First Learn then Earn: Optimizing Mobile Crowdsensing Campaigns through Data-driven User Profiling

Merkourios Karaliopoulos
mkaralio@aueb.gr

Iordanis Koutsopoulos
jordan@aueb.gr

Michalis Titsias
mtitsias@aueb.gr

Department of Informatics
Athens University of Economics and Business
104 34 Athens, Greece

ABSTRACT

We study the optimal design of mobile crowdsensing campaigns in terms of the aggregate quality of contributions attracted for a set of tasks. The interaction of the campaign with users is realized through a mobile app interface that recommends tasks to users and offers them incentives. The main contribution is a novel perspective on the payment distribution problem faced by the crowdsensing campaign organizer in light of originally unknown individual user preferences. Contrary to common practice, we acknowledge that users exhibit high diversity in decision making because they assess differently *attributes* related to a task such as their proximity to the place of interest (PoI), the payment made for contributing data, or the task context/theme.

We draw on logistic-regression techniques from machine learning to *learn* users' individual preferences from past data rather than hypothesizing about them. We then formulate non-linear (sigmoid) optimization problems to determine the tasks and incentives (payments) that should be optimally offered to each user. Our mechanism is validated against synthetic but also real data about the way users choose tasks, collected through an online questionnaire. It achieves very good approximations of the optimal solutions and substantially outperforms alternative preference-agnostic policies that do not exercise behavioral user profiling to target the provision of incentives.

CCS Concepts

•**Information systems** → *Incentive schemes;* •**Human-centered computing** → *Ubiquitous and mobile computing systems and tools;* •**Computing methodologies** → *Supervised learning by classification;*

Keywords

mobile crowdsensing; user profiling; logistic regression

1. INTRODUCTION

Mobile crowdsensing has emerged as a new sensing paradigm over the recent years. The proliferation of smartphones with a multitude of embedded sensors together with the rich offer of mobile applications transform the mobile users from passive consumers of information to producers and active contributors of various kinds of data [9] [4].

Mobile devices and the humans behind them participate in crowdsensing campaigns originated by corresponding service providers (SPs). Such campaigns leverage the "wisdom of the crowd" by collecting individual data from mobile devices and turning it to collective knowledge that would otherwise be very difficult or impossible to obtain. Example applications span the areas of environmental monitoring and awareness (*e.g.*, [19] [15]) transportation (*e.g.*, [12]), and even lifestyle and healthcare management (*e.g.*, [14]).

Central to the design and operation of an online crowdsensing platform is the recruitment of users. Their contributions to tasks assigned to them are subject to various types of costs they or their devices may experience including the mobile device energy consumption, mobile operator bill, location or other data privacy cost, and the devotion of precious time and attention to perform the task. Hence, the scientific literature is rich in studies that design incentive mechanisms for motivating users to participate in the campaign and optimizing the collection of data in terms of their acquisition cost or quality.

However, most (if not all) of them rely on a fundamental assumption: the devices are instances of intelligent software agents that take fully rational decisions on behalf of users, often by solving complex optimization problems. In these problems, the user preferences are abstracted to analytical utility functions and the aim becomes to design a mechanism that optimizes a certain objective such as minimizing the expenses or maximizing the quality of collected information.

The problem with this assumption is that the decisions regarding the participation in crowdsensing tasks are more often than not taken by the humans behind the mobile devices, in ways that deviate considerably from the norms of perfect rationality. Such decisions depend on the different and highly personal ways individual users prioritize or weigh various attributes of the tasks. Attempting to a priori model the users' decision-making process through (homogeneous) analytical utility functions, *e.g.*, concave function for relating their satisfaction to offered payments, is an oversimplification that renders the results of the model questionable.

In this work, we radically depart from the current common assumption that individual user preferences may be modeled by utility functions in analytic form. Instead, *we*

MobiHoc'16, July 04 - 08, 2016, Paderborn, Germany

ISBN 978-1-4503-4184-4/16/07...$15.00

DOI: http://dx.doi.org/10.1145/2942358.2942369

use machine learning techniques to learn these preferences through records of past crowdsensing task offers and user choices. The derived behavioral user decision-making profiles capture the varying importance that each user places on different task attributes. We then use these personalized user preference models to *predict* the probability that a user will carry out specific tasks. A key observation is the following. Each task comprises attributes that are beyond the control of the campaign designer and attributes that are under its control, such as the incentives to be designed. These attributes are present in the user decision-making model. The goal, then, is to compute appropriate values for controllable attributes, *i.e.*, the payments to be offered to users, so that the probability that users carry out tasks is maximized and the purpose of the campaign is best fulfilled.

1.1 Motivating example

Consider a mobile crowdsensing campaign offering some payment to users who visit specific places and take a photo with their mobile devices. The campaign interacts with users through a mobile app that makes suggestions of the following form to users: "Consider visiting place X at distance d away for a payment \$$p$". Assume that the objective of the campaign designer is to maximize the total expected number of photos from all places subject to a finite available budget for paying users. Let us also assume that users decide whether to visit the suggested place to take a photo based on two parameters, their physical proximity to the place, and the offered payment. The first challenge for the campaign organizer would be to profile users, *i.e.*, use historical data to build a model that describes how probable it is that a user accepts a task request depending on the pair (d, p) of the offered task. Then, in a second step, it could draw on these derived user profiles to issue targeted place recommendations to users and tune the offered payments so as to maximize the probability that users indeed visit the recommended places and contribute photos. Our paper demonstrates how these two component operations of the crowdsensing campaign can be formulated and executed.

1.2 Our contribution

Input to our problem is a set of tasks that need to be accomplished and our objective is to maximize the aggregate quality at which this happens. In this paper, the achieved quality per task is an additive function of the qualities of the contributions made by the users carrying out the task. The expected quality contributed by a user to a task is the product of a quality index that characterizes the appropriateness of the specific user and task, and the probability that a user will perform the task. Each task comprises a set of attributes, and the profile of a user is a vector of attribute weights that are obtained by training the model from past data. The control actions of the campaign design are the selection of the task(s) to be presented through the mobile app interface to users, as well as the determination of the incentive payment to them.

The main contributions of our work are as follows.

- We use a logistic-regression model and machine learning techniques to build personalized behavioral decision-making profiles for users based on their responses to past offers with different values of attribute vectors (Section 3). The model allows to capture the uncertainty on whether a user will ultimately perform a task or not.
- We formulate the problem of maximizing the sum of expected qualities of tasks through task and incentive allocation to users as a sigmoid optimization one (Section 3).
- We proceed to specific instances of the problem, corresponding to different policies the matching of users with tasks, each time identifying the type of problem and proposing a solution for it (Sections 4.1 and 4.2.2). The solutions may be exact (special case of assigning at most one user per task, reducing to a max-weight matching problem); approximately optimal(for one task and multiple users); or call for some combination of heuristic rules with approximately optimal solutions (when there are multiple tasks and more than one users may be assigned to a task, yielding an instance of the generalized assignment problem).
- We study the even more general scenario with two recommended tasks to a user, together with the option to perform the one or the other, and we extend the attribute models so as to capture the task selection process involved in user decision-making (Section 4.3).
- We validate our approach through experimenting with real data collected through an online questionnaire that seeks to infer the user preferences in a virtual mobile crowdsensing campaign. Our methods achieve very good approximations of the optimal solutions and substantially outperform different benchmark policies that lack the perspectives of behavioral user profiling and incentive-targeting (Section 4).

In section 2, we present an overview of state-of-the-art work. and in section 5 we conclude the paper iterating on directions for future work.

2. BACKGROUND-RELATED WORK

There exists a large body of research work on incentives for mobile crowdsensing. In [20] the objective is to select a subset of data contributors for maximizing total utility minus sum of payments. The submodularity of the objective is exploited to devise a truthful greedy algorithm and show its effectiveness. In [7], the optimal auction framework is used to design a data market that takes into account the strategic behavior of data contributors, who may misreport the cost of data contribution. An incentive-compatible mechanism is designed to determine participation levels and payments to users with the aim to minimize data acquisition cost and ensure a certain quality of aggregate information.

The provision of incentives has been looked upon across a longer-time scale as well. In [5], the authors use the framework of Lyapunov optimization to design an online algorithm for sensor selection at each time slot for maximizing social welfare, which is defined as total sensing value minus sensing cost. The long-term participation of users is achieved by ensuring that the probability of selecting each of them is no smaller than a threshold. On the other hand, in [8], user payments are determined via auctions and the long-term participation of end users is motivated by providing them with virtual credit just for participating in a round. The participants may then use this credit for reducing their offer in subsequent rounds, so that the set of winning users changes over different rounds. This technique proves to be effective in discouraging frequent winners from constantly increasing their offers, and it is shown to reduce compensation cost.

Another set of works has been concerned mainly with the quality of contributed data [16], [13], [6]. In [16], a mathematical framework is devised that involves self-interested data contributors, service consumers, and a service provider. The quality and timeliness of contributed data is characterized through a novel metric that helps shape the market in terms of compensation to contributors and service consumption rate of service consumers. In [13], an expectation maximization algorithm continuously estimates the quality of gathered data, while the anticipated quality of contributed information for each user is estimated based on mutual information. Both are taken into consideration in a payment scheme that pays participants in accordance with their effective contribution. In [6], the aggregate quality of all tasks minus the set of costs that users undergo is maximized through a truthful incentive mechanism based on reverse combinatorial auctions. Potential data contributors place bids on subsets of tasks they may contribute data to. The mechanism takes into account the envisioned qualities and outputs the selected winners and their compensations for executing the declared subsets of tasks.

Finally, a different perspective on incentives for mobile crowdsourcing is exemplified in [18]. Therein, a gamification mechanism seeks to boost the user engagement via user ranking and status level schemes based on reward points.

3. MODEL AND PROBLEM STATEMENT

3.1 The Crowdsensing Application Model

Denote by $\mathcal{U}$ the set of potential crowdsensing data contributors participating in the campaign. This set comprises mobile users who own smart devices, have registered with a crowdsensing platform, and run the related mobile application on their device. The crowdsensing campaign designer interacts with those users through the mobile app and suggests specific tasks to each of them as they roam in the city, on their way to work, back home or during their leisure time. These suggestions change over time depending on the users' current physical locations and possibly on other context information collected by the mobile application.

The campaign has a certain time horizon, typically several days up to several months. During the campaign, the mobile app interacts with users regularly and asks them to perform certain tasks, say once in fifteen minutes or once every hour or so. Whether the task amounts to data collection and contribution, photo-shooting or another task, it has to be performed within a certain time interval since it is suggested to a user.

We restrict our attention to the task selection and allocation problem during such regular time intervals, over which a set $\mathcal{L}$ of crowdsensing tasks have to be performed. The mobile application issues task recommendations to a user, *e.g.*, *"You could get a reward p if you carry out task l at distance d from your current location"*. If the user responds to the recommendation and contributes data within the relevant time interval for which the information is useful to the campaign, (s)he gets a reward p, which could generally be monetary or indirect (*e.g.*, credit points that can be exchanged for purchases or discounts).

Each task $l \in \mathcal{L}$ may be viewed as a set of *attributes* $l = \{l_1, l_2, ..., l_n\}$ that may be numerical or categorical, including the reward that a task may offer when performed, the physical location where the task needs to be carried out, the average time or effort it will take to perform the task, the battery/computational requirements posed by the task, and the context (*e.g.*, commercial or non-profit) of the underlying service that is facilitated with the requested task.

Users' aptitude for specific tasks varies. For instance, an amateur photographer is more appropriate for a task involving photo-shooting; but so maybe the case with someone who owns a smart device with excellent camera resolution. Formally, to each user-task pair u, l with $u \in \mathcal{U}$ and $l \in \mathcal{L}$ we assign a quality index $q_{ul} \in [0, 1]$ that quantifies how good a contribution user u can make to task l. This index may also depend on various persistent user attributes *e.g.*, their interests and pro-social attitude.

There exist limitations on the compensation expenses for the campaign. Here, we assume that each task $l \in \mathcal{L}$ comes up with its own independent budget B_l, which sets an upper bound to the total payments that can be made to potential task contributors. An alternative, which we do not consider here, would be that the budget for all tasks is globally managed, including the possibility to move budget across tasks as far as some performance objective is reached. Payments are taken to be continuous variables.

Contrary to the dominant assumption in the literature that tasks are *assigned* to users and users *deterministically* carry them out, in our work tasks are *offered* to users and they *choose* whether to carry them out or not. Namely, users undertake a task, or select one task over another *probabilistically*, depending on how well different tasks compare with their own individual preferences. We explain how we let this happen in the section that follows.

3.2 User Profiling and Incentive Allocation

We approach the user response to the requested tasks as an instance of the two-class probabilistic classification problem. The two classes correspond to the possible outcomes of the user decision process, which vary depending on the actual decision setting. For instance, when a take-or-leave-it request is made to the user for a single task, class 1 (C_1) corresponds to responding positively to it and carrying out the task, while class 0 (C_0) corresponds to ignoring it.

Likewise, when two tasks are presented to a user, we assume that (s)he may choose one of the two to contribute to, C_1 may correspond to the choice of the one and C_0 to the selection of the other. Clearly, this model can be extended to one comprising four classes to account also for the possibilities that a user may perform both suggested tasks, or none of them. However, in this work we adhere to a two-class model as a basic reference in order to demonstrate our approach. The model essentially implies that the mobile application suggests one or two tasks to the end users so that their cognitive load remains manageable in light of the usually constrained time frame of their decisions. Nevertheless application instances that propose to the end users more than two tasks may arise; we have ourselves experimented with such an instance in [10].

The way each user u weighs the two choices at hand to reach a decision is modeled by a logistic regression model, a popular machine-learning model for probabilistic classification. Logistic regression makes the hypothesis that, given

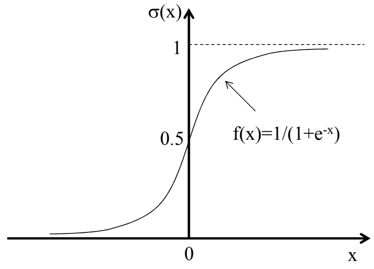


Figure 1: Sigmoid function $\sigma(x)$. It takes values in [0,1] as x varies in $[-\infty, +\infty]$

a vector of values $\mathbf{x}$ for the different attributes (*features* in the machine learning terminology), the first alternative is chosen with probability

$$P_u(C_1|\mathbf{x}) = \frac{1}{1+e^{-\mathbf{w}_u\cdot\mathbf{x}}} = \sigma(\mathbf{w}_u \cdot \mathbf{x}), \qquad (1)$$

where $\sigma(a) = (1+e^{-a})^{-1}$ denotes the logistic sigmoid function (ref. Fig. 1), while $\mathbf{w}_u \cdot \mathbf{x}$ denotes vector dot product, and $\mathbf{w}_u$ is the vector of the feature weights for a user u. These weights are learned from historical data and capture the significance that user u places on different features and their values in reaching a decision. Similarly, the alternative choice is selected with probability $P_u(C_0|\mathbf{x}) = 1 - P_u(C_1|\mathbf{x})$.

Logistic regression [2, 11] is the most celebrated instance of generalized linear models, whereby the underlying decision boundary that separates the two classes is linear in the feature vector $\mathbf{x}$.

$$P_u(C_1|\mathbf{x}) = P_u(C_2|\mathbf{x}) \;\Rightarrow \mathbf{w}_u \cdot \mathbf{x} = 0$$

Furthermore, important properties of logistic regression are: (i) the feature weights can be learned easily, as the single global solution to a convex optimization problem (see section 3.2.1), *i.e.*, without complications related to local optima, and (ii) the model is probabilistic, hence it comes with well-calibrated estimates of uncertainty in the classification decision [2, 11]. Note that there are other classification algorithms with convex objectives such as the well-known Support Vector Machines. However, they are not suitable in our case since they do not provide probabilistic predictions.

The feature vector $\mathbf{x}$ can be broadly partitioned into two subsets of features. First, it involves features that are inherent to the task and user and may not be controlled by the service provider (SP). For example, the physical distance of the task from the current user location, the social/community *vs.* commercial orientation of the task, and its context/theme. We denote this feature subset as $\mathbf{x}^s$. A second subset, referred to as $\mathbf{x}^c$, consists of features that are dynamic and could be manipulated by the SP in order to shape the task acceptance probability $P(C_1|\mathbf{x})$. One such feature is the payment that the SP may provide to the users as an incentive to carry out a task.

To figure out how to tune these payments, the SP first needs to learn how each user weighs the different features upon making a choice, that is, it needs to learn the users' weight vectors (or *profiles*) $\{\mathbf{w}_u\}, u \in \mathcal{U}$ in the logistic regression models. This is realized through a supervised learning process, whereby data concerning past users' choices in similar settings are used to train the individual user models.

3.2.1 Training the logistic regression models

The training dataset $\mathcal{D}_{tr}(u)$ for each user u is made up of the feature values of m past task suggestions and the user responses to each,

$$\mathcal{D}_{tr}(u) = (\mathbf{x}_{u1}, y_{u1}), (\mathbf{x}_{u2}, y_{u2}), ..., (\mathbf{x}_{um}, y_{um}), \qquad (2)$$

where $\mathbf{x}_{uj}$ are the feature values for the j-th task suggested to u and y_{uj} are the labels, 1 or 0, depending on u's choice.

The likelihood function for the estimation of $\mathbf{w}_u$ is written as (see *e.g.*, [2])

$$P(\mathbf{y}_u|\mathbf{w}_u) = \prod_{k=1}^{m} P_u(C_1|\mathbf{x}_{uk})^{y_{uk}}(1 - P_u(C_1|\mathbf{x}_{uk}))^{1-y_{uk}},$$

with $\mathbf{y}_u = (y_{u1}, \ldots, y_{um})$. One way to estimate the weights is to follow the Maximum-Likelihood principle and maximize the logarithm of the above quantity (the so called log-likelihood) with respect to $\mathbf{w}_u$. However, this could lead to overfitting and over-confident predictions. Therefore, in practice, the preferred estimation procedure is to add a regularization penalty on the weights, of the form $\frac{\lambda}{2}||\mathbf{w}_u||^2$ (where $||\cdot||$ denotes the Euclidean norm), which places preference on smaller weight values and formally arises by introducing a Gaussian prior distribution on $\mathbf{w}_u$ [2, 11]. Hence, the aim of the per user training process is to find the weights $\mathbf{w}_u$ that minimize $E(\mathbf{w}_u)$

$$\begin{aligned} E(\mathbf{w}_u) &= -\ln P(\mathbf{y}|\mathbf{w}_u) + \frac{\lambda}{2}||\mathbf{w}_u||^2 \\ &= -\sum_{k=1}^{m}\big[y_{uk}\cdot\ln(P_u(C_1|\mathbf{x}_{uk})) \\ &\quad + (1-y_{uk})\cdot\ln(1-P_u(C_1|\mathbf{x}_{uk}))\big] + \frac{\lambda}{2}||\mathbf{w}_u||^2, \quad (3) \end{aligned}$$

with $P(C_1|\mathbf{x}_{uk}) = \sigma(\mathbf{w}_u \cdot \mathbf{x}_{uk})$. The gradient of this function with respect to $\mathbf{w}_u$ can be shown to be

$$\nabla E(\mathbf{w}_u) = \sum_{k=1}^{m}(P_u(C_1|\mathbf{x}_{uk}) - y_{uk})\mathbf{x}_{uk} + \lambda\mathbf{w}_u \qquad (4)$$

and can be used by a gradient-descent algorithm to iteratively converge to the optimum vector $\mathbf{w}_u$ through

$$\mathbf{w}_u^{(\tau+1)} = \mathbf{w}_u^{(\tau)} - \eta\nabla E(\mathbf{w}_u)\,, \qquad (5)$$

where η is the learning rate parameter that determines the aggressiveness with which the algorithm will move towards the minimum. Since the error function above is a convex function [2, 11], the minimum is a global one, *i.e.*, there is no risk of stumbling over a local minimum.

In the numerical investigations presented in Section 4, the regularization parameter λ is determined by cross-validation [2, 11], which selects λ so that the classification performance in held-out (test) data is maximized.

3.2.2 Allocating incentives as a sigmoid optimization problem

The model training process in section 3.2.1 is carried out separately for each user u yielding its feature weight vector $\mathbf{w}_u$. Then the probability $P_u(C_1|\mathbf{x}_{ul})$ that user u will accept a request to contribute to task l, is a sigmoid function of individual feature values $\mathbf{x}_{ul}$, as shown in Eq. (1). If $\mathbf{w}_u^s$ and $\mathbf{w}_u^c$ are the weight sub-vectors corresponding to the feature sub-vectors $\mathbf{x}^s$ and $\mathbf{x}^c$, respectively, this can be written as

$$P_u(C_1|\mathbf{x}_{ul}) \;\; = \;\; \sigma(\mathbf{w}_u^c \cdot \mathbf{x}_{ul}^c + \mathbf{w}_u^s \cdot \mathbf{x}_{ul}^s). \qquad (6)$$

The SP is faced with a set of tasks $\mathcal{L}$ and a set $\mathcal{U}$ of possible task contributors. Depending on his/her location, each user u is a candidate contributor only for a subset of tasks $\mathcal{L}_u \subseteq \mathcal{L}$, *e.g.*, those lying in u's proximity, and the quality of u's contribution to each of them is determined by the quality indices q_{ul}, $l \in \mathcal{L}_u$. Then, in the general case, the SP seeks to optimally control the subset of task features $\mathbf{x}^c$ to maximize the expected aggregate quality of user contributions to tasks. We take the aggregate quality to be an additive function of individual contributed qualities, but this assumption could be relaxed. The problem faced by the SP is

$$\begin{aligned} &\textit{maximize} && \sum_{u\in\mathcal{U}}\sum_{l\in\mathcal{L}_u} q_{ul}\sigma(\mathbf{w}_u^s \cdot \mathbf{x}_{ul}^s + \mathbf{w}_u^c \cdot \mathbf{x}_{ul}^c) \\ &\textit{subject to:} && \mathbf{x}_{ul}^c \in \mathcal{C}, \forall l \in \mathcal{L}, \forall u \in \mathcal{U}, \end{aligned} \tag{7}$$

where $\mathcal{C}$ is a nonempty bounded closed convex set which determines the constrained feasible solution space for the subset of all controlled features $\{\mathbf{x}_{ul}^c\}$.

More often than not, each vector $\mathbf{x}_{ul}^c$ may be reduced to a scalar corresponding to the payment made to a user as an incentive to contribute to some task, *i.e.*, $\mathbf{x}_{ul}^c \equiv p_{ul}$ and $\mathbf{w}_u^c \equiv w_{u,p}$. If we incorporate in the formulation the total budget constraint per task, (7) reduces to

$$\begin{aligned} &\textit{maximize} && \sum_{u\in\mathcal{U}}\sum_{l\in\mathcal{L}_u} q_{ul}\sigma(\mathbf{w}_u^s \cdot \mathbf{x}_{ul}^s + w_{u,p}p_{ul}) \\ &\textit{subject to:} && \sum_{u:l\in\mathcal{L}_u} p_{ul} \le B_l \quad \forall l \in \mathcal{L} \\ & && 0 \le p_{ul} \le p_{max} \quad u \in \mathcal{U}, l \in \mathcal{L}_u, \end{aligned} \tag{8}$$

where p_{max} is the maximum allowed payment available for a single user to carry out a task.

In what follows, we describe how this framework is tailored to different crowdsensing service settings, starting from simpler ones and moving towards more composite scenarios.

4. SPECIFIC INSTANCES OF TASK AND INCENTIVE ALLOCATION

The payment allocation framework outlined above and the optimization problem formulation in (8) are generic in at least *four* main respects. Firstly, they accept different definitions of *classes.* The two classes may correspond to accepting or rejecting an offer for a single task, respectively; or they may note which task is selected out of two task offers made (*e.g.*, the one that lies closer to the user *vs.* one that is further away but pays more). Secondly, the *feature set* may depend on one or both tasks (ref. Section 4.3), yielding different types of training datasets. Thirdly, they do not explicate the *subset of static features* $\{\mathbf{x}_{ul}^s\}$. Finally, they do not specify the subset of tasks $\mathcal{L}_u$ that becomes relevant for each user. This subset may typically result from constraints related to both the physical distance of task locations and the task request policies on the side of the SP.

In what follows, we study specific instances of this framework, where the subset of static features includes only the physical distance(s) of task(s), $\{\mathbf{x}_{ul}^s\} \equiv d_{ul}$. This way, users essentially view tasks as pairs of attributes (l_1, l_2), i.e. their physical locations and the payments they offer. We consider different cases of the problem that lead to different modes of determining the classes, feature vectors and subsets $\mathcal{L}_u$ across different users. These cases give rise to distinct instances of what is originally a *joint user-to-task assignment and payment allocation problem* in Eq. (8). In all numerical investigations that follow, the locations of the crowdsensing task(s) and users are randomly dispersed across a rectangular area of 1x1 km^2, and the user quality indices $\{q_{ul}\}$ are randomly drawn from [0,1].

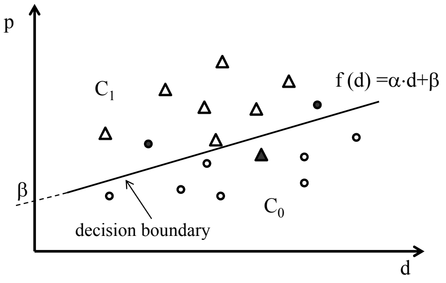


Figure 2: Generation of the synthetic training dataset. The generated (d, p) samples are first classified according to a strict linear rule. Then, points close to the decision boundary are probabilistically relabeled (shaded points denote relabeled points).

For the solution of the resulting sigmoid optimization problems, we draw on the method proposed in [17] by Udell and Boyd for the general problem of maximizing sums of sigmoids. Therein, the authors propose an approximation algorithm that uses a branch-and-bound method to find globally optimal approximate solutions. Notably, they show that their algorithm can find approximate solutions very quickly on problems with small number of variables or constraints. Finally, in the scenarios we consider, we compare the payments made by our framework against alternative schemes that do not invest effort on profiling individual users.

4.1 Single task

To begin with, we consider that the SP seeks to select the users and tune the offers it makes to users for a single crowdsensing task. Thus, $|\mathcal{L}_u| = |\mathcal{L}| = 1, \forall u \in \mathcal{U}$, the decision feature set is $\mathbf{x}_{ul} = (d_{ul}, p_{ul})$, class $C_1(C_0)$ denotes acceptance (resp. rejection) of a task request and the respective payment offer, and (8) reduces to a continuous-valued payment allocation problem.

4.1.1 Evaluation methodology

For this case, the training dataset, $\mathcal{D}_{tr}$, is synthetic. For each user, we assume that the minimum payment p_{ul}^{min} that will convince her to accept the task offer varies linearly with the task physical distance, *i.e.*, $p_{ul}^{min} = \alpha_u \cdot d_{ul} + \beta_u$. The terms $\alpha_u > 0$, and β_u are randomly chosen to capture how each user resolves the tradeoff distance *vs.* payment presented by each task. We generate random pairs of (d_{ul}, p_{ul}) values and label them as C_1 or C_0 depending on how the points (d_{ul}, p_{ul}) position with respect to the p_{ul}^{min} line in the 2D space (see Figure 2). To accommodate deviations from a perfectly linear decision boundary, we probabilistically shift the label of points that lie close to the boundary.

We then use these synthetic datasets to train the feature weight vector $w_u = (w_{u,d}, w_{u,p})^T$ for each user. As part of this process, the original features (d_{ul}, p_{ul}) are normalized to yield $(d_{ul,n}, p_{ul,n})$, where

$$\begin{aligned}\tilde{d}_{ul} &= (d_{ul} - \mu_d)/\sigma_d \\ \tilde{p}_{ul} &= (p_{ul} - \mu_p)/\sigma_p \qquad (9)\end{aligned}$$

$\mu_d(\mu_p)$ and $\sigma_d(\sigma_p)$ being the mean and standard deviation of the distance (resp. payment) values across each user training dataset. The task of the SP is then to optimize the payments made to each user for the single task. We drop the task index and obtain

$$\begin{aligned}&\mathit{maximize} \quad \sum_{u\in\mathcal{U}} q_u \sigma(w_{u,d} \cdot \tilde{d}_u + w_{u,p}\tilde{p}_u) \\ &\mathit{subject\ to:} \quad \sum_{u\in\mathcal{U}} p_u \leq B \\ &\qquad 0 \leq p_u \leq p_{max} \quad u \in \mathcal{U}\,. \qquad (10)\end{aligned}$$

The solution of the optimization problem implies that the task will be recommended to those users for whom $p_u > 0$.

4.1.2 Numerical results

In all cases, the solution of (10) with the branch-and-bound method yields values that are exactly (the lower and upper bounds coincide) or very close to (the gap of the two bounds is minimal) optimal. Indicatively, in 30 out of 30 simulation runs with B=30 and 50 users and in 48 out of 50 runs with B=30 and 100 users, the difference between the two bounds was smaller than 0.001, in most cases the two bounds being practically identical.

We compare our solution against three alternative schemes for payment allocation. Common to all of them is that they do not invest effort to learn how the features affect users' decisions (individual preferences), although they are aware of user locations and their skills for individual tasks. Two of them split the task budget equally among the top-k candidate contributors, differentiating in how they rank users. The first one makes offers to the k users with the highest quality indices q_u (*pay equally the k most skilled-pekms*), whereas the second one makes offers to the k users lying closest to the task location (*pay equally the k closest-pekc*). In either case, each of these k contributors is offered an amount $p_u = p = B/k$, whereas the remaining ones are not offered any payment. Finally, the third scheme (*pay proportionally the k most skilled-ppkms*) again distributes the budget to the k most skilled users, but proportionally to their quality indices, resulting in payments:

$$p_u = \begin{cases} q_u / \sum_{j=1}^{k} q_u & \text{if } \ rank(u;q_u) \leq k \\ 0 & \text{otherwise} \end{cases}$$

where $rank(u;q_u)$ denotes the rank of user u across $\mathcal{U}$ according to her quality index, q_u.

The typical way that k, the number of offers made per task, affects the performance of the three individual-preference-agnostic payment allocation schemes is shown in Fig. 3. At small k values, splitting the budget across more users increases the expected number of contributors and results in higher aggregate expected contribution quality. The payments are adequately high to increase substantially the probabilities of accepting the task offer, driving the sigmoid curve towards its rightmost values (see Fig. 1). At this range of values, sharing the budget between more users is beneficial for the aggregate welfare. On the contrary, at high k values, splitting the budget to many users results in low individual payments that cannot substantially alter the task acceptance probabilities. At this range of values, the excess budget fragmentation hurts the aggregate welfare. At intermediate k values, the aggregate score exhibits short-term variations (ups and downs) depending on whether the expected contribution quality from the k^{th} user outweighs the drop in the expected qualities of the $(k-1)$ existing users due to reductions of their own payments. For the sake of comparison, in the same example, our scheme distributes the budget to 22 different users, determining payments in the range of \$0.28 to \$3.4 and achieving an overall score of 15.13 (with coinciding upper and lower bounds).

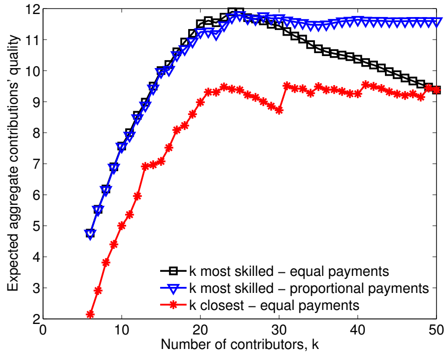


Figure 3: Aggregate utilities achieved under the three individual preference-agnostic payment allocation rules: $|\mathcal{L}|$=1, B=40, $|\mathcal{U}|$=50.

In Fig. 4, we show how the percentage gain of the approximately optimal over the other three payment allocation schemes varies over 50 simulation runs (problem instances). For a single simulation run, this gain is defined as:

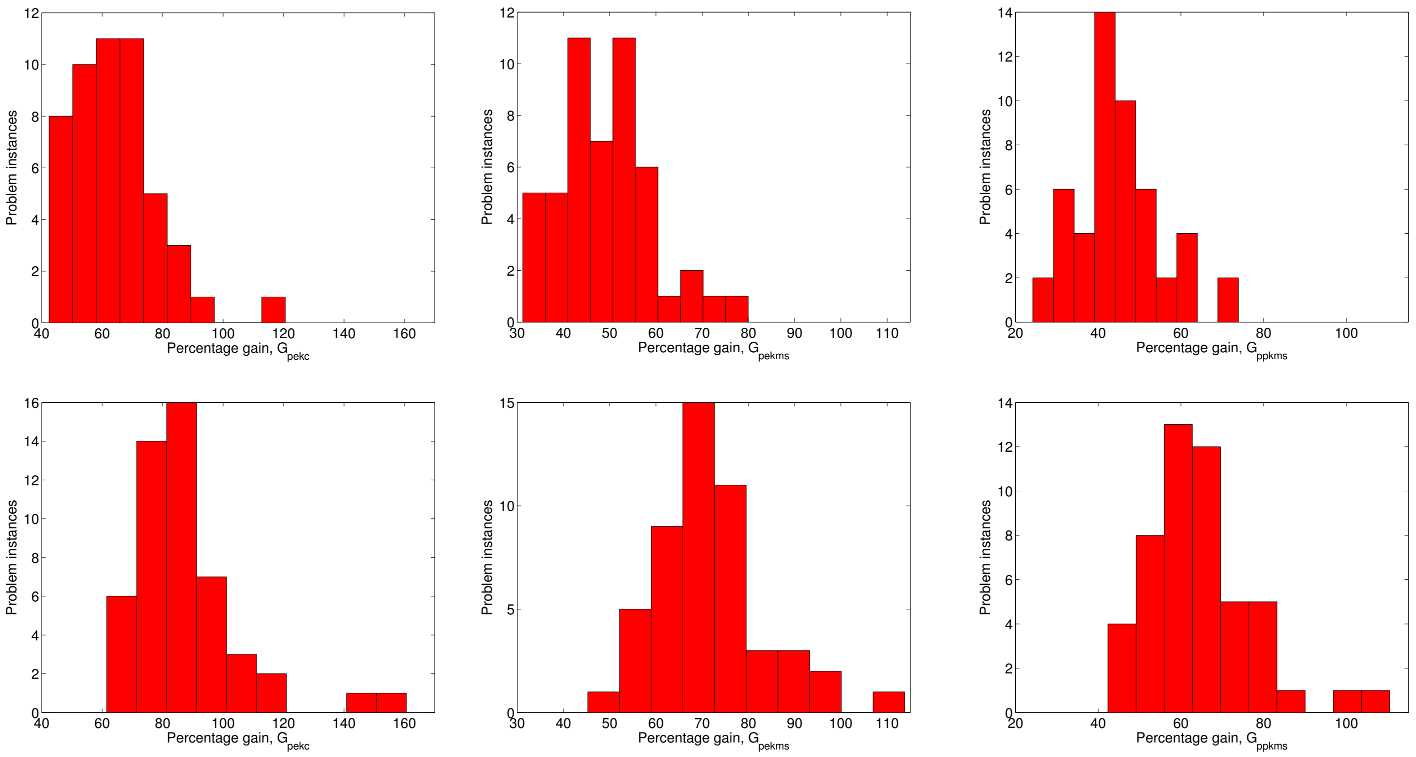
$$G_s = 100 \cdot \frac{z_{opt} - z_s}{z_s}, \quad s \in \{\text{'}pekms\text{'}, \text{'}pekc\text{'}, \text{'}ppkms\text{'}\}$$

where z_{opt} is the objective function value under the (approximately) optimal payments, and z_s is its *maximum* value, over all possible values of k, obtained with each of the three other payment allocation rules.

Two remarks are worth making regarding Fig. 4. First, the achievable gain with our scheme is anything but negligible. It varies significantly depending on how each time users and tasks are distributed in the physical space and the specific scheme under comparison; it is stochastically higher over the scheme that splits payments to the task-closest users and lower over the one that aligns payments with the skills of users. However, it does not fall below 25% (40%) for 50(resp. 100) users, while it exceeds 100% and even 150% in certain cases. Secondly, the gain over all schemes improves significantly when the pool of potential task contributors doubles in size. The proposed scheme maps the increased diversity of user skills and preferences more precisely and targets its payments more efficiently to maximize the quantity and quality of user contributions.

4.2 Multiple tasks, single-task offers

We now consider the case where multiple tasks are to be performed, and the mobile app may suggest at most one task per user. Single-task offers and incentives are targeted to a subset of users, while the rest of the users get no offer. The SP now needs to determine: (i) the subset of users to make



a. equal payments to the k closest users

b. equal payments to the k most skilled users

c. proportional payments to the k most skilled users

Figure 4: Histograms of the percentage gain in aggregate expected contribution quality achieved with the approximately optimal payment allocation when compared with the three individual-preference agnostic payment allocation schemes: $|\mathcal{L}|$=1, B=30, $|\mathcal{U}|$=50 (top row) and $|\mathcal{U}|$=100 (bottom row), 50 simulation runs.

offers to, (*ii*) the task to recommend to each user, and (*iii*) the payment to be offered to each user.

4.2.1 One user per task: Max-weight matching

First, consider the special case of the problem with one task recommended to each user. This scenario is meaningful if the task nature implies that the allocation of more than one users is not feasible or does not contribute to the quality of the task. This is the case in typical mobile crowdsourcing applications (*e.g.*, the crowdsourced delivery of parcels in [1] and the small job-to-user matching in TaskRabbit).

Consider the weighted bipartite graph $\mathcal{G}$, with node set $\mathcal{U} \cup \mathcal{L}$, edge set $\{e_{ul} : \forall u \in \mathcal{U}, \forall l \in \mathcal{L}\}$, and edge weights

$$\delta_{ul} = q_{ul} P_u(C_1|\mathbf{x}_{ul}) = q_{ul}\sigma(w_{ud}d_{ul} + w_{up}B_l)\,. \qquad (11)$$

Note that the entire budget B_l of each task l is allocated to one user. Then allocating tasks to users so that the total expected quality over all tasks is maximized reduces to solving the *max-weight matching* problem on $\mathcal{G}$.

4.2.2 More than one users per task: Mixed Integer Nonlinear problem

In more general cases, each task l is recommended to more than one users. Let $\mathcal{U}_l$ be the set of users to which task l is recommended. The emerging optimization problem is nontrivial to solve since the user selection per task and the budget allocation are coupled. For given user subsets $\mathcal{U}_l$ for each task l, budget allocation for each task is a continuous-valued (nonconvex) sigmoid optimization problem. However, budget allocation of a task l across users depends on the subset of users $\mathcal{U}_l$ to which task l is allocated. The determination of $\mathcal{U}_l$ is an integer-programming problem, and it is affected by the constraint that at most one task is allocated to each user. Therefore, the problem of task selection and payment allocation turns out to be a Mixed Integer Nonlinear Programming (MINLP) one and can be solved heuristically with numerical methods [3].

4.2.3 Evaluation methodology

Hereafter we assume that the SP decouples the two problems, by adopting a static rule for determining which tasks to be offered to users. Two examples of such rules are,

- “make an offer to a user for the task that lies closest to her”, implying that

$$\mathcal{L}_u \doteq \{l : d_{ul} \leq d_{ul'}, \forall\, l' \in \mathcal{L}, l' \neq l\}^1 \qquad (12)$$

- “make an offer to a user for the task that she is most skilled for”, whereby

$$\mathcal{L}_u \doteq \{l : q_{ul} \geq q_{ul'}, \forall\, l' \in \mathcal{L}, l' \neq l\} \qquad (13)$$

With respect to the single-task scenario, the decision feature set and the class context remain the same. We generate a synthetic training dataset in the same way and train the model using normalized features (ref. Eq. (9)) and cross-validation techniques. Having determined the sets of users

[1] If there are more than one tasks at exactly the same minimum distance, the task to include in $\mathcal{L}_u$ is chosen randomly.

that will receive offers for each task, the payment allocation proceeds separately for each one of them. Namely, the SP has to solve $|\mathcal{L}|$ different instances of the optimization problem,

$$\begin{aligned} &\textit{maximize} && \sum_{u:l\in\mathcal{L}_u} q_{ul}\sigma(w_{u,d}\cdot d_{ul,n}+w_{u,p}p_{ul,n}) \\ &\textit{subject to} && \sum_{u:l\in\mathcal{L}_u} p_{ul}\le B_l \\ &&& 0\le p_{ul}\le p_{max} \quad \{u: l\in\mathcal{L}_u\} \end{aligned} \tag{14}$$

one for each task $l \in \mathcal{L}$.

4.2.4 Numerical results

As with the single-task setting, and since there is only one constraint function for each instance of (14), the solutions of the branch-and-bound method are extremely close to the optimal ones. The gap between the upper and lower bounds for the objective function of (14) remains below 0.1 in all problem instances we have experimented with and below 0.01 in 90% of the instances.

Likewise favorable is the comparison of these (approximately) optimal payments with those determined under individual-preference-agnostic alternatives. As comparison reference for this scenario, we consider the scheme that pays users in proportion to their quality indices (see the *ppkms* scheme in section 4.1.2). This scheme consistently outperforms its competitors in our experimentation with single task scenarios (refer to Figs. 3 and 4) and sets a more demanding benchmark for the approximately optimal payment allocation scheme.

The comparison of the two solutions for payments allocation is summarized in Fig. 5 for an example scenario involving 150 users and 10 tasks. The gain of the approximately optimal scheme varies from 20% to 60%, depending on the rule for determining which task issues offers to each user (Eqs. (12) *vs.* (13)) and the way users and tasks are spread over the physical space. The advantage of our scheme is amplified when users receive offers from the task they are most skilled for, as can be seen from Fig. 5a and 5b. This becomes all the more important when looking at the *absolutes scores* of the scheme under the two ways users can be matched to tasks in Fig. 5c. The approximately optimal payment allocation is more efficient when users receive offers from tasks they are skilled for in that it results in more than 20% higher aggregate expected quality of user contributions.

4.3 Multiple tasks, offers for pairs of tasks

For given task attributes, the "contribute *vs.* not contribute" decision-making setting is probably the simplest a user may be presented with by an application. However, it is questionable whether it is the *right* one, *i.e.*, the one that will ease a positive response (contribution) from the user.

More specifically, it is well established that a comparison reference is always sought for when assessing alternatives. In the marketing community, the term "decoy effect" was introduced to characterize extended experimental evidence that consumers tend to predictably change their preferences between two options when they are presented with a third option that is asymmetrically dominated[2].

Our argument is that the application should be issuing task offers to users in ways that respect these psychological effects. Practically, the hint is that the offers for each task should not be made independently but rather in pairs (or even groups of larger size). In what follows, we discuss how the problem faced by the SP changes in that case.

4.3.1 Evaluation methodology

For this scenario, we draw on real data collected through an online customized questionnaire. Fifty people, mostly graduate students of our University, replied to the questionnaire that was put online for two weeks during early summer 2015. The questionnaire invites its participants to consider that they are visiting the city center during their leisure time and receive offers about tasks on their smartphones. The task description is minimal and neutral, to avoid bias effects due to individual interests of participants: users are asked to walk to an abstract place (only distance is provided) and take a couple of photos for some monetary reward. The task requests are presented in pairs and accompanied by two pieces of information: the monetary payment p awarded to those who carry out the task and the physical distance d that a user would need to travel in order to get to the physical location of the task. Hence, users are presented with instances of a two-feature choice problem with two alternatives.

The 50 questionnaire participants made 20 such choices sequentially, without having the option to go back to a previous choice and change it. The tasks that were paired within each offer were chosen carefully so that they present a tradeoff between the two task attributes, reward and physical location; that is, there was no instance where one task dominated the other by simultaneously featuring higher reward and smaller physical distance from the user. The questionnaire can be retrieved online[3].

The choice problem faced by the users in this service setting is different and so are both the class context and the feature set. More specifically, we label user u as C_1 at her m^{th} choice ($y_{um} = 1$), when she chooses to carry out the task that lies closer to her. The choice features are now four

$$\mathbf{x}_{ul_1l_2} = (d_{ul_1}, p_{ul_1}, d_{ul_2}, p_{ul_2}) \tag{15}$$

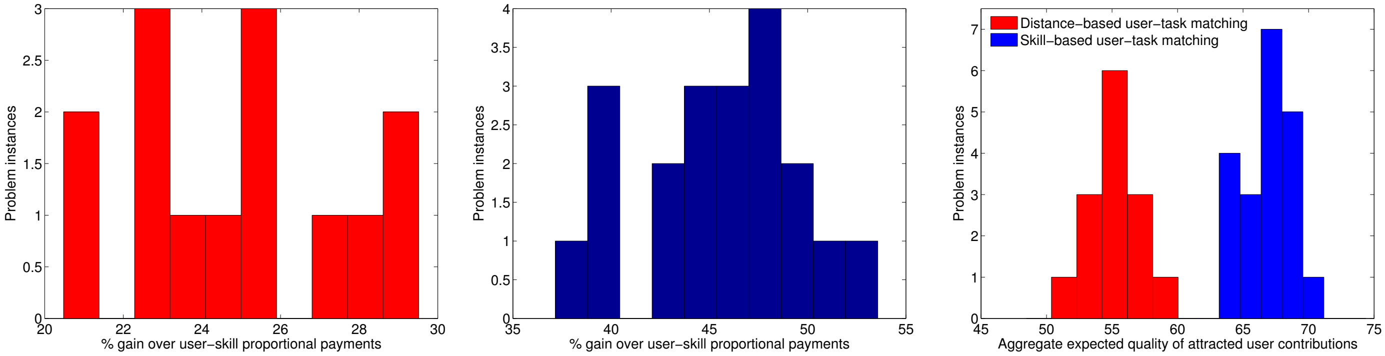
the notation $\mathbf{x}_{ul_1l_2}$ reflecting that the feature vector now involves attributes (*i.e.*, distances and payments) of two tasks rather than a single one: the distance (d_{ul_1}) and payment(p_{ul_1}) related to the task lying closer to the user, and the respective quantities for the task lying further away from her (d_{ul_2}, p_{ul_2}).

As with offers for a single task, the problem is originally a joint user-to-task and payment allocation problem. The complexity concerns can be overcome if a static rule is adopted for determining which tasks to be offered to which users. To this end, the rules presented in Section 4.2.2 could be paraphrased to "make an offer to the user for the two tasks that lie closest to her" so that

$$\mathcal{L}_u \doteq \{l_1, l_2 : d_{ul_1} \le d_{ul_2} \le d_{ul'}, \forall\, l' \in \mathcal{L}, l' \ne l_1, l_2\} \tag{16}$$

[2] One of the most pronounced instances of the effect in the marketing world relates to the way the Williams-Sonoma company used this cognitive bias to start sales on a bread maker with retail price set at $275 that no one was earlier buying. The company released a second bread maker (as a decoy), with slightly more capabilities but asymmetrically higher cost ($400). The larger machine did not sell itself but spectacularly increased the sales of the smaller bread maker by making it seem like a great purchase opportunity.

[3] http://5.101.107.163:8080/Questionnaire/ (in Greek)



a. Offer from the closest task

b. Offers from the task the user is most skilled for

c. Approximately optimal payments

Figure 5: (a,b) Approximately optimal payments *vs.* payments in proportion to the quality indices of users: % gain in the aggregate expected contribution quality, under the two options for matching task offers to users, eqs. (12) and (13), respectively. c) Aggregate expected contribution quality with optimal payments, under the two options for matching task offers to users: B_l=10 $\forall l \in \mathcal{L}$, $|\mathcal{L}|$=10, $|\mathcal{U}|$=150.

or "make an offer to the user for the two tasks that she is most suitable for", where

$$\mathcal{L}_u \doteq \{l_1, l_2 : q_{ul_1} \geq q_{ul_2} \geq q_{ul'}, \forall\, l' \in \mathcal{L}, l' \neq l_1, l_2\} \quad (17)$$

The optimization problem the SP faces is:

$$\begin{aligned} max \quad & \sum_{u \in \mathcal{U}} \sum_{l_1, l_2 \in \mathcal{L}_u} q_{ul_1} P(C_1|\mathbf{x}_{ul_1l_2}) + q_{ul_2}(1 - P(C_1|\mathbf{x}_{ul_1l_2})) \\ s.t \quad & \sum_{u: l \in \mathcal{L}_u} p_{ul} \leq B_l \quad \forall l \in \mathcal{L} \\ & p_{min} \leq p_{ul} \leq p_{max} \quad u \in \mathcal{U}, l \in \mathcal{L}_u \qquad (18) \end{aligned}$$

where

$$P(C_1|\mathbf{x}_{ul_1l_2}) = \sigma(\mathbf{w}_u \cdot \mathbf{x}_{ul_1l_2})$$

with $\mathbf{w}_u = (w_{u,d_1}, w_{u,p_1}, w_{u,d_2}, w_{u,p_2})$. p_{min} is the minimum payment offered to a user for any of the two tasks, and the remaining notation has been adapted to reflect the new class and feature vector definitions: w_{u,d_1} and w_{u,p_1} are the weights assigned by user u to the distance and payment of the closer task; and q_{ul_1} is the quality index measuring the expected quality of u's contribution to this task.

4.3.2 Numerical results

We consider two rules for determining which tasks a user should receive offers from, as encoded in Equations (16) and (17). For the resulting task user sets, we compare the payments induced by the solution of (18) with those emerging when distributing the task budget in proportion to the quality indices of its users. The gain achievable by our scheme is computed as in (11). Figure 6 plots the distributions of this gain over 30 instances of the problem with 50 users, 8 tasks, and budget equal to \$10 for each task. Its plots reflect the two main trends in this scenario. First, the second rule (Eq. (17)) for deciding which tasks to be offered to a user results in consistently higher values of aggregate expected quality of contributions. In Fig. 6c, this advantage is in the order of 60%, when looking at median values. Secondly, within the constraints of the specific user-to-task matching rules, the approximately optimal payment allocation scheme yields gains that are smaller and less variable when compared to the scenarios in sections 4.1 and 4.2.2 but still significant.

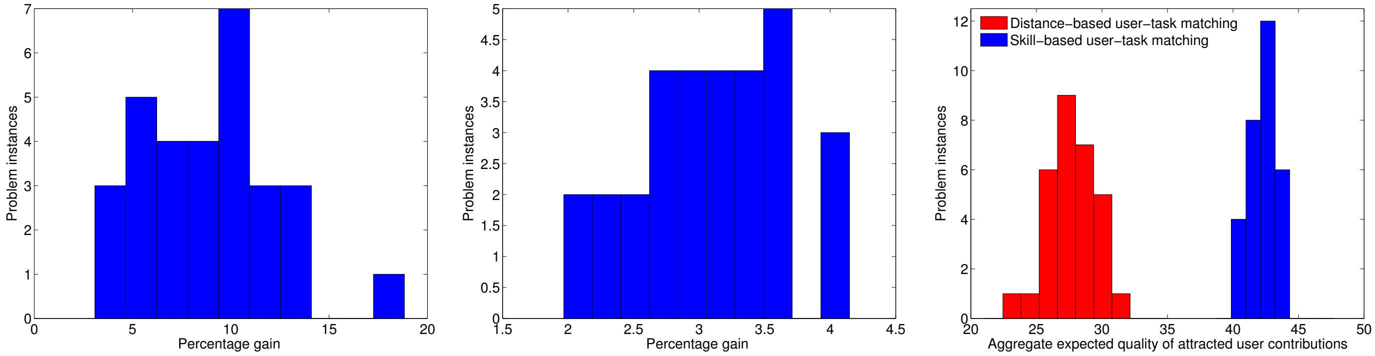
5. CONCLUSIONS AND DISCUSSION

In this paper, we draw on machine learning models and techniques to directly learn users' individual preferences with respect to crowdsensing tasks. We, then, build on this knowledge to optimize the recommendation of tasks to users and better target the provision of incentives to them. Our methods consistently outperform alternative task and payment allocation policies not accounting for individual preferences.

Our formulations and analysis in this paper are concerned with the maximization of the aggregate expected quality of user contributions to tasks. A natural direction for advancing the research thread this paper initiates would be to consider alternative objective functions or constraints reflecting per task performance requirements, *i.e.*, that each task should accumulate some minimum number of quality of contributions. Accommodating those in the sigmoid optimization framework would present an interesting challenge.

Another front for further work is presented by alternative classification models. In this work, we considered logistic regression as the probabilistic framework for modeling the choices of users in mobile crowdsensing campaigns. With such a modeling choice, the user heterogeneity was captured in different values of the feature vector $\mathbf{w}_u$ per user. However, other probabilistic classification models that have been developed in machine learning and statistics could be also considered. For instance, an interesting class of models is that of generative classifiers, which first estimate the class conditional densities, such as $p(\mathbf{x}|C_1)$, and then perform classification by computing posterior probabilities through Bayes' theorem [2]. Since these models are generative, *i.e.*, they model the distribution of user profiles, they could enable additional tasks such as detection of outlying behavior, where an observed user profile does not follow the estimated distribution. Subsequently, such detections could motivate specialized allocation payment procedures.

Finally, in this paper, we deal with the static one-shot incentive provision problem. Equally interesting is the dynamic variant of the problem, where tasks and payments are offered to users online taking also into account the current quality of attracted contributions per task, the residual task budget, and the remaining time before the end of the crowd-



a. Offer from the closest tasks

b. Offers from the tasks the user is most skilled for

c. Approximately optimal payments

Figure 6: (a,b) Approximately optimal payments *vs.* payments in proportion to the quality indices of users: % gain in the aggregate expected contribution quality, under the two options for matching task offers to users. c) Aggregate expected contribution quality with optimal payments, under two options for matching task offers to users: B_l=10 $\forall l \in \mathcal{L}$, $|\mathcal{L}|$=8, $|\mathcal{U}|$=50, p_{min}=0.25.

sensing campaign. The treatment of this problem could draw on the user profiling framework introduced in this paper and our ideas for online payment allocation in [10].

6. ACKNOWLEDGEMENTS

The work of the first two authors was supported by the European Commission H2020 Research Program under Grant Number 688768 netCommons (Network Infrastructure as Commons). We would also like to thank Madeleine Udell for valuable clarifications about her sigmoidal programming algorithm and related code.

7. REFERENCES

[1] Koiki, crowdsourced parcel delivery website. http://www.koiki.eu/es/, 2016.

[2] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Secaucus, NJ, USA, 2006.

[3] M. R. Bussieck and A. Pruessner. Mixed-integer nonlinear programming. *SIAG/OPT Newsletter: Views & News*, 14(1):19–22, 2003.

[4] R. Ganti, F. Ye, and H. Lei. Mobile crowdsensing: current state and future challenges. *Comm. Mag., IEEE*, 49(11):32–39, November 2011.

[5] L. Gao, F. Hou, and J. Huang. Providing long-term participation incentive in participatory sensing. In *Proc. 34th IEEE INFOCOM*, pages 2803–2811, Hong Kong, April 2015.

[6] H. Jin, L. Su, D. Chen, K. Nahrstedt, and J. Xu. Quality of information aware incentive mechanisms for mobile crowd sensing systems. In *Proc. 16th ACM Mobihoc*, pages 167–176, June 2015.

[7] I. Koutsopoulos. Optimal incentive-driven design of participatory sensing systems. In *Proc. IEEE Infocom*, pages 1402–1410, Turin, Italy, April 2013.

[8] J.-S. Lee and B. Hoh. Dynamic Pricing Incentive for Participatory Sensing. *Pervasive Mobile Computing*, 6:693–708, December 2010.

[9] H. Ma, D. Zhao, and P. Yuan. Opportunities in mobile crowd sensing. *IEEE Comm. Mag.*, 52(8):29–35, August 2014.

[10] P. Micholia, M. Karaliopoulos, I. Koutsopoulos, L. M. Aiello, G. De Francisci Morales, and D. Quercia. Incentivizing Social Media Users for Mobile Crowdsourcing. submitted, 2016.

[11] K. P. Murphy. *Machine learning: a probabilistic perspective.* MIT Press, Cambridge, MA, 2012.

[12] S. Nawaz, C. Efstratiou, and C. Mascolo. Parksense: A smartphone based sensing system for on-street parking. In *Proc. 19th ACM Mobicom*, pages 75–86, Miami, Florida, USA, 2013.

[13] D. Peng, F. Wu, and G. Chen. Pay as how well you do: A quality based incentive mechanism for crowdsensing. In *Proc. 16th ACM Mobihoc*, pages 177–186, Hangzhou, China, June 2015.

[14] R. Pryss, M. Reichert, J. Herrmann, B. Langguth, and W. Schlee. Mobile crowd sensing in clinical and psychological trials – a case study. In *Proc. 28th IEEE CBMS*, pages 23–24, Sao Paolo, Brazil, June 2015.

[15] R. K. Rana, C. T. Chou, S. S. Kanhere, N. Bulusu, and W. Hu. Ear-phone: An end-to-end participatory urban noise mapping system. In *Proc. 9th ACM/IEEE IPSN*, pages 105–116, Stockholm, Sweden, 2010.

[16] C.-K. Tham and T. Luo. Quality of contributed service and market equilibrium for participatory sensing. *IEEE Trans. on Mobile Computing*, 14(4):829–842, April 2015.

[17] M. Udell and S. Boyd. Maximizing a sum of sigmoids. *accepted to Optimization and Engineering*, 2013.

[18] Y. Ueyama, M. Tamai, Y. Arakawa, and K. Yasumoto. Gamification-based incentive mechanism for participatory sensing. In *Proc. IEEE PERCOM Workshops*, pages 98–103, Budapest, Hungary, March 2014.

[19] Y. Wang, X. Liu, H. Wei, G. Forman, and Y. Zhu. Crowdatlas: Self-updating maps for cloud and personal use. In *Proc. 11th ACM Mobisys*, pages 469–470, Taipei, Taiwan, 2013.

[20] D. Yang, G. Xue, X. Fang, and J. Tang. Crowdsourcing to smartphones: Incentive mechanism design for mobile phone sensing. In *Proc. 18th ACM Mobicom*, pages 173–184, Istanbul, Turkey, 2012.