%所有的平均价
%x 是“任务-会员”匹配的 0-1 矩阵
%F2 是所有任务“被选择率”
[F2,x]=fun2(Price,Cx,S);
% FF=F1/90-F2;
FF=1-F2;                %max F2  被选择率最大
end

%% fun1.m
function Price=fun1(XX)

%总成本最小函数
load City
```

```
A=XX(1);
B=XX(2);
C=XX(3);
D=XX(4);
E=XX(5);
%A B C D E 为四个自变量
X1=City(:,1);
X2=City(:,2);
X3=City(:,3);
X4=City(:,4);
Price=A*X1+B*X2+C*X3+D*X4+E;
end


%% fun2.m
function [F2,x]=fun2(Price,Cx,S)
%会员与任务匹配函数
load data_Q2_2
m=length(Ren);
n=length(All);
x=zeros(m,n);
%任务吸引度指标  得到吸引度矩阵 Y_xiyin
for i=1:m
Y_xiyin(i,:)=Price'-2*Dis(i,:)*Cx;
end
for i=1:length(Thing)
    huiyuan=Thing(i).IDnumber;%圈内所有会员的编号
     Xinyu=Thing(i).xinyu;%与上述会员对应的信誉度
     renwu=Thing(i).IDofMission;%圈内所有任务的编号
     if sum(x(:,i))==1
          continue            %如果中心任务被选择就跳出循环，接着遍历下一个任务点
      end

     for j=1:length(huiyuan)
     Tiao1=huiyuan(j);%Tiao1：挑选任务的会员的编号,从 1 开始挑选



     xiyin=zeros(1,length(renwu));
     for k=1:length(renwu)
     xiyin(k)=Y_xiyin(Tiao1,renwu(k));
     end
     renwu_paixu=paixu(renwu,xiyin);%根据吸引度从大到小排序完成后，是一个 2*n 的
矩阵，第一行任务编号，第二行吸引度
     if  sum(x(Tiao1,:))>Ren(Tiao1,3)||max(xiyin)<S          %达到预定额度或者总吸引度
不足,则此会员离开，下一位选择，j++
          continue
     end
     for J=1:size(renwu_paixu,2)
```

```
    Tiao2=renwu_paixu(1,J);   %Tiao2 是当前选择的任务的编号
    if sum(x(:,Tiao2))==0
%如果在 x 矩阵中判断出，Tiao2 任务还没有被选择，那么当前会员和当前任务匹配
      x(Tiao1,Tiao2)=1;
      end

    end
  end
end
beixuan=sum(sum(x));
F2=beixuan/n;
end
```

附录四：问题二支持向量机代码

```
clc, clear
a0=xlsread('训练数据集.xls');
a1=xlsread('预测数据集.xlsx');
b0=a0(:,[1:5]); dd0=a1(:,[1:5]); %提取已分类和待分类的数据
% dd0=[dd0,80*ones(835,1)];
[b,ps]=mapstd(b0); %已分类数据的标准化
dd=mapstd('apply',dd0,ps); %待分类数据的标准化
group=a0(:,[6]); %已知样本点的类别标号
s=svmtrain(b,group) %训练支持向量机分离器
sv_index=s.SupportVectorIndices   %返回支持向量的标号
beta=s.Alpha   %返回分类函数的权系数
bb=s.Bias   %返回分类函数的常数项
mean_and_std_trans=s.ScaleData  %第 1 行返回的是已知样本点均值向量的相反数，第 2
行返回的是标准差向量的倒数
check=svmclassify(s,b)   %验证已知样本点
err_rate=1-sum(group==check)/length(group) %计算已知样本点的错判率
solution=svmclassify(s,dd) %对待判样本点进行分类
solution(isnan(solution))=0.5;%部分信息缺失导致的无法分类
```

附录五：问题三计算价格代码

```
clc
clear
load data
load data_dabao
k=1;
kk=1;
for i=1:length(data_dabao)
    if haohuai(i)==0
        Hao(k,:)=data_dabao(i,:);
        k=k+1;
    else
        Huai(kk,:)=data_dabao(i,:);
        kk=kk+1;
    end
```

```
end
for i=1:length(Hao)
    t=Hao(i,:);
    t(find(t==0))=[];
    x(i)=length(t);
    for j=1:length(t)
        PP(i,j)=Price(t(j));
    end
    t=[];
    PP_sum(i)=sum(PP(i,:));
end

PP_sum=PP_sum';
x=x';
A=2.56125;
B=0.89983;
C=0.67898;
a=0.93958;
F=PP_sum.*(C+power((A+B.*x),(-a)));

for i=1:length(Huai)
    t=Huai(i,:);
    t(find(t==0))=[];
biaozhuncha(i)=std(t);
t=[];
end
biaozhuncha=biaozhuncha';

for i=1:length(Huai)
    t=Huai(i,:);
    t(find(t==0))=[];
    x2(i)=length(t);
    for j=1:length(t)
        PP2(i,j)=Price(t(j));
    end
    t=[];
    PP2_sum(i)=sum(PP2(i,:));
end
PP2_sum=PP2_sum';
x2=x2';
R=biaozhuncha./PP2_sum;
F2=PP2_sum.*(R+1);
Price_new=zeros(size(Price));
Price_new=Price;

%好包
for i=1:length(Hao)
    t=Hao(i,:);
    t(find(t==0))=[];
    for j=1:length(t)
```

```
        Price_new(t(j))=F(i)/length(t);
        end
end

%坏包

for i=1:length(Huai)
        t=Huai(i,:);
        t(find(t==0))=[];
        for j=1:length(t)
        Price_new(t(j))=F2(i)/length(t);
        end
end
```

附录六：问题四计算价格代码

```
clc
clear
load dataX
load data_dabao
X1=dataX4(:,1);
X2=dataX4(:,2);
X3=dataX4(:,3);
X4=dataX4(:,4);
Price=99.6850214574949-0.368858792985045.*X1+0.859320411428785.*X2-0.0127846883
952565.*X3+0.856064276339207.*X4       ;
gama=0.46;

Price=Price.*gama;
for i=1:length(data_dabao)
        t=data_dabao(i,:);
        t(find(t==0))=[];
        x(i)=length(t);
        for j=1:length(t)
                PP(i,j)=Price(t(j));
        end
        t=[];
        PP_sum(i)=sum(PP(i,:));
end
PP_sum=PP_sum';
x=x';
A=2.56125;
B=0.89983;
C=0.67898;
a=0.93958;
F=PP_sum.*(C+power((A+B.*x),(-a)));

JIA=F./x;
Price_new=Price;
for i=1:length(data_dabao)
        t=data_dabao(i,:);
```

```
        t(find(t==0))=[];
        x(i)=length(t);
        for j=1:length(t)
            Price_new(t(j))=JIA(i);
        end
        t=[];
end
```