# **3**系统模型与问题陈述

## 3.1网络模型

本文采用模型与通用MCS一致，包含三个实体组成部分：工人集合，平台方以及数据请求端。图1说明了该模型的具体组成成分：其中我们使用表示工人，表示数据请求者。对工人而言，可分为信任工人和非信任工人；对平台而言，可分为高效益平台，中效益平台，低效益平台；对数据请求者而言，可分为高薪数据请求者，中薪数据请求者和低薪数据请求者。数据请求者是数据的需求方，当其有需要获得的数据，会向平台发布任务信息，平台经过评估以后选择性接收委托任务，到此完成了平台端和数据需求端的相互选择。随后平台端发布任务，工人可以从平台提供的任务集中选取任意个数的任务，并向平台提交申请，由平台选择优胜者来执行任务，由此平台端与工人之间达成协议。最后，工人向平台报告感知数据，平台对这些数据进行处理或者直接发送给信息需求端。这样我们就完成了一轮任务，并定义为第轮任务。

Picture

图1：MCS的网络模型

在每一轮任务中，工人需要向平台提交任务集合，位置集合，承诺质量集合以及期望报价集合，因此我们使用来表示工人在第次感知任务的提交数据，其中是每次感知的任务上限，此处引入来表示每次感知的任务前缀，显然有。其中表示在第i次感知的任务集，与此相对应的有其他三个集合数据。此时，基于参照效应的考虑，工人群体每次任务的报价与其移动距离和数据质量存在一定内联关系，为实现平台效益和感知数据质量的综合最优解，我们可以引入新的元素因子来表示不同任务的竞争权重，并设计函数来映射工人提交数据质量与标准数据的关系，其中是一个双向不均浮动函数，与标准数值差距越小，则映射数值越大。同时为了统一衡量标准，我们选用来表示等级为的标准数据质量，并将其映射到一维数组中，构建标准数据表。

**定义 1（工人信任度、工人信任度集合**、**工人信任度改变量）**

信息需求端发送给平台的数据收集任务是持续的，为提高数据质量、降低收集成本，需对工人质量进行评估筛选。因此引入工人信任度和工人信任度集合，用来在每次任务中累计工人每次感知任务中的，为工人每次感知任务实际数据质量等级与承诺数据质量差值的函数映射值。以此衡量工人在以后的任务中是否可信，估计其实际数据的利用价值。

**定义 2（信任工人集合 、非信任工人集合）**

在工人集合中，通过对信任度设置阈值，将工人集合分为信任工人集合和非信任工人集合，其中和为两类工人的索引，信任工人将直接参与计算准确数据和标准数据表，以此评估更新工人信任度集合、计算工人报酬。

**定义 3（移动距离）**

工人在向平台提交任务集合时，包含任务集合，位置集合表示工人所处位置，平台依据工人任务集合和位置集合计算出工人移动距离，并用分别做为感知任务、任务和工人的索引。参与后续工人竞争权值的计算。

**定义 4（覆盖原则、重合原则）**

覆盖原则为平台为工人分配任务时，优先考虑任务覆盖广度，再根据竞争权值选取工人。在选取非信任工人时，为提高本次任务对工人的信任度更新，以便后续感知任务评估工人可信程度，引入重合原则，即尽量增加非信任工人任务与信任工人任务的重合度，从而增强信任度更新梯度和准确性。

**定义 5（任务分配二向图**、**分辨强度**、**成本因子）**

平台得到工人提交的任务信息集合后，根据工人可信度对工人进行评估，并计算出工人的竞争权重，开始任务分配。此时用任务分配二向图来模拟工人任务分配情况。首先根据覆盖原则并比较信任工人竞争权值分配信任工人形成。然后根据重合原则、覆盖原则和竞争权值选取非信任工人。此时设置成本因子，用于限制非信任工人重合任务规模和数量，保证覆盖广度和报酬成本合理性。最终形成任务分配二向图。

**表 本文所用**符号

|  |  |
| --- | --- |
| 参数 | 描述 |
|  | 平台 |
|  | 工人集合、工人集合、工人索引 |
|  | 数据请求者集合、数据请求者、数据请求者索引 |
|  | 任务集合、单次任务、感知任务索引 |
|  | 工人在第次感知任务中提交的数据集合 |
|  | 工人在感知任务中所提交任务集合的索引和数量上限 |
|  | 工人在第次感知任务中提交的第个任务及位置 |
|  | 工人在第次感知任务中提交的第个任务的承诺数据质量及报价 |
|  | 工人在第次感知任务中提交的第个任务的竞争权重 |
|  | 工人在第次感知任务中提交的第个任务的移动距离、分辨强度 |
|  | 工人在第次感知任务中提交的第个任务的成本因子 |
|  | 工人在第次感知任务中提交的第个任务的数据、第次感知任务中第个任务准确数据 |
|  | 工人信任度、工人信任度集合、工人在第次感知任务结束后信任度改变量 |
|  | 信任工人集合、信任工人索引 |
|  | 非信任工人集合、非信任工人索引 |
|  | 标准数据质量等级表、数据质量等级索引、等级为的标准数据质量 |
|  | 工人在第i次感知任务中索引为j的最终收益 |
|  | 工人提交数据可获得的额外报酬 |
|  | 工人提交数据相较于标准数据的参照因子 |
|  | 平台方规定第i次感知中对索引为k的任务承诺质量的最低水平 |
|  | 平台方选取第i次感知中索引为j的任务的优胜者集合 |
|  | 工人a在第i次感知任务中对于索引为j的任务的实际质量 |
|  | 工人a在第i次感知任务中对于索引为j的任务的移动成本 |
|  | 报酬、工人在第次感知任务中获得的报酬、工人在第次感知任务中任务获得的报酬 |
|  | 第次感知任务中信任工人任务分配二向图、工人任务分配二向图 |
|  | 工人a在第i次感知任务中对于索引为j的任务的成本需求 |

## 3.2问题表述

在我们的网络中，主要存在两种交互过程，（a）工人集合向平台提交感知任务申请，经由平台方衡量工人对于任务j的竞争权重，并向选择优胜者集合发布任务，随后工人经由移动设备在任务地点进行数据感知并向平台提交数据；（b）平台存储不同工人的提交数据，并基于工人承诺质量对工人集合进行报酬计算，其中报酬 =+，并针对工人i的任务索引发放工资。

在这样的过程中，为了解决理想模型中对工人报酬的理想衡量的问题，即通过假想理论值进行额外报酬的计算，我们引入了信任度的概念，用于修正同类论文中的理想模型，使得感知网络更方便的用于现实生活。从平台方出发，我们期待通过激励来实现感知数据质量的尽可能提升，但在实际运作中，我们往往无法衡量一个数据的质量，从而造成激励的部分无效化，甚至对感知质量的提升造成负反馈效果，因而，如何实现感知数据的宽覆盖范围和高质量与低成本的动态最优便成为本文要解决的问题。

本文的优化目标可分为两个方向，即平台方针对不同任务的工人选择，更新感知任务的数据属性处理以及设计更为合理的报酬解决方案。从传统MCS模型上看，工人数据质量与其获得报酬呈线性相关，但这显然无法较好的拟合实际情况，同时工人数据质量与平台方实际效益和数据请求者的需求均呈正相关，因此我们只需要用一个较好的因子来构建适当的复合函数即可实现。

首先从平台角度看，任务请求者给出一定收益，随后平台发布任务并从用户端用户端取得数据，此时请求端给出的收益可以视作，随后用户端领取任务会产生一定损耗，具体包括期望报价和额外报酬，设定用户端实际成本为，其中数据质量会同步影响和，我们期望通过较低的总成本尽可能提高数据质量，同时使用户获得适当薪资从而对平台产生依赖，符合现状偏好效应，从而在多轮任务执行中降低报价并提供更为可靠的数据质量，因此我们期望构建复合式响应函数：

（1）

从而实现平台方收益的动态变化最优，即多轮中的期望和最大，同时目前相关研究中并未将作为因子参与动态衡量计算，按照我们先前所述，这种情况显然缺乏应用场景，因此如何选取适当因子来解决此函数的构建，以及如何让用户与平台产生粘性，都是目前需要解决的问题。

另一方面，从工人的角度出发，期望高质量的感知数据能够与激励相匹配，即工人初始提供质量保证和实际质量相互向上式匹配时，工人可以获得更高的报酬，即匹配的更高，同时在进行数据感知时，工人方会不可避免地产生距离成本或移动成本，而成本的存在又会迫使工人衡量感知任务的选择必要性以及工人在进行感知任务的有效程度，因而在设计工人对于任务j的参照因子对于额外报酬的映射时，成本效应同样显得十分重要。同时，在上述叙述中，我们可以看到用户的感知数据质量是需要一个衡量标准，而在本文采用的实际MCS模型中，这个衡量标准并无法事前得知，因而衡量工人a在不同任务的数据质量难以做到，在此基础上的具体计算显然更具挑战性。

而对于工人而言，显然能提供更高报酬的平台会更具吸引力，同时工人普遍更倾向于能对自己实际感知成本做出反馈的平台，因而我们可以构建工人报酬计算函数：

（2）

其中的计算在实际应用中需要考虑信任度和承诺质量，因而工人的实际报酬也是一个动态变化的过程，即工人提交感知数据与承诺质量的映射差值越小，则其信任度增加，大于一定值则需降低，同时额外报酬的计算整体效果与信任度变化一致，但在实际应用中我们需要不同的衡量标准，平台对信任度偏低的用户持否定意见，故信任度的变换需要存在一个下界限，而额外报酬的计算显然对于不同的承诺质量应作出不同的定价，同时额外报酬也不应让用户产生负收益。整体而言，在这个相互选择的过程中，工人初始的信任度是相当主要的影响因素，同时通过影响数据质量间接影响平台方和工人方的效益，因此，本文通过设计一种基于信任度和参照效应的选择机制，实现平台方综合效益，感知数据质量和工人实际报酬的最优解。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | |  |  |
|  |  | |  |  |
|  | |  | |  |
|  | |  | |  |
|  | |  | |  |