# “拍照赚钱”的任务定价


## 摘 要

本文针对“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规律模型，研究了任务定价规律；建立了 *Logistic* 回归分析模型，定量分析了任务未完成的原因；建立了基于单目标规划的定价模型，对已完成任务重新定价并与原方案进行比较；建立了基于遗传模拟退火算法的模糊 *C*-均值聚类模型和多目标规划模型，解决了在任务打包发布的情况下定价模型的修改问题；建立了基于支持向量机的动态价格预测模型，重新确定了定价方案，并定义了方案评价指标评价该方案的实施效果。

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与 *Logistic* 回归分析模型，分别研究了任务定价规律和任务未完成的原因。首先，利用 *K-means* 聚类算法将任务聚类；其次，建立基于偏最小二乘回归分析的定价规律模型，研究定价规律；再次，建立 *Logistic* 回归模型分析任务未完成的原因；最后对模型进行检验。最终得密集区定价较低，偏远地区定价较高的结论；任务未完成的原因为任务分布与标价不合理、周边会员数量与任务数量不匹配。

针对问题二，建立了基于单目标规划的定价模型，解决了对已完成任务的重新定价问题。首先，定义会员意愿度以及会员期望价格函数；其次，以任务完成率最高为目标建立单目标规划模型；最后计算改进方案的任务完成率并与原方案对比。最终得到对于未完成的任务以及偏远地区的任务价格稍有提高，对于完成度较高的任务密集区的任务价格稍有降低。改进方案的任务完成率为 0.677844，大于原方案的 0.625598。

针对问题三，建立了基于遗传模拟退火算法的模糊 *C*-均值聚类模型和多目标规划模型，解决了在任务打包发布的情况下定价模型的修改问题。首先，建立基于遗传模拟退火算法的模糊 *C*-均值聚类模型对任务进行打包；其次，考虑商家成本与会员的预定限额建立多目标规划模型，并用贯序算法进行求解；最后，调整打包数量计算不同情况下的任务完成率并分析对其的影响。最终得到各类最优打包数量分别为 280、220、260 个；打包数量过小容易导致任务完成率低，打包数量过多容易导致商家成本过大。

针对问题四，建立了基于支持向量机的动态价格预测模型，解决了定价方案的确定问题，并定义了方案评价指标评价该方案的实施效果。首先对任务进行聚类和打包；其次查找相关数据，得到 638 个历史数据集，将前 600 个数据作为训练集、后 38 个数据作为测试集训练支持向量机；再次建立基于支持向量机的动态价格预测模型对任务进行价格预测；最后定义了方案评价指标评价了该方案的实施效果。最终得到较优的定价方案，各类评价指标值分别为 0.1983 和 0.1532。



**关键词：**偏最小二乘回归模型 *Logistic* 回归模型 模拟退火算法 支持向量机
模糊 *C*-均值聚类模型

# 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

根据附件一中所给的信息，分析任务的定价规律可以从任务所处的地理位置和任务的标价两个方面进行研究。首先，对任务的定价规律进行定性分析，观察任务的定价与外部环境的联系。通过每项任务的标价和分配该项任务数量之间的大致情况，初步推断定价与任务分配数量之间存在的某种关系。其次，对任务的定价规律进行更深一步的探讨，即定量分析。这需要选取合适的指标，使得其能大致反映对价格变动的影响情况。由此可以从该项任务的成本费用、发布任务所处地理位置的繁华程度、会员到任务地点的距离、会员的信誉值等方面考虑，得到定价与各项指标之间的具体规律。最后，由于任务完成情况是 0-1 变量，因此通过 *Logistic* 回归分析任务未完成的原因，根据其回归系数的大小分析任务未完成的主要原因。

## 2.2 问题二的分析

设计新的定价方案即要求该方案与原方案相比的任务完成度有一定程度的提高。从会员的角度出发，该任务是否可以完成取决于会员完成该任务的意愿，若实际定价高于会员内心期望的任务定价，会员就有极大的意愿完成该任务，任务完成的可能性较大，反之任务完成的可能性较小；从任务本身的角度出发，任务定价的合理性对于该任务是否能够被完成有重要意义，任务点与会员之间的距离远近会在一定程度上影响任务的完成度。因此，新的定价方案从以上两个角度出发，建立单目标规划模型，以任务完成数

量最多为目标，求得新的任务定价方案。最后定义任务完成度指标，与原定价方案对比，分析新定价方案的优势。

### 2.3 问题三的分析

本题需要解决的问题是在原有定价方案的基础上合理修改定价模型，提高任务的完成度，解决众包分配中存在的一系列问题。由于任务位置集中而出现的将任务打包发布的做法，避免了由于任务密集用户会争相选择的问题。然而任务打包方式多样，密集区各任务距离较近，并没有明显的界限，会出现多个任务在一个包中导致会员无法完成的情况，因此传统的聚类方法已经不能满足条件的需要，考虑选用基于智能算法的聚类算法解决此问题。在这种情况下，考虑用户的预定限额、预定任务开始时间与信誉，修改问题二的定价模型，建立多目标规划模型，以任务完成数量最多为第一目标，总定价额度最小为第二目标，列出相应的约束条件，利用贯序算法对问题进行求解。

### 2.4 问题四的分析

本题要求制定新的任务定价方案。然而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。但是由于本题数据获得途径有限，可用数据样本数量较小，由此，采用支持向量机回归预测的方法对以往的定价及其会员参与度进行学习，并以此为基础预测价格和会员参与度。将预测数据加入历史数据集并再次预测，最终得到最优任务定价方案。最后，定义方案实施效果评价指标对该方案的实施效果进行评价。

## 三、基本假设

1、任务难度相同；
2、会员的拍照水平相同；
3、会员在选择任务时不受自身信誉、活跃度等的影响;
4、所有任务点位于中国境内，即纬度为北纬、经度为东经。

## 四、符号说明

| | |
|---|---|
| C | 基本成本费用 |
| $x_1$ | 周边会员个数 |
| $x_2$ | 周边相同任务数 |
| $x_3$ | 任务点与道路距离 |
| σ | 偏最小二乘偏差系数 |

| | |
|---|---|
| $\varepsilon$ | *Logistic* 回归分析偏差率 |
| $P$ | 任务完成的概率 |
| $\beta_i$ | 回归系数 |
| $m_{ij}$ | 会员意愿程度 |
| $u_{ij}$ | 第j个会员对于第i项任务的期望定价 |
| $p_i$ | 第 i 项任务的实际定价 |
| $n_i$ | 第 i 项任务的完成情况 |
| $A_i$ | 任务定价 |
| $\gamma_2$ | 任务完成度指标 |
| $K(x,y)$ | 核函数 |
| $\mu$ | 会员参与度 |
| $\kappa$ | 方案实施效果评价指标 |

## 五、模型的建立与求解

### 5.1 模型的准备

#### 5.1.1 无效数据的剔除

假设所有任务点位于中国境内，即纬度为北纬、经度为东经，对三个附件中的所给数据进行观察，剔除附件二中经纬度颠倒的无效数据，防止错误数据对结果产生影响。

#### 5.1.2 经纬度与实际距离的转化

由于题目中所给经度与经度、纬度与纬度之间相差较小，因此在计算时本文将地球表面近似看作一个球面而非椭球面来表达地球的空间几何形态[1]。由此，两个坐标$A(x,y)$和$B(a,b)$间的球面距离为：

$$D = R \times \{arccos[cosb \times cosy \times cos(a-x) + sinb \times siny]\}$$

其中，$R$为地球半径。由此计算得到任务与任务之间的距离、任务与会员之间的距离。

### 5.2 问题一的模型建立与求解

为研究任务定价规律，本文首先进行定性分析，通过附件一中所给信息绘出任务完成情况散点图、任务与会员分布图、任务定价空间分布图，初步分析任务的定价规律。其次，考虑影响定价的因素，定义周边会员个数、周边等同任务数、任务点到达便利程度三个指标，利用偏最小二乘回归分析得出定价与指标之间的函数关系与各个指标对价格的影响力度。最后，利用 *Logistic* 回归分析任务未完成的可能原因。

### 5.2.1 模型的建立

#### 5.2.1.1 寻找任务定价规律——定性与定量分析相结合

**1、定性分析**

找出项目中任务的定价规律，要保证分析进行的合理全面。首先，从定性的角度对任务的定价规律进行研究。根据附件一中所给的数据，利用 *MATLAB* 绘出任务完成情况分布图，如图 1 所示：

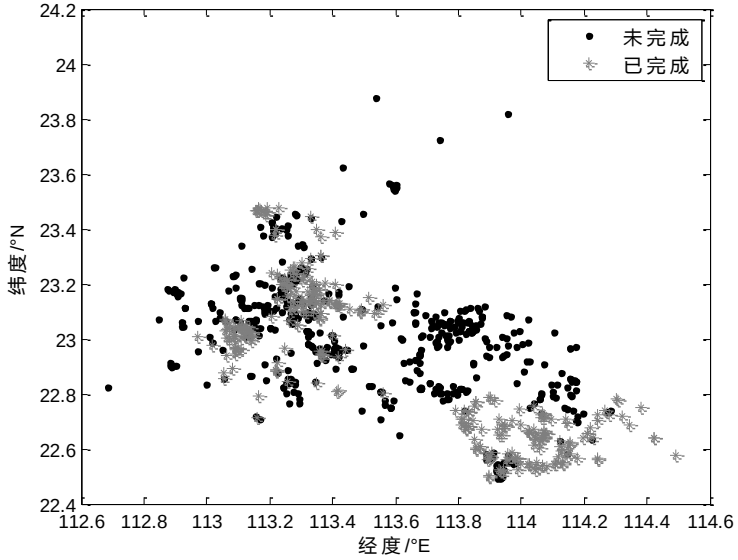


图 1 附件一任务完成情况分布图

根据图 1 可以得到，已完成的任务分布较为集中，多集中在$22.6°N$~$23.6°N$ ，$113°E$~$113.5°E$和$22.5°N$~$22.8°N$，$113.8°E$~$114.3°E$两个地区，经查阅地图得集中的区域多为人流量较多的公园、住宅繁华区等地段；而未完成任务的散点分布较零散，多分布在广州市与深圳市的郊区以及东莞市，大致分布在$23°N$~$23.2°N$，$113.6°E$~$113.8°E$。

其次，利用附件一中的信息得到任务位置与会员位置分布的散点图，如图 2 所示：

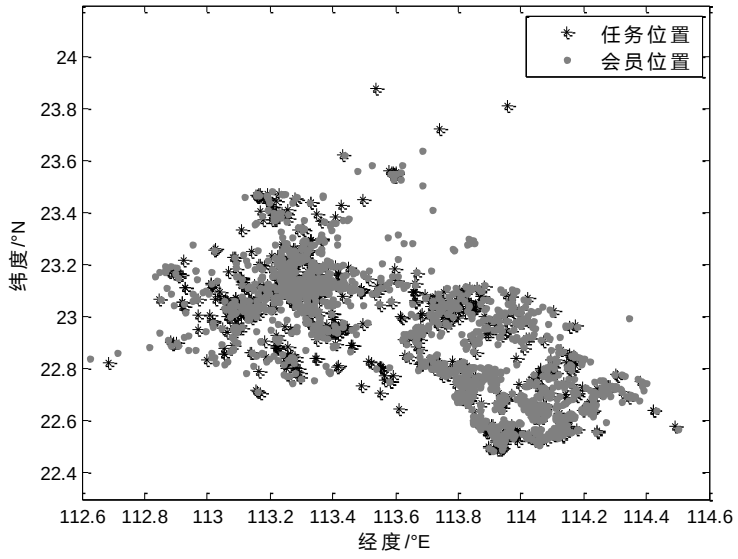


图 2 附件一任务位置与会员位置散点图

由图 2 可知，任务多分配于会员周围，多个任务的位置可能存在比较集中的现象，导致会员争相选择；而有些任务数量多于会员数量或者周边会员距离任务位置较远而导致部分任务无人完成的情况。且可以发现，任务的分布大多在会员密集程度较高的区域。

再次，对所给任务经纬度和其对应价格通过插值拟合的方法绘出其空间分布图，如图 3 所示：

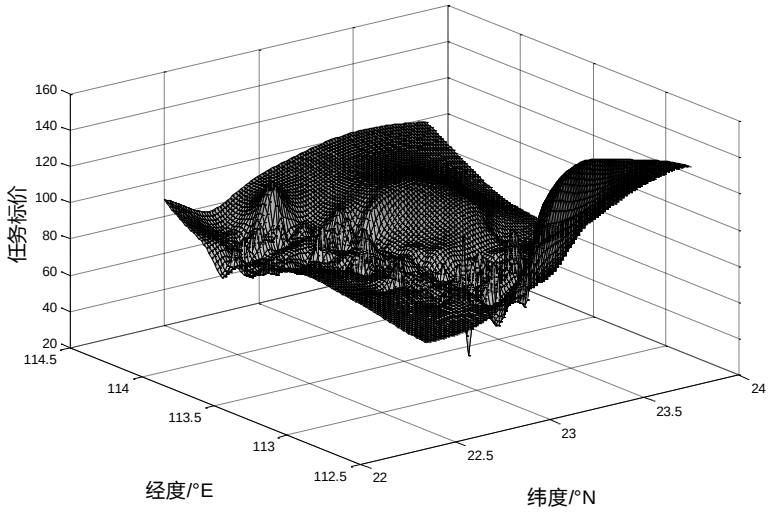


图 3 任务定价的空间分布

根据任务定价的空间分布图，可以看出其空间分布图的表面凹凸不平，即价格的高低由于实际地理位置的不同而存在差异。分布图的边缘地带即实际地图中的偏远地区任务的定价较高，其原因可能是由于发布任务所处的地理位置较偏远，会员人数较少，任务的完成难度较高，因此需要利用较高的定价吸引会员来完成发布的任务。

最后，绘制定价区间与任务个数之间的关系示意图，如图 4 所示：

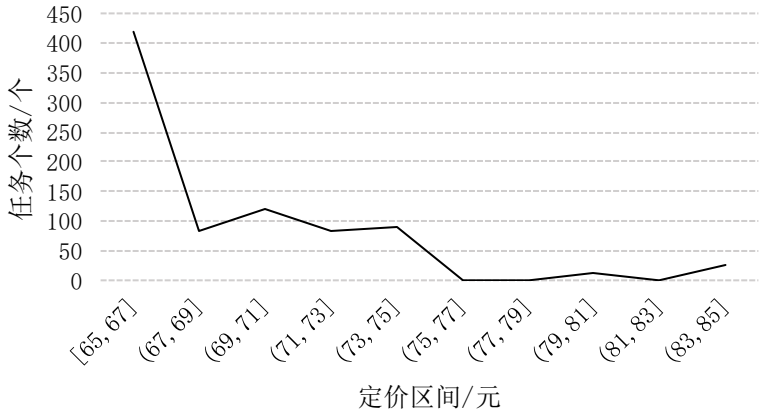


图 4 定价区间与任务个数的关系示意图

通过图 4 可以发现，当定价在区间[65,67]内时，开发商发布的任务个数最多，总数在 400 个以上，且与其他价格区间发布的任务个数悬殊极大；当处于区间(67,69]、(69,71]、(71,73]、(73,75]时，各任务数量相差较小，都维持在 100 个左右；其他剩余区间分配的任务数量极小，在 0~50个之间。由此可以大致推断出，区间[65,67]可能为会员最愿意接受的任务价格区间。

**2、定量分析——基于偏最小二乘回归分析的任务定价规律模型**

由定性分析可知，任务位置集中于 3 个地域，分别为广州、深圳、东莞附近，所以，本题分别对各城市内部进行聚类分析，寻找各城市的任务定价规律。

（1）*K-means* 聚类算法[2]

*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。初始规定聚类的类别数为 $k$，基于初始的 $k$ 个聚类中必对样本进行聚类，并得出相应的聚类结果。根据聚类效果的判别准则，对 *K-means* 聚类算法采用迭代更新的办法，每一次的迭代过程使聚类效果更加接近判别准则，最终使聚类效果达到满足判别准则的目标，取得较优的聚类效果。其基本过程如下：

设 $n$ 个数据样本为$X=\{x_1,x_2,\dots,x_n\}$,$c(2\le c\le n)$是要将数据样本分成的类型的数目，$\{A_1,A_2,\dots,A_c\}$表示相应的 $c$ 个类别，$U$ 是其相似分类矩阵，各类别的聚类中心为$\{v_1,v_2,\dots,v_c\}$，$\mu_k(x_i)$是样本$x_i$对于类$A_k$的隶属度（简写为$\mu_{ik}$）。则目标函数$J_b$可以用下式表达：

$$J_b(U,v)=\sum_{i=1}^{n}\sum_{k=1}^{c}(\mu_{ik})^b(d_{ik})^2$$

其中，$d_{ik}=d(x_i-v_k)=\sqrt{\sum_{j=1}^{m}(x_{ij}-v_{kj})^2}$，$d_{ik}$是欧几里得距离，用来度量第 $i$ 个样本$x_i$与第 $k$ 个中心点之间的距离；$m$ 是样本的特征数；$b$ 是加权参数，取值范围是$1\le b\le\infty$。这种聚类方法使分类能产生最小的函数值$J_b$，它要求一个样本对于各个聚类的隶属度值和为 1，即：

$$\sum_{j=1}^{c}\mu_j(x_i)=1\,,i=1,2,\dots,n$$

对于类$A_k$的隶属度$\mu_{ik}$和 c 个聚类中心$\{v_i\}$：

$$\mu_{ik}=\frac{1}{\sum_{j=1}^{c}(\frac{d_{ik}}{d_{lk}})^{\frac{2}{b-1}}}$$

设$I_k=\{i|2\le c<n;d_{ik}=0\}$，对于所有的 i 类，$i\in I_k,\mu_{ik}=0$。

$$v_{ij}=\frac{\sum_{k=1}^{n}(\mu_{ik})^b x_{kj}}{\sum_{k=1}^{n}(\mu_{ik})^b}$$

其具体算法步骤为：

*Step1*：为每个聚类确定一个初始聚类中心点；

*Step2*：将数据集中的数据按照欧式距离原则分配到最邻近簇；

*Step3*：使用每个簇中的样本数据均值作为新的聚类中心；

*Step4*：重复步骤 *Step2* 与 *Step3* 直至算法收敛；

*Step5*：结束，得到 $k$ 个结果簇。

其中，每个任务分配的需求点之间不能有交集，即要求每一个需求点只能被分在某一个聚类中。在初始聚类中心选取方法上采用随机选择法，即随机的从所有点中选取 $k$ 个需求点作为初始聚类中心。将所给的地理位置信息进行聚类分析后，再对每个类别进行进一步的细分处理，通过查找每个任务点周边的公交站台数量和地铁站数量，大致分析分布任务点的交通便利程度。

（2）指标的选取

为了更准确的找出其定价规律，再对各项目的标定价格进行定量分析。充分考虑开发商在进行任务定价时需要考虑的各个方面，我们选取一定的指标具体反映其定价规律。

①基本成本费用$\boldsymbol{C}$

对于“拍照赚钱”项目的开发，其根本目的是为开发商赢得一定的利润，由于企业进行生产经营活动的同时必然会消耗一定的资源，换言之开发商的经营活动是以消耗生产资料和劳动力为前提条件的，因此必须在考虑该过程中的资源消耗的基础上找出合理的定价规律。这也是本文在确定其定价规律时需要考虑的首要因素。

②周边会员个数$\boldsymbol{x_1}$

通过对图 2 的分析，可以看出在经纬度近似相同的位置可能发布多个任务。假设各用户的拍照水平相同，类比于供求市场，当同一时间小范围内分配的任务数量多于会员人数，即出现“供大于求”的情况时，会导致部分任务出现“空分配”现象，使得许多任务在该时间内无法完成，此时就需要开发商通过提高任务的定价来刺激会员完成任务的积极性；相反，当地区内会员的人数多于分配的任务数量时，会导致会员的“抢单”现象，这时的任务定价较低，而鉴于此时“供不应求”的情况，较低的定价并不会导致出现过多未完成的任务。

因此，分配的任务周边的会员数量是本文在寻求定价规律时需要考虑的第二个因素。

③周边等同任务数$\boldsymbol{x_2}$

假设各任务难度相同，由图 2 可得，实际情况下由于多个任务的位置比较集中，会员会争相选择相对方便并且定价较高的任务，导致部分任务存在未完成的情况。如任务数量过多，会出现任务过剩的现象；任务数量过少，则会浪费任务分布点的会员资源。因此，每个会员周边的相同任务数也是开发商在定价时需要纳入考虑的因素之一。

④任务点交通便利程度$\boldsymbol{x_3}$

当发布的任务点处于交通便利的道路附近，即周边会员到达任务地点的难度系数较低，完成该任务的可能性也较大；而当任务分配于地势崎岖或交通不发达的地带，会员完成该任务的可能性就较低，因此，任务点周围的交通便利程度是本文在寻找定价规律时考虑的第三个因素，通过任务周边的公交站数量衡量。根据附件一，采用 *K-means* 聚类算法将所给任务分配的地理位置进行聚类。

（3）基于最小二乘回归的定价规律模型的建立

对上述三个指标进行归一化处理，以归一化后的三个指标为自变量，价格为因变量，进行偏最小二乘回归分析。偏最小二乘回归是一种改进的多元线性回归分析的方法。它可以解决多重共线性造成的问题，如进行判定时没有十分可靠的检验方法等。由此解出定价与上述指标之间的相关系数。

设定价与上述四个指标之间的关系为：

$$y = C + \alpha_1 x_1 + \alpha_2 x_2 + \alpha_3 x_3$$

其中，$C$为基本成本费用；$x_1, x_2, x_3$分别为归一化后的周边会员个数、周边相同任务数和任务点交通便利程度；$\alpha_1, \alpha_2, \alpha_3$分别为三个指标的系数。由此得到项目的任务定价规律。

#### 5.2.1.2 任务未完成的原因——*Logistic* 回归分析模型的建立

任务完成情况的取值有两种，完成为1，未完成为0，是典型的 0-1 变量。*Logistic* 回归分析是一种广义的线性回归分析模型，是处理此类问题的典型方法。

本题通过 *Logistic* 回归分析大致了解任务未完成的原因。其基本原理如下所示：

设自变量$X=(x_1,x_2,\dots,x_k)^T$是分组数据，取有限的几个值。研究任务完成的概率$P(y=1|X)$与自变量 *X* 的关系，其 *Logistic* 回归方程为：

$$P(y=1|X)=\frac{e^{\beta_0+\beta_1x_1+\cdots+\beta_kx_k}}{1+e^{\beta_0+\beta_1x_1+\cdots+\beta_kx_k}}$$

其中，$\beta_0,\beta_1,\dots,\beta_k$为类似于多元线性回归模型中的回归系数。

通过分析任务未完成事件与各项指标之间的回归系数，得到导致任务未完成的原因。

### 5.2.2 模型的求解

#### 5.2.2.1 定价规律的求解

首先，根据 *K-means* 聚类分析得到最终的聚类中心如表 1 所示：

**表 1 最终聚类中心**

| | 聚类 | | |
|---|---|---|---|
| | 1 | 2 | 3 |
| 任务gps纬度 | 22.667097654619 | 23.019327858248 | 23.110602419723 |
| 任务gps经度 | 114.041272050886 | 113.725453587602 | 113.227249441987 |

根据表 1，将所有任务点分为 3 类，其最终聚类中心的地理位置坐标分别为：( 22.667097654619°N ，114.041272050886°E ) 、 ( 23.019327858248°N ，113.725453587602°E)、(23.110602419723°N，113.227249441987°E)，中心位置坐标分别位于广州市、深圳市、东莞市。因此，以此为依据将任务点分为三类，每类的任务大致分布在即为广州市、深圳市、东莞市。

其次，以广州市为例，计算周边任务个数和会员个数，并通过 “高德位智”查询附近公交站台数，列出其中 10 组相关数据如表 2 所示：

**表 2 相关数据**

| 任务号码 | 纬度 | 经度 | 标价 | 任务个数 | 会员个数 | 公交站台数 |
|---|---|---|---|---|---|---|
| A0008 | 22.56277 | 113.9565 | 65.5 | 10 | 19 | 30 |
| A0014 | 22.50616 | 113.9314 | 66 | 14 | 20 | 29 |
| A0015 | 22.49962 | 113.9365 | 66 | 12 | 15 | 43 |
| A0028 | 22.54808 | 113.9453 | 66.5 | 11 | 28 | 39 |
| A0046 | 22.78892 | 113.8966 | 66 | 2 | 18 | 37 |
| A0051 | 22.63010 | 114.0638 | 66.5 | 7 | 34 | 40 |
| A0074 | 22.53317 | 114.0831 | 66.5 | 2 | 33 | 23 |
| A0093 | 22.68008 | 113.8347 | 70 | 7 | 18 | 31 |
| A0099 | 22.73335 | 113.8312 | 66 | 4 | 21 | 24 |
| A0108 | 22.70038 | 113.8449 | 66 | 4 | 16 | 52 |

最后，根据上表信息，得到项目定价与各项指标之间的系数如表 3 所示：

**表 3 定价与各项指标之间系数**

| 任务个数 | 会员个数 | 公交站台数 | 基本成本费用 |
|---|---|---|---|
| -0.2209 | -0.2110 | 0.1303 | 67.6317 |

由此可以得到广州市任务定价与四个指标之间的关系为：

$$y=67.6317-0.2209x_1-0.2110x_2+0.1303x_3$$

通过观察任务完成情况分布图、任务定价分布图并结合地图，发现广州市内远离街道、公车站台的位置任务定价较高且无人完成，可能是因为其偏远的地理位置而导致会

员到达任务点存在一定的难度从而使得会员无法完成该项任务。

#### 5.2.2.2 任务未完成的原因的求解

通过之前的计算可以发现任务标价与基本成本费用 $C$ 和任务点与道路距离$x_4$的相关程度较大，而和任务个数和会员个数的相关系数较小。因此，可以用任务标价代替基本成本费用 $C$ 和任务点与道路距离$x_4$，得到各因素的回归系数如表 4 所示：

表 4 各因素回归系数

| 任务个数 | 会员个数 | 任务标价 | 常量 |
|---|---|---|---|
| 0.061 | -0.059 | 0.057 | -2.983 |

根据上表，得到 *Logistic* 回归方程为：

$$\hat{P}_i = \frac{e^{-2.983+0.061x_1-0.059x_2+0.057x_3}}{1+e^{-2.983+0.061x_1-0.059x_2+0.057x_3}}$$

由此可以看出，任务完成的概率与任务个数、会员个数、任务标价之间的回归系数相差不大，即说明任务完成的概率与上述三个指标之间的相关程度近似相同。而任务未完成的可能原因是任务个数分配不合理、会员个数过多或过少以及任务的标价不合理。

### 5.2.3 模型的检验

#### 5.2.3.1 偏最小二乘回归分析的检验

定义偏差系数$\sigma$，其计算公式为：

$$\sigma = \sqrt{\sum_{i=1}^{n}(y_{i回归} - y_{i真实})^2}$$

其中，$y_{i回归}$是通过偏最小二乘回归得到的该任务最终定价；$y_{i真实}$是该任务的真实定价。

计算附件一中 835 个有效数据的偏差系数，其值为 0.5346。由于偏差系数值较小，可以大致认为上述求解结果是较为可信的。

#### 5.2.3.2 Logistic 回归的检验

定义 *Logistic* 回归分析的偏差率为$\varepsilon$，其计算公式如下：

$$\varepsilon = \frac{任务完成情况不同的个数}{总任务数量} \times 100\%$$

其中，任务完成情况不同的个数指的是新定价方案实施后的任务完成情况与原定价方案下的任务完成情况的差异个数。

经过计算，得到 *Logistic* 回归分析的偏差率$\varepsilon$为 5.221%。可以看出采用 *Logistic* 回归分析法造成的误差较小，模型可信度较高。

## 5.3 问题二的模型建立与求解

新的定价方案即要求任务完成数量的增加，首先，通过分析会员心理与实际现实情

况，分别从会员的意愿程度和任务点与会员之间的距离两个角度出发，类比于出租车定价方案分别得到不同距离对应的不同任务定价。其次，以最多任务完成数量为目标，建立单目标规划模型，得出新的任务定价方案。最后，定义任务完成度指标，与原定价方案比较，分析两者的优劣。

### 5.3.1 模型的建立

为项目设计新的任务定价方案，即要求新的定价方案的任务完成数量与原定价方案相比有一定程度的提升。

从“效用价值论”的角度来说，从任务对会员欲望的满足能力出发，从主观心理角度解释定价及其形成过程，即根据会员的取向定价更能保证定价的合理性。由于分配的任务与周围的会员人数可能存在多对多的关系，因此会员是否会完成该任务实际取决于该会员完成该项任务的意愿程度。由此，本文提出会员意愿程度$m_{ij}$，其基本表达式为：

$$m_{ij} = \frac{u_{ij}}{p_i}$$

其中，$u_{ij}$为第 j 个会员对于第 i 项任务的期望定价；$p_i$是第 i 项任务的实际定价。

记第 i 项任务的完成情况为$n_i$，则：

$$n_i = \begin{cases} 1 & \exists u_{ij} \geq 1 \\ 0 & \text{其他} \end{cases}$$

上式说明当该项任务的实际定价高于会员的期望定价，则该项任务被完成的机率较大，视为完成；反之则该项任务被完成的机率较小，视为无人愿意完成。而会员对任务期望定价还受到会员所在位置距任务点的远近的影响，由于生活服务的范围是会员自身范围的 *3km* 之内[2]，鉴于此，本文依据任务分布点距会员的远近程度进行讨论：

（1）任务点与会员的距离小于 *3km*

根据之前的分析，可以发现开发商在每个任务点周围还会发布其他任务，任务点密集的地域会员密度也相对较大。为了整体反映一定地域内的任务标价，在这里取该地域内发布的所有任务的标价均值作为该地域任务标价的总体反映，通过分析可以得到，当分配任务数大于周边会员时，任务定价随着会员人数的增加呈上升趋势；当分配任务数小于周边会员时，任务定价呈下降趋势。考虑到实际情况，任务的定价不可能无限减小，因此我们初步推断该种情况下的任务定价$A_i$满足：

$$A_i = \bar{A} \times (a - e^{-\frac{R}{M}})$$

其中，$\bar{A}$为该地域所有任务的定价均值；$R$ 为该地域的任务数；$M$ 为该地域的会员数。$R/M$实际为一定范围内的供求比，这里提出一种基于经验模型的定价方法，该方法通过过去的经验数据建立模型，利用已经成功实施的定价经验指导当前的定价决策[3]。则当$R/M = 1$时，即供求平衡时，任务定价与定价均值相等，即：

$$a = 1 + e^{-1}$$

（2）任务点与会员的距离大于 *3km*

类比于出租车定价方案，当两者相距 *3km* 以上时开发商的定价方案内存在一个基础价格 $b$，即为该项任务的最低价格。当会员与任务点的距离逐渐加大时，开发商会在基础价格的基础上往上加价以保证任务的完成，距离越大，定价越高。由此得到该种情况下的任务定价$A_i$为：

$$A_i = b \times (l_{ij} - 3)$$

其中，$l_{ij}$为会员与任务点之间的距离。

由此，建立单目标规划模型。

目标函数为：

$$Z = max \sum_{i=1}^{N} n_i$$

约束条件为：

$$\begin{cases} \sum_{i=1}^{N} p_i \leq \sum_{i=1}^{N} m_{ij} \\ p_i \geq 0 \\ n_i = \begin{cases} 1 & \exists u_{ij} < p_i \\ 0 & \text{其他} \end{cases} \end{cases}$$

由此得出新的任务定价方案。

接着，为了准确的描述任务的完成情况，定义任务完成度指标$\gamma_i$，其计算公式为：

$$\gamma_i = \frac{\text{完成任务数}}{\text{总任务数}}$$

将新定价方案的任务完成度指标$\gamma_2$与原定价方案的任务完成度指标$\gamma_1$相对比，分析新定价方案的优势。

### 5.3.2 模型的求解与检验

以广州市为例，按照随机原则挑选 10 个任务的定价与完成情况，如表 5 所示：

表 5 新方案下任务的完成情况

| 任务编号 | 任务原标价 | 任务原完成情况 | 任务新标价 | 任务现完成情况 |
|---|---|---|---|---|
| A0003 | 65.5 | 0 | 53.9 | 1 |
| A0015 | 66 | 1 | 71.5 | 1 |
| A0018 | 66 | 0 | 73.2 | 1 |
| A0027 | 65.5 | 1 | 60.8 | 1 |
| A0034 | 66 | 1 | 53.1 | 1 |
| A0040 | 66 | 0 | 66.2 | 1 |
| A0044 | 69 | 0 | 62.4 | 0 |
| A0050 | 70 | 0 | 59.2 | 0 |
| A0066 | 67 | 0 | 68.7 | 0 |
| A0078 | 66.5 | 1 | 50.4 | 1 |

根据表 5 发现，新的定价方案使得部分原未完成的任务被完成，适当的提高或降低标价可以在一定程度上改变任务完成率。如任务 A0003 的定价由原来的 65.5 降低到 53.9；而任务 A0018 和任务 A0040 的定价分别由原来的 66 提高到 53.9、66.2。据此也可以得到新的定价方案与原方案相比完成的任务数量较多，即任务完成率较高。最后得出，新方案下任务的完成率$\gamma_2 = 0.677844$，原方案下的任务完成率$\gamma_1 = 0.625598 < \gamma_2$，新定

价方案与原方案相比完成率有所提高，但提高幅度并不显著。

### 5.4 问题三的模型建立与求解

由于多个任务集中而导致的会员争相选择的情况，通常采用任务联合打包的方法解决此问题。任务打包方式多样，密集区各任务距离较近，并没有明显的界限，考虑到该种情况，建立基于遗传模拟退火算法的模糊 *C*-均值聚类模型，克服传统聚类算法的缺点，对任务进行打包；其次，考虑用户的预定限额、信誉、最早任务选取时间和商家的成本问题，建立多目标规划模型，求得新的定价方案并将任务完成度进行比较。最后，通过调节打包的数量分别计算各个任务点的任务完成情况，确定最优打包个数，并分析其对任务完成情况的影响。

#### 5.4.1 模型的建立

##### 5.4.1.1 基于遗传模拟退火算法的模糊 C-均值聚类模型的建立

由于现实中存在着多个任务点位置比较集中而导致会员争相选择的现象，一种考虑是将这些任务联合在一起打包发布。根据问题一中对任务点的 *K-means* 聚类分析的结果，本文对会员也采取相同的聚类分析方法，并在各类的内部进行研究，计算其聚类中心。利用公式反复修改聚类中心、数据隶属度和进行分类，当算法收敛时，理论上得到了各类的聚类中心以及各个样本对于各模式类的隶属度，从而完成聚类划分。由于会员距离其他城市距离较远，本题对各个城市内的任务分别打包，但是由于该算法是一种局部搜索算法，且对聚类中心的初值十分敏感，如果初值选择不当，它会收敛到局部极小点。因此，本文建立基于遗传模拟退火算法的模糊 *C*-均值聚类模型[4]，其过程如下所示：

*Step1*：初始化控制参数：种群个体大小 *sizepop*，最大进化次数 *MAXGEN*，交叉概率$P_e$，变异概率$P_m$，退火初始温度$T_0$，温度冷却系数 $k$，终止温度$T_{end}$；

*Step2*：随机初始化 $c$ 个聚类中心，并生成初始种群 *Chrom*，对每个聚类中心计算各样本的隶属度，以及每个个体的适应度值$f_i$，其中$i = 1,2,\dots,sizepop$；

*Step3*：设循环计数变量 *gen=0*；

*Step4*：对群体 *Chrom* 实施选择、交叉和变异等遗传操作，对新产生的个体计算 $c$ 个聚类中心，各样本的隶属度，以及每一个体的适应度值$f_i'$。若$f_i' > f_i$，则以新个体替换旧个体；否则，以概率$P = exp((f_i - f_i')T)$接受新个体，舍弃旧个体；

*Step5*：若 *gen<MAXGEN*，则 *gen=gen+1*，转至 *Step4*；否则，转至 *Step6*；

*Step6*：若$T_i < T_{end}$，则算法成功结束，返回全局最优解；否则，执行降温操作$T_{i+1} = kT_i$，转至 *Step3*。

##### 5.4.1.2 多目标规划模型的建立

由于一个任务包内的任务仅由一位会员完成，且包内的任务分布的比较密集,所以在一定程度上能够减少不打包情况下包内任务的总价，因此，重新定义任务定价，记包内任务总价$A_i'$为：

$$A_i' = b \times \left(l_{ij}' - 3\right) + b' \times \mathrm{n}$$

其中，$l_{ij}'$为会员与任务包的聚类中心之间的距离，n为任务包内任务的个数。

针对任务定价的变化，改进问题二的目标规划模型，建立多目标规划模型。

根据上述步骤得到各城市打包后任务的新聚类中心，然后考虑会员的限额、信誉以及商家的成本问题，将问题二的单目标规划模型修改为多目标规划模型。

目标 1——任务完成数量最多:

$$max \sum n_i$$

目标 2——总定价额度最小：

$$min \sum p_i$$

约束条件如下：

$$\begin{cases} \sum_{j=1}^{n} n_{ij} \le 1 \\ \sum_{i=1}^{n} p_i \le \sum_{i=1}^{n} m_{ij} \\ p_i \ge 0 \\ \sum_{j=1}^{n} y_{ij} \le y_{max} \\ n_{ij} = \begin{cases} 1 & \exists u_{ij} < p_i \\ 0 & \text{其他} \end{cases} \\ y_{max} = \begin{cases} y_{ij} & n_{ij} = 1 \\ 0 & n_{ij} = 0 \end{cases} \end{cases}$$

其中，$n_{ij}$代表第 i 项任务由第 j 个会员的完成情况，其可能的取值为 0 和 1；$y_{ij}$为每位会员预定的任务数量；$y_{max}$为每位会员的任务限额。采用求解目标规划算法中的序贯算法对上式进行求解。

最后，利用问题二中的任务完成度指标$\gamma_i$，分析采用此种改进的方法进行定价与原定价方案相比的优劣。

分析打包方案对于任务完成情况的影响，通过调节打包的数量分别计算各任务点的任务完成情况，确定最优的打包个数，根据会员距聚类中心的距离重新定价。

### 5.4.2 模型的求解与检验

首先，参照问题一中 *K-means* 聚类分析的求解结果，将所给任务点聚类为广州、深圳、东莞三类。分别在这三个类中进行基于遗传模拟退火算法的聚类分析，对任务进行打包。以广州市为例，将任务重新编号，得到打包后聚类中心的部分结果如下：

**表 6 聚类中心求解部分结果**

| 打包任务编号 | 包含任务编号 | 周边任务数量 | 聚类中心 |
|---|---|---|---|
| 1 | A0542、A0829 | 2 | (23.1779°N, 112.8773°E) |
| 2 | A002、A0110、A0364、A0446 | 4 | (22.6906°N, 113.9407°E) |
| 3 | A0486、A0489、A0513、A0516、A0755 | 5 | (22.7886°N, 113.7584°E) |
| 4 | A0773 | 1 | (23.0733°N, 114.0234°E) |

其次，通过调整打包的个数，得到广州市的任务完成率与任务打包个数之间的关系如图 5 所示：

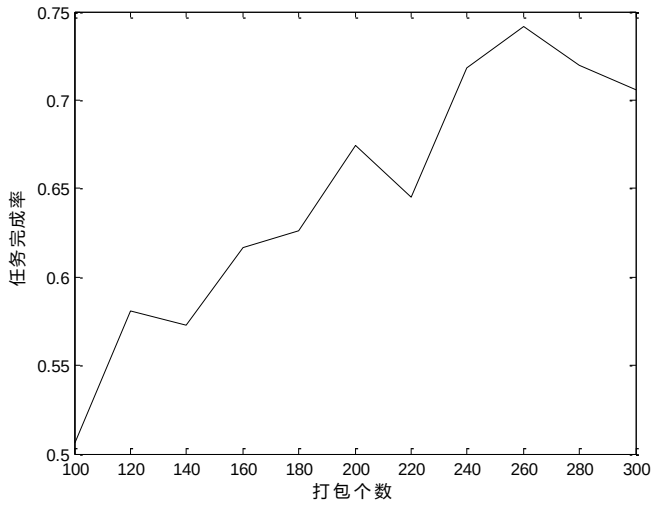


图 5 广州市任务完成率

由图 5 可得，广州市的任务完成率随着打包个数的增加大致呈增长趋势，并在打包个数为 260 个时出现峰值，其任务完成率约为 0.74；打包个数为 100 个时任务完成率较低，约为 0.51。由此可以得到，打包数量过少时任务的完成情况并不乐观，即对最终任务完成情况造成负面影响；而打包数量在 200 个以上时，任务完成率较高，对最终情况的影响较好，但是可能出现开发商的成本较高。

同理，深圳市、东莞市的最优任务打包数量为 260 个、220 个，其任务完成率随着打包个数增加的趋势与广州市大致相同。因此，较优的任务完成率需要开发商分配合理的打包个数。

## 5.5 问题四的模型建立与求解

由于略高与略低的价格都会影响会员的积极性，都一定程度影响任务的完成情况，因此找到价格的平衡点十分重要。但是，开发商在进行价格定位时，往往难以从任务本身所具有的价值上做出恰当的评判，因此需要通过对以往的定价方案与该定价方案的实施结果的研究学习，预测出新的定价方案，不断进行动态修正与完善。由于拥有的数据量小、维度少，因此采用支持向量机回归预测的方法对原始数据集中的相关定价信息进行学习训练并动态预测出最初的定价方案，并定义实施效果评价指标用以评价初始定价方案的实施效果。

### 5.5.1 模型的建立

开发商在进行价格定位时，往往难以从任务本身所具有的价值上做出恰当的评判。在基础价格之上，略高与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与完成时间。因此，单单从任务的实时完成情况进行定价是不准确的，这种现实情况不能看作是价格的变化规律，而是定价不符合规律的阶段性特征。由于每个会员完成任务的成本不同，设置统一的价格也会影响会员的积极性，因此，找到价格的平衡点十分重要。

在成本可按的情况下，更快更好的完成全部任务是最好的结果。对于以上纳入考虑的复杂相关关系，需要通过对以往的定价方案与该定价方案的实施结果的研究学习，预

测出新的定价方案，不断进行动态修正与完善。因此本文采用支持向量机回归预测的方法给出新的项目定价。其基本原理如下[5]：

对于给定的包含 $n$ 个数据的训练样本集$\{(x_1,y_1),\dots,(x_n,y_n)\}\subset R^m\times R, R^m$为 $m$ 维输入空间，支持向量机回归就是采用非线性映射函数$\varphi(x)$，将输入样本集映射到高维的特征空间。

设在高维特征空间中建立的线性回归函数为：

$$y=f(x)=\omega^T\Phi(x)+b$$

其中，$\omega$为权值，$b$ 为偏置项。对于任务分配相关数据，支持向量机回归预测可以表示为：

$$\begin{cases}\min\dfrac{1}{2}|\omega|^2+C\sum_{i=1}^{l}(\xi_i+\xi_i^*)\\ s.t.\begin{cases}y_i-\omega\Phi(x_i)-\mathrm{b}\le\varepsilon+\xi_i\\ -y_i+\omega\Phi(x_i)+\mathrm{b}\le\varepsilon+\xi_i^*\\ \xi_i\ge 0,\xi_i^*\ge 0\end{cases}\end{cases}$$

其中，$C$ 为惩罚系数，用于调整置信范围和经验误差之间的平衡。进行求解时，引入 *Largrange* 函数，并用核函数$K(x_i,x_j)=\varphi(x_i)\varphi(x_j)$代替特征空间中的内积计算，将原问题转换为对偶形式：

$$\max_{\alpha,\alpha^*}\left[-\frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}(\alpha_i-\alpha_i^*)(\alpha_j-\alpha_j^*)K(x_i,x_j)-\sum_{i=1}^{l}(\alpha_i+\alpha_i^*)\varepsilon+\sum_{i=1}^{l}(\alpha_i-\alpha_i^*)y_i\right]$$

$$s.t.\begin{cases}\sum_{i=1}^{l}(\alpha_i-\alpha_i^*)=0\\ 0\le\alpha_i\le C\\ 0\le\alpha_i^*\le C\end{cases}$$

综上可以得到支持向量机决策函数：

$$y=\sum_{i=1}^{n}(\alpha_i^*-\alpha_i)K(x_i,x)+b$$

通过引入核函数的方法，在保证原有计算复杂度的同时有效的避免了维数灾难的问题，此处选用最常用的径向基(*RBF*)核函数：

$$K(x,y)=exp(-\frac{|x-y|^2}{\sigma^2})$$

由于定价方案会根据地点的不同而产生变化，同一任务点的定价也应根据外部环境对任务的作用采取分级定价，这实际上是一种动态定价的过程。根据大数据机器学习的思想，通过分析历史数据集中的数据并以此为基础预测新数据的方法是比较合理可行的。当任务的定价过低，会员参与度较低，会员接受该任务的可能性就较小，出现任务分配过久而无人完成的现象；而任务定价过高则会引起会员争相完成，若任务选择过快也侧面说明了该项任务的定价过高，那么开发商就会在下次发布任务前对该任务的定价做出相应的调整。出于对以上两种情况的考虑，定义会员参与度$\mu$，其与任务定价的关系可以表示为：

$$y = a + b\frac{1}{\mu}$$

其中，*a,b* 为常系数。

然后，提出一种基于机器学习来预测定价的方法，对数据集进行学习后给出一个新的定价，并计算该定价下会员的参与度；若参与度过低则通过参与度任务定价的关系一定程度上调高定价，反之则调低；将预测数据加入隶属数据集进行再次训练学习预测，不断进行动态修正与完善，得到最终的定价方案。其基本步骤如下所示：

*Step1*：获取关于定价和会员的各项信息，并进行归一化处理；

*Step2*：基于获取的信息产生基于机器学习的定价模型的预测样本所包括的属性特征，并通过将产生的属性特征分别与待定价项的候选价格进行组合产生一条预测样本；

*Step3*：通过使用产生的预测样本预估一定时间内会员的参与度；

*Step4*：基于预估的会员参与度确定一个新的定价方案；

*Step5*：当新训练集的数据持续输入时重复 *Step3* 和 *Step4*，动态调整定价方案，完成任务定价的动态预测，得到最终的预测结果。

最后，定义方案实施效果评价指标$\kappa$，其表达式如下：

$$\kappa = \sqrt{(\mu - 1)^2 + (p - p^*)^2}$$

其中，$\mu$是会员参与度；$p$是该定价方案下的总花费；$p^*$是该定价方案下的最优花费数额。$\kappa$的数值越小，表明该定价方案与最优定价方案的吻合程度越高，以此来评价该定价方案的实施效果。

### 5.5.2 模型的求解与检验

根据附件三得到会员与任务的位置分布图如图 6 所示：

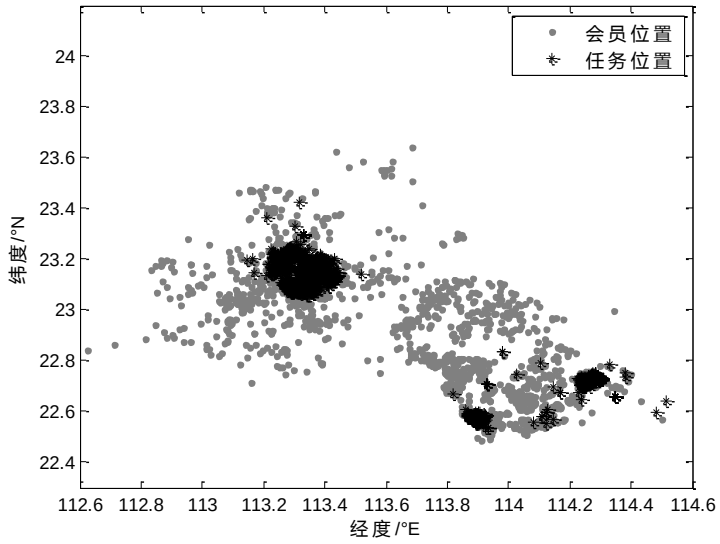


图 6 会员与任务位置分布图

由图 6 可得，任务位置多集中在两个区域，而会员位置分布较广泛，散布于集中任务点的周围，因此极有可能会出现由于“供小于求”而导致会员争相选择任务的情况。通过查阅地图，图中任务聚集的位置大致位于广州市和深圳市。由问题三的答案可得，最优打包个数约占总数的 40%~60%，此时每个包中包含的任务数量多为 1~10 个不等，因此，本题设置打包的总个数为 900 个。利用问题一的聚类算法将任务分为两类，打包个数分别为 463 个、437 个。

根据调研，得到 638 组较优的历史数据。其中 600 组作为训练集、38 组作为测试集，以任务个数、会员人数、任务标价作为输入，会员满意度为输出，迭代次数为 100 次，对定价方案进行支持向量机的动态回归预测。得到部分结果展示如下：

表 7 支持向量机回归预测部分结果

| 任务编号 | 会员人数 | 任务个数 | 任务标价 | 会员满意度 |
|---|---|---|---|---|
| C0015 | 10 | 11 | 70 | 1.537019 |
| C0024 | 1 | 8 | 72 | 1.505299 |
| C 0037 | 5 | 15 | 66 | 0.598084 |
| C 0042 | 3 | 2 | 65.5 | 0.59025 |
| C0049 | 2 | 6 | 65.5 | 0.583991 |
| C0056 | 12 | 14 | 65.5 | 0.576013 |
| C0082 | 13 | 6 | 70 | 1.401619 |
| C0087 | 3 | 2 | 85 | 1.390435 |
| C0130 | 8 | 11 | 50.4 | 1.297037 |

最后得出，聚类后两城市的方案实施效果评价指标值分别为 0.1983、0.0532。可以看出其实施效果评价指标的数值较小，即制定的新定价方案较为合理。

## 六、模型的评价与改进

### 6.1 模型的优点

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；问题四中采用支持向量机回归预测的方法，把线性的回归转化为非线性，具有较强的非线性学习能力，使得预测从一定层面上避免了维数灾难的问题；本题对任务定价的动态预测方法，充分利用了历史数据，得到了不断更新后的实时定价，可信度较高。

### 6.2 模型的改进与推广

由于附加所给的相关信息的数量和维度较少，文中使用的支持向量机进行回归预测会具有一定的不准确性。要想进行长时间的定价预测，需要足够多的、高维度的数据集。从会员的角度来说，如会员抢单的具体时间、会员的活跃度以及行动路径等；从任务的角度来说，任务完成的时间、任务难易程度等信息，都是进行准确预测的必要条件。最后通过对大量数据的收集，运用神经网络对任务定价进行预测，在一定量大数据的积累下可得到不同时空下更为准确的定价。支持向量机的回归预测近年来在很多方面得到应用，例如市场预测、股价预测、财务危机预测、交通流量预测、基因预测等领域。而大数据机器学习的方法可以为欺诈检测、需求预测、点击预测和其他数据密集型分析创建预测应用程序，是互联网加大数据时代下的发展趋势，具有广阔的发展前景。

## 参考文献

[1]黎珍惜,黎家勋.基于经纬度快速计算两点间距离及测量误差[J].测绘与空间地理信息,2013,36(11):235-237
[2]赵兴龙.基于K-means遗传算法的众包配送网络优化研究[D].北京交通大学,2016.25-35
[3]毛可.软件众包任务的定价模型与人员匹配方法研究及工具实现[D].中国科学院大学,2014
[4]何云斌,张晓瑞,万静,李松.基于改进遗传模拟退火 K-means 的心电波形的分类研究[J].计算机应用研究,2014,31(11):3328-3332
[5]安吉宇,杨瑜,刘志中,申艳梅.基于支持向量机与事例推理的 Web 服务 QoS 动态预测方法研究[J].小型微型计算机系统.2015,36(11):2521-2524

# 附 录

## 附录一 ：问题一代码

```
/t1_huatu.m /问题一画图
%%
%任务完成情况图
B=xlsread('C:\Users\Administrator\Desktop\ 题 目 \B\ 附 件 一 ： 已 结 束 项 目 任 务 数据.xls','t_tasklaunch','B2:E836');
B_x=B(:,1);
B_y=B(:,2);
B_z=B(:,3);
B_FL1=zeros(522,4);
B_FL0=zeros(313,4);
j1=1;j2=1;
for i=1:835
    if B(i,4)==1
        B_FL1(j1,:)=B(i,:);
        j1=j1+1;
    else
        B_FL0(j2,:)=B(i,:);
        j2=j2+1;
    end
end
plot(B_FL1(:,2),B_FL1(:,1),'k.');
hold on
plot(B_FL0(:,2),B_FL0(:,1),'*','color',[0.5 0.5 0.5]);
 legend('未完成','已完成');
%画任务位置图
figure
plot(B_y,B_x,'k*');
hold on
%%
%画会员位置图
c=xlsread('C:\Users\Administrator\Desktop\题目\B\附件二：会员信息数据.xlsx','GPS','A1:B1877');
c_x=c(:,1);
c_y=c(:,2);
plot(c_y,c_x,'.','color',[0.5 0.5 0.5]);axis([112.6 114.6 22.3 24.2]);
legend('任务位置','会员位置');
hold on
%%
%画价格分布图
D=xlsread('C:\Users\Administrator\Desktop\ 题 目 \B\ 附 件 一 ： 已 结 束 项 目 任 务 数据.xls','t_tasklaunch','B2:D863');
```

```
x=D(:,1);y=D(:,2);
[X,Y,Z]=griddata(x,y,B_z,linspace(22.493,23.879)',linspace(112.683,114.494),'v4');%插值
figure,surf(X,Y,Z),colormap(flipud(gray))%三维曲面
%t1_num.m 计算任务方圆 3 公里内任务与会员个数
B=xlsread('C:\Users\Administrator\Desktop\ 题 目 \B\ 附 件 一 ： 已 结 束 项 目 任 务 数 据.xls','t_tasklaunch','B2:C836');
A=xlsread('C:\Users\Administrator\Desktop\题目\B\附件二：会员信息数据.xlsx','GPS','A1:B1877');
m=1877;%会员数
n=835;%任务数
d=zeros(n);
num_1=zeros(n,1);
for i=1:n
    for j=1:n
        d(i,j)=distance(B(i,1),B(i,2),B(j,1),B(j,2))/180*pi*6378.1;
        if d(i,j)<=3
            num_1(i)=num_1(i)+1;
        end
    end
end

num_2=zeros(n,1);
d1=zeros(n);
for i=1:n
    for j=1:m
        d1(i,j)=distance(B(i,1),B(i,2),A(j,1),A(j,2))/180*pi*6378.1;
        if d1(i,j)<=3
            num_2(i)=num_2(i)+1;
        end
    end
end

%t1_huigui.m  偏最小二乘回归
clc,clear
format long g
B=xlsread('C:\Users\Administrator\Desktop\题目\B\附件一：已结束项目任务数据.xls','Sheet2','B2:H11');
X0_1=B(:,5);X0_2=B(:,6);X0=[X0_1,X0_2,B(:,7)];
x=1:10;
Y0=B(:,3);
ab0=[X0,Y0];
mu=mean(ab0);sig=std(ab0); %求均值、标准差
ab=zscore(ab0); %数据标准化
a=ab(:,[1:3]);b=ab(:,4);
ncomp=2;
[XL,YL,XS,YS,xishu1,PCTVAR,MSE,stats]=plsregress(a,b,ncomp)
```

```
n=size(a,2); m=size(b,2); %n 是自变量的个数,m 是因变量的个数
xishu12(1,:)=mu(n+1:end)-mu(1:n)./sig(1:n)*xishu1([2:end],:).*sig(n+1:end); %原始数据回归方程的常数项
xishu12([2:n+1],:)=(1./sig(1:n))'*sig(n+1:end).*xishu1([2:end],:) %计算原始变量 x1,...,xn 的系数
format
y=xishu12(1,1)+xishu12(2,1).*X0_1+xishu12(3,1).*X0_2+xishu12(4,1).*B(:,7);
%%
%误差
wucha=0;
for i=1:10
    wucha=wucha+(y(i)-Y0(i))^2;
end
wucha=sqrt(wucha)/10
```

**附录二　问题二代码**

```
/GH_2.lg4/
model:
sets:
variable/1..835/:a;   !定价;
num/1..835/:b; !任务数是否完成;
hy/1..1877/; !会员个数;
links(num,hy):qw,n; !n 为完成度;
endsets

data:
!期望价格;
qw=@file(data_2.txt);
enddata
max=@sum(num(i):b(i));
@sum(variable(j):a(j))<=180;   !总价不超过原先总价;
@for(links(i,j):n(i,j)=(@if((a(i)/qw(i,j))#ge#1,1,0)));
@for(num(i):b(i)=(@if(@max(links(i,j):n(i,j))#eq#1,1,0)));
End
```

**附录三　问题三代码**

```
/ FCMfun.m/
function [obj,center,U]=FCMfun(X,cluster_n,center,options)
%% FCM 主函数
% 输入
% X：样本数据
%cluster_n：聚类数
%    center：初始聚类中心矩阵
%   options：设置幂指数，最大迭代次数，目标函数的终止容限
% 输出
```

```
%       obj：目标输出 Jb 值
% center：优化后的聚类中心
%          U：相似分类矩阵
X_n=size(X,1);
in_n=size(X,2);
b=options(1);                % 加权参数
max_iter=options(2);         % 最大迭代次数
min_impro=options(3);        % 相邻两次迭代最小改进（用来判断是否提前终止）
obj_fcn=zeros(max_iter,1);   % 初始化目标值矩阵
U = initFCM(X,cluster_n,center,b);                % 初始化聚类相似矩阵
% 主函数循环
for i = 1:max_iter,
    [U, center,obj_fcn(i)]=iterateFCM(X,U,cluster_n,b);
    % 核对终止条件
    if i > 1
        if abs(obj_fcn(i) - obj_fcn(i-1)) < min_impro, break; end,
    end
end
iter_n = i;% 真实迭代次数
obj_fcn(iter_n+1:max_iter)=[];
obj=obj_fcn(end);

/GAFCM.m/
% 遗传算法改进的模糊 C-均值聚类
function [BESTX,BESTY,ALLX,ALLY]=GAFCM(K,N,Pm,LB,UB,D,c,m)
%% 此函数实现遗传算法，用于模糊 C－均值聚类
%% 输入参数列表
% K          迭代次数
% N          种群规模，要求是偶数
% Pm         变异概率
% LB         决策变量的下界，M×1 的向量
% UB         决策变量的上界，M×1 的向量
% D          原始样本数据，n×p 的矩阵
% c          分类个数
% m          模糊 C 均值聚类数学模型中的指数
%% 输出参数列表
% BESTX      K×1 细胞结构，每一个元素是 M×1 向量，记录每一代的最优个体
% BESTY      K×1 矩阵，记录每一代的最优个体的评价函数值
% ALLX       K×1 细胞结构，每一个元素是 M×N 矩阵，记录全部个体
% ALLY       K×N 矩阵，记录全部个体的评价函数值
%% 第一步：
M=length(LB);%决策变量的个数
%种群初始化，每一列是一个样本
farm=zeros(M,N);
```

```
for i=1:M
    x=unifrnd(LB(i),UB(i),1,N);
    farm(i,:)=x;
end
%输出变量初始化
ALLX=cell(K,1);%细胞结构，每一个元素是 M×N 矩阵，记录每一代的个体
ALLY=zeros(K,N);%K×N 矩阵，记录每一代评价函数值
BESTX=cell(K,1);%细胞结构，每一个元素是 M×1 向量，记录每一代的最优个体
BESTY=zeros(K,1);%K×1 矩阵，记录每一代的最优个体的评价函数值
k=1;%迭代计数器初始化
%% 第二步：迭代过程
while k<=K
%% 以下是交叉过程
    newfarm=zeros(M,2*N);
    Ser=randperm(N);%两两随机配对的配对表
    A=farm(:,Ser(1));
    B=farm(:,Ser(2));
    P0=unidrnd(M-1);
    a=[A(1:P0,:);B((P0+1):end,:)];%产生子代 a
    b=[B(1:P0,:);A((P0+1):end,:)];%产生子代 b
    newfarm(:,2*N-1)=a;%加入子代种群
    newfarm(:,2*N)=b;
    for i=1:(N-1)
        A=farm(:,Ser(i));
        B=farm(:,Ser(i+1));
        P0=unidrnd(M-1);
        a=[A(1:P0,:);B((P0+1):end,:)];
        b=[B(1:P0,:);A((P0+1):end,:)];
        newfarm(:,2*i-1)=a;
        newfarm(:,2*i)=b;
    end
    FARM=[farm,newfarm];
%% 选择复制
    SER=randperm(3*N);
    FITNESS=zeros(1,3*N);
    fitness=zeros(1,N);
    for i=1:(3*N)
        Beta=FARM(:,i);
        FITNESS(i)=FIT(Beta,D,c,m);
    end
    for i=1:N
        f1=FITNESS(SER(3*i-2));
        f2=FITNESS(SER(3*i-1));
        f3=FITNESS(SER(3*i));
```

```
        if f1<=f2&&f1<=f3
            farm(:,i)=FARM(:,SER(3*i-2));
            fitness(:,i)=FITNESS(:,SER(3*i-2));
        elseif f2<=f1&&f2<=f3
            farm(:,i)=FARM(:,SER(3*i-1));
            fitness(:,i)=FITNESS(:,SER(3*i-1));
        else
            farm(:,i)=FARM(:,SER(3*i));
            fitness(:,i)=FITNESS(:,SER(3*i));
        end
    end
    %% 记录最佳个体和收敛曲线
    X=farm;
    Y=fitness;
    ALLX{k}=X;
    ALLY(k,:)=Y;
    minY=min(Y);
    pos=find(Y==minY);
    BESTX{k}=X(:,pos(1));
    BESTY(k)=minY;
    %% 变异
    for i=1:N
        if Pm>rand&&pos(1)~=i
            AA=farm(:,i);
            BB=GaussMutation(AA,LB,UB);
            farm(:,i)=BB;
        end
    end
    disp(k);
    k=k+1;
end
%% 绘图
BESTY2=BESTY;
BESTX2=BESTX;
for k=1:K
    TempY=BESTY(1:k);
    minTempY=min(TempY);
    posY=find(TempY==minTempY);
    BESTY2(k)=minTempY;
    BESTX2{k}=BESTX{posY(1)};
end
BESTY=BESTY2;
BESTX=BESTX2;
plot(BESTY,'-ko','MarkerEdgeColor','k','MarkerFaceColor','k','MarkerSize',2)
```

```
ylabel('函数值')
xlabel('迭代次数')
grid on

/ initFCM.m/
function U=initFCM(X,cluster_n,center,b)
%% 初始化相似分类矩阵
% 输入
% X：样本数据
%cluster_n：聚类数
% center：初始聚类中心矩阵
% b：设置幂指数
% 输出
% U：相似分类矩阵
dist=distfcm(center,X);          % 求出各样本与各聚类中心的距离矩阵
%% 计算新的 U 矩阵
tmp=dist.^(-2/(b-1));
U=tmp./(ones(cluster_n,1)*sum(tmp));

/iterateFCM.m/
function [U_new,center,obj_fcn]=iterateFCM(X,U,cluster_n,b)
%% 迭代
% 输入
% X：样本数据
% U：相似分类矩阵
%cluster_n：聚类数
% b：幂指数
% 输出
%obj_fcn：当前目标输出 Jb 值
% center：新的的聚类中心
%U_new：相似分类矩阵
mf=U.^b;          % 指数修正后的 mf 矩阵
center=mf*X./((ones(size(X,2),1)*sum(mf'))'); % 新的聚类中心
%% 目标值
dist=distfcm(center,X);          % 求出各样本与各聚类中心的距离矩阵
obj_fcn=sum(sum((dist.^2).*mf));  % 目标函数值
%% 计算新的 U 矩阵
tmp=dist.^(-2/(b-1));
U_new=tmp./(ones(cluster_n,1)*sum(tmp));

/ ObjFun.m/
function [Jb,center,U]=ObjFun(X,cn,V,options);
%% 计算种群中每个个体的目标值
% 输入
```

```
% X：样本数据
% cn：聚类数
% V：所有的初始聚类中心矩阵
% options：设置幂指数，最大迭代次数，目标函数的终止容限
%  输出
% Jb：各个体的目标输出
% center：优化后的各个体的聚类中心
% U：各样本的相似分类矩阵
[sizepop,m]=size(V);
ch=m/cn;
Jb=zeros(sizepop,1);
center=cell(sizepop,1);
U=cell(sizepop,1);
for i=1:sizepop
    v=reshape(V(i,:),cn,ch);
    [Jb(i),center{i},U{i}]=FCMfun(X,cn,v,options);
end

/JL_3.m/
clc
clear all
close all
X=load('C:\Users\Desktop\dm\fj3_3.txt');
m=size(X,2);% 样本特征维数
%  中心点范围[lb;ub]
lb=min(X);
ub=max(X);
%% 模糊 C 均值聚类参数
%  设置幂指数为 3，最大迭代次数为 20，目标函数的终止容限为 1e-6
options=[3,20,1e-6];
%  类别数 cn
cn=200;
for i=cn-100:20:cn+100

%% 模拟退火算法参数
q =0.8;        % 冷却系数
T0=100;       % 初始温度
Tend=99.999;   % 终止温度
%% 定义遗传算法参数
sizepop=10;                    %个体数目
MAXGEN=100;                     %最大遗传代数
NVAR=m*cn;                      %变量的维数
PRECI=10;                       %变量的二进制位数
pc=0.7;
```

```
pm=0.01;
trace=zeros(NVAR+1,MAXGEN);

%建立区域描述器
FieldD=[rep([PRECI],[1,NVAR]);rep([lb;ub],[1,cn]);rep([1;0;1;1],[1,NVAR])];
Chrom=crtbp(sizepop, NVAR*PRECI); % 创建初始种群
V=bs2rv(Chrom, FieldD);
ObjV=ObjFun(X,cn,V,options); %计算初始种群个体的目标函数值
T=T0;
while T>Tend
    gen=0;                                                    %代计数器
    while gen<MAXGEN                                          %迭代
        FitnV=ranking(ObjV);                %分配适应度值        SelCh=select('sus', Chrom,
FitnV);          %选择
        SelCh=recombin('xovsp', SelCh,pc);                %重组
        SelCh=mut(SelCh,pm);                                  %变异
        V=bs2rv(SelCh, FieldD);
        newObjV=ObjFun(X,cn,V,options);   %计算子代目标函数值
        newChrom=SelCh;

        %是否替换旧个体
        for i=1:sizepop
            if ObjV(i)>newObjV(i)
                ObjV(i)=newObjV(i);
                Chrom(i,:)=newChrom(i,:);
            else
                p=rand;
                if p<=exp((newObjV(i)-ObjV(i))/T)
                    ObjV(i)=newObjV(i);
                    Chrom(i,:)=newChrom(i,:);
                end
            end
        end
        gen=gen+1;                                            %代计数器增加
        [trace(end,gen),index]=min(ObjV);                 %遗传算法性能跟踪
        trace(1:NVAR,gen)=V(index,:);
        fprintf(1,'%d ',gen);
    end
    T=T*q;
    fprintf(1,'\n 温度:%1.3f\n',T);
end
[newObjV,center,U]=ObjFun(X,cn,[trace(1:NVAR,end)]',options);   %计算最佳初始聚类中心的目标函
数值
% 查看聚类结果
```

```
Jb=newObjV
U=U{1};
center=center{1};
figure
plot(X(:,1),X(:,2),'o')
hold on
maxU = max(U)

%计算分类结果
for k=1:cn
    index(i)=find(U(i,:)  == maxU);

end
hold off
end
```

```
/GH_3_1.lg4/
model:
sets:
variable/1..835/:a;    !定价;
num/1..835/:b; !任务数是否完成;
hy/1..1877/; !会员个数;
links(num,hy):qw,n; !n 为完成度;
S_Con_Num/1..2/dplus,dminus;
endsets

data:
!期望价格;
qw=@file(data_3.txt);
enddata
max=dminus(1);
@sum(num(i):b(i))+dplus(1)-dminus(1)=0;
@sum(variable(j):a(j))<=180;    !总价不超过原先总价;
@for(links(i,j):n(i,j)=(@if((a(i)/qw(i,j))#ge#1,1,0)));
@for(num(i):b(i)=(@if(@max(links(i,j):n(i,j))#eq#1,1,0)));
End
```

```
/GH_3_2.lg4/
model:

sets:
variable/1..835/:a;    !定价;
num/1..835/:b; !任务数是否完成;
hy/1..1877/; !会员个数;
```

```
links(num,hy):qw,n; !n 为完成度;
SConNum/1..2/dplus,dminus;
endsets

data:
!期望价格;
qw=@file(data_3.txt);
enddata
```

**附录四　问题四代码**

```
/ PJ_4.m/
%% 清空环境变量
clear all
%% 导入数据
attributes=load('C:\Users\Desktop\shumo\13\testData.txt');
ycj=load('C:\Users\Desktop\shumo\13\testData.txt');
% 随机产生训练集和测试集
%n = randperm(size(attributes,2));
n=size(attributes,1);
time=attributes(3,:)';
% 训练集——num_train 个样本
num_train=20;
num_x=4;%因变量个数
p_train = attributes(1:num_x,[1:num_train])';
t_train = attributes(num_x+1:end,[1:num_train])';
% 测试集——总数-num_train 个样本
p_test = attributes(1:num_x,[num_train+1:end])';
t_test = attributes(num_x+1:end,[num_train+1:end])';
%% 数据归一化

% 训练集
[pn_train,inputps] = mapminmax(p_train');
pn_train = pn_train';
pn_test = mapminmax('apply',p_test',inputps);
pn_test = pn_test';
% 测试集
[tn_train,outputps] = mapminmax(t_train');
tn_train = tn_train';
tn_test = mapminmax('apply',t_test',outputps);
tn_test = tn_test';
%% SVM 模型创建/训练
% 寻找最佳 c 参数/g 参数
[c,g] = meshgrid(-10:0.5:10,-10:0.5:10);
[m,n] = size(c);
```

```
cg = zeros(m,n);
eps = 10^(-4);
v = 5;
bestc = 0;
bestg = 0;
error = Inf;
for i = 1:m
    for j = 1:n
        cmd = ['-v ',num2str(v),' -t 2',' -c ',num2str(2^c(i,j)),' -g ',num2str(2^g(i,j) ),' -s 3 -p 0.1'];
        cg(i,j) = svmtrain(tn_train,pn_train,cmd);
        if cg(i,j) < error
            error = cg(i,j);
            bestc = 2^c(i,j);
            bestg = 2^g(i,j);
        end
        if abs(cg(i,j) - error) <= eps && bestc > 2^c(i,j)
            error = cg(i,j);
            bestc = 2^c(i,j);
            bestg = 2^g(i,j);
        end
    end
end
%  创建/训练 SVM
cmd = [' -t 2',' -c ',num2str(bestc),' -g ',num2str(bestg),' -s 3 -p 0.01'];
model = svmtrain(tn_train,pn_train,cmd);
w=model.SVs'*model.sv_coef   %系数
b=-model.rho                 %常数

%% SVM 仿真预测
[Predict_1,error_1] = svmpredict(tn_train,pn_train,model);
[Predict_2,error_2] = svmpredict(tn_test,pn_test,model);
%  反归一化
predict_1 = mapminmax('reverse',Predict_1,outputps);
predict_2 = mapminmax('reverse',Predict_2,outputps);
%  结果对比
result_1 = [t_train predict_1];
result_2 = [t_test predict_2];
%{
%误差计算
result=[result_1;result_2];
for i=1:size(result,1)
    x=abs(result(i,2)-result(i,1));
end
x=x/size(result,1)
```

```
%}
%% 绘图

figure(1)
plot(time,[t_train;t_test],'k-*',time,[predict_1;predict_2],'k:o');
grid on
legend('真实值','预测值')
xlabel('时间(s)')
ylabel('用户意愿度')

%% 会员参与度预测
yn=60;
num=0;     %迭代次数
length=size(ycj,1) ;
i=1;
while i<length
      x=[yn,ycj(i,1),ycj(i,2)];
      while num<100
            %重新训练向量机
            if num~=0
                  x0=[x;myd];%将预测数据加入历史数据集
                  attributes=[attributes x0];
                  p_train=attributes(1:num_x,:)';
                  t_train=attributes(end,:)';
                  [t_train0,outputps] = mapminmax(t_train');
                  % 寻找最佳 c 参数/g 参数
                  [c,g] = meshgrid(-10:0.5:10,-10:0.5:10);
                  [m,n] = size(c);
                  cg = zeros(m,n);
                  eps = 10^(-4);
                  v = 5;
                  bestc = 0;
                  bestg = 0;
                  error = Inf;
                  for i = 1:m
                        for j = 1:n
                              cmd = ['-v ',num2str(v),' -t 2',' -c ',num2str(2^c(i,j)),' -g ',num2str(2^g(i,j) ),' -s 3 -p
0.1'];
                              cg(i,j) = svmtrain(t_train,p_train,cmd);
                              if cg(i,j) < error
                                    error = cg(i,j);
                                    bestc = 2^c(i,j);
                                    bestg = 2^g(i,j);
                              end
```

```
                    if abs(cg(i,j) - error) <= eps && bestc > 2^c(i,j)
                        error = cg(i,j);
                        bestc = 2^c(i,j);
                        bestg = 2^g(i,j);
                    end
                end
            end
            % 创建/训练 SVM
            cmd = [' -t 2',' -c ',num2str(bestc),' -g ',num2str(bestg),' -s 3 -p 0.01'];
            model = svmtrain(t_train,p_train,cmd);
        end
        attribute=attributes([1:num_x],:)';
        attribute=[attribute;x'];
        [train,puts]=mapminmax(attribute');
        train=train';
        [Predict_3,error_3] = svmpredict(myd,train(end,:),model);
        myd = mapminmax('reverse',Predict_3,outputps);%反归一化
        num=num+1;
        yn=0.5/myd+0.5;
    end
    i=i+1;
end
```