PODIUM: Procuring Opinions from Diverse Users in a Multi-Dimensional World

Yael Amsterdamer[1] and Oded Goldreich[2]

[1]Department of Computer Science, Bar-Ilan University, Israel

[1]first.last@biu.ac.il
[2]first.last@live.biu.ac.il

Abstract

The procurement of opinions is an important task in many contexts. When selecting members of a certain population to ask for their opinions, *diversity* inside the selected subset is a central consideration. People with diverse profiles are assumed to provide a wider range of opinions and thus to better represent the opinions of the entire population. However, in platforms with a large user base such as crowdsourcing applications and social networks, defining and realizing notions of diversity are both nontrivial. The profiles of users typically contain information that is high-dimensional and semantically rich. We present PODIUM, a tool for opinion procurement that accounts for complex user profiles and enables customizable user selection. Beyond selecting a subset of users with diverse profiles, PODIUM produces explanation for the choice of each user and visual aids to compare the selected subset to the entire population on different dimensions. We demonstrate the use of PODIUM on the TripAdvisor user base, which further enables us to examine the ability of our system to predict diverse opinions in user reviews.

1 Introduction

The need to procure a diverse and representative set of opinions arises in multiple contexts, such as surveys, market research, and crowdsourcing applications. Consider, for example, a traveler planning a trip and looking for specific "tips" on some destination; an owner of a new restaurant wishing to perform a preliminary customer survey; or a website manager seeking usability feedback. Platforms such as TripAdvisor (`https://www.tripadvisor.com`), that have a large user base and high-dimensional, rich data on each user, provide an opportunity for procuring opinions from a diverse set of users. At the same time, these characteristics of the data also pose challenges in realizing this potential: how do we *define* diversity while accounting for high-dimensional data? Can we *efficiently compute* a diverse subset of users? Can the resulting selection be *explained to* and *customized by* the client user?

The latter challenge is of particular interest since the requirements on user selection may greatly vary across different scenarios. For instance, a traveler may seek the opinions of users with different culinary preferences, whereas a website manager may seek feedback from users with diverse activity history.

Previous work on diversification either focus on covering a range or a set of categories, and thus do not account for covering the full range of opinions in multiple dimensions, which is provided in user profiles and can be leveraged for user selection. Moreover, explanations and customization has not been considered in this context. See Section 2.4 for details.

To address these challenges, we introduce PODIUM: a novel tool for the procurement of diverse opinions, utilizing multi-dimensional user profiles. Our solution consists of the following components:

User Profile Model The model that we consider for user profiles enables capturing, in a uniform manner, personal characteristics of users (e.g., nationality) and their past interactions with the platform (e.g., feedback they provided on restaurants). These properties are associated with a score from some range (Boolean, rating score, etc.) and thus form high-dimensional data.

Capturing Diversity Different notions of diversity has been considered in the literature (see Section 2.4). In the present work, we focus on a notion of diversity that is *coverage-based*, *customizable* and designed for the multi-dimensional, rich contents of user profiles, as briefly explained next.

Coverage-based diversity aims to select a set (of users, in our case) that in some sense represents or "covers" many of the different, possibly overlapping *groups* within a source population [1, 8, 9]. This class of diversity notions fits typical scenarios of opinion procurement (e.g., surveys, market research) in comparison with *distance-based diversity*, which focuses on maximizing the differences between the members of the selected group [9]. We provide a specific problem definition for coverage-based diversity in our setting, relying on the available properties in user profiles.

Customizable diversity allows the client an informed control over the user groups/data dimensions whose coverage is targeted. For that, we define general notions of *explanations* for the user selection result, which enable visualizing the coverage of different user properties and the role of such properties in the selection of each user. The client can then provide, via a user-friendly interface, a *feedback* with well-defined semantics that serves to refine the user selection.

Complexity and Algorithms Based on our model, we formalize the problem of optimizing user subset diversity. The corresponding decision problem is NP-complete and thus we employ an effective greedy algorithm with provable approximation guarantees.

Demonstration We will demonstrate PODIUM in the context of travel-related opinion procurement, based on real data from TripAdvisor. The demonstration will be interactive, allowing participants to select users for opinion procurement about destinations of interest, showing them explanations of the user selection process and engaging them in refining selection criteria accordingly. We will

Property	Alice	Bob	Carol	David	Eve
livesIn	Tokyo$^{(2)}$	NYC$^{(1)}$	Bali$^{(1)}$	Paris$^{(1)}$	Tokyo
ageGroup	50-64$^{(1)}$	–	25-34$^{(1)}$	–	–
avgRating Mexican	0.75$^{(3)}$	0.8	–	0.7	0.25$^{(1)}$
visitFreq Mexican	0.8$^{(2)}$	0.25$^{(2)}$	–	0.75	0.2
avgRating CheapEats	0.1$^{(2)}$	0.75$^{(2)}$	0.6	–	0.2
visitFreq CheapEats	0.5$^{(2)}$	0.85	0.25$^{(2)}$	–	0.2

Table 1: Example user profiles

further show an online evaluation of the selection results by running PODIUM on data where actual user reviews are already known. See Section 4 for details.

2 Technical Background

2.1 User Profiles

Let $\mathcal{U}$ be a population of users and $\mathcal{P}$ be some domain of properties. Following [2], we define the profile of a user $u \in \mathcal{U}$ as a tuple $D_u = \langle P_u, S_u \rangle$ where $P_u \subseteq \mathcal{P}$ is a set of properties relevant to u and $S_u : P_u \to [0, 1]$ maps each property to a score (normalized to $[0, 1]$). This score may have different interpretations depending on the type of property, e.g., true/false, user rating, and so on, and may be provided directly by u or automatically derived from u's activity in the website.

Example 2.1 *Table 1 shows a few profiles from a travel website (for now, ignore the numbers in superscript). In the first two rows,* `livesIn <city>` *and* `ageGroup <X-Y>` *are true/false properties for relevant cities and age ranges. E.g.,* `livesIn Tokyo` *is a property with score 1 (i.e., true) in Alice's profile. The third and fifth rows show scores that reflect the user average ratings for different types of restaurants, normalized to* $[0, 1]$*. Note that properties are usually not recorded for every user, e.g., Carol has never rated Mexican food. The fourth and sixth rows show scores reflecting the relative frequency that each of the users visits different types of restaurants.*

In practice, user profiles can contain many properties. This may be due to a diverse activity of a user in the system (e.g., providing opinions about many types of destinations), due to various analyses performed over the data (e.g., one can compute the average rating, maximum rating...) and so on. In the dataset that we have constructed from the TripAdvisor website (see Section 3), each user has up to 2189 properties.

2.2 Diversity Notion

We define diversity based on user properties and their scores. Scores are crucial in this respect: e.g., it makes sense to group Mexican food lovers and dislikers separately, rather than grouping all the Mexican food reviewers together. For that, we split the range of scores of each property $p \in \mathcal{P}$ into a set of *non-overlapping buckets* $\beta(p)$ (the bucketing method is described in Section 2.3). A

group is then the subset of users with relevant property and score, formally,

$$g_{p,b} := \{u \in \mathcal{U} \mid D_u = \langle P_u, S_u \rangle \wedge p \in P_u \wedge b \in \beta(p) \wedge S_u(p) \in b\}$$

We are now ready to define our notion of diversity, as follows.

Definition 2.2 *Let $\mathcal{G}$ be the set of all non-empty groups $g_{p,b}$ with respect to a given set of users $\mathcal{U}$ and profiles $\{D_u\}_{u \in \mathcal{U}}$. We say that a subset of the users $U \subseteq \mathcal{U}$* covers *a group $g_{p,b}$ if $|U \cap g_{p,b}| > 0$, and denote the set of covered groups by* $\mathrm{cov}(U)$. *Further let $W : \mathcal{G} \to \mathbb{R}^+$ be a* weight function *indicating the importance of each bucket. Given also a budget $B \in \mathbb{N}$, we define* `MAX-DIVERSITY` *as the problem of finding a subset $U \subseteq \mathcal{U}$ such that $|U| \leq B$ and the weight of U, defined as $W_{\mathcal{G}}(U) := \Sigma_{g_{p,b} \in \mathrm{cov}(U)} W(g_{p,b})$, is maximized.*

The weight function W captures the importance of different groups. A natural choice, which we employ in our implementation, is defining the weight of a group as the *number of users* in it. The purpose of this choice is to increase the likelihood of covering large groups before small ones.

Example 2.3 *Reconsider the user profiles in Table 1 and assume that each property is divided into two buckets: scores in $[0.5, 1]$ ("high") and scores in $[0, 0.25]$ ("low"). The numbers in superscript show the weights – i.e., number of users – on the first occurrence of each property bucket. E.g., there is only one bucket with 3 users:* `avgRating Mexican high`*. In this case, the user group of size 2 that should be selected is {Alice, Carol} with sum of weights 18.*

Customization and Explanations The *explanation* of a selected subset $U \subseteq \mathcal{U}$ is the partition of $\mathcal{G}$ to $\langle \mathrm{cov}(U), \mathcal{G} - \mathrm{cov}(U)$, namely, covered and non-covered groups in $\mathcal{U}$. The explanation of a selected user $u \in U$ is $\mathrm{cov}(\{u\})$, namely, the groups covered by u's properties. Intuitively, the properties of $\mathcal{P}$ are assumed to have meaningful names since they are used in user profiles, and score buckets can also be easily given names ("high", "medium"...), which yields meaningful names to the groups of $\mathcal{G}$. The client user can then see which groups are covered (and by which selected user) and which are not. (The UI of PODIUM further enables easy and intuitive browsing through these groups, see Section 3.)

A *customization feedback* of the user is composed of four distinct subsets of $\mathcal{G}$, denoted $\mathcal{G}_+, \mathcal{G}_-, \mathcal{G}_d$ and $\mathcal{G}_{d?}$. $\mathcal{U}$ is refined to consider only users of interest $\mathcal{U}_c$, who belong to every group in $\mathcal{G}_+$ (if $\mathcal{G}_+$ contains more than one bucket of some property p, users need only belong to one of them) and to none in $\mathcal{G}_-$. Formally,

$$\begin{aligned}\mathcal{U}_c = &\{u \in \mathcal{U} \mid \forall g_{p,b} \in \mathcal{G}_+,\ \exists b' \in \beta(p) :\ u \in g_{p,b'} \wedge g_{p,b'} \in \mathcal{G}_+\} \\ &\cap \{u \in \mathcal{U} \mid \forall g_{p,b} \in \mathcal{G}_- :\ u \notin g_{p,b}\}\end{aligned}$$

The customized diversity problem `MAX-DIVERSITY-C` is then to select new subset $U \subseteq \mathcal{U}_c$ that maximizes $W_{\mathcal{G}_d}(U)$, namely, the sum of weights over covered groups from $\mathcal{G}_d$, breaking ties by $W_{\mathcal{G}_{d?}}(U)$. Note that `MAX-DIVERSITY-C` can be easily reduced to `MAX-DIVERSITY` by a proper selection of weight function.

Example 2.4 *Reconsider the problem of selecting a user subset of size 2 from Table 1, but now assume the client prefers users from diverse locations and*

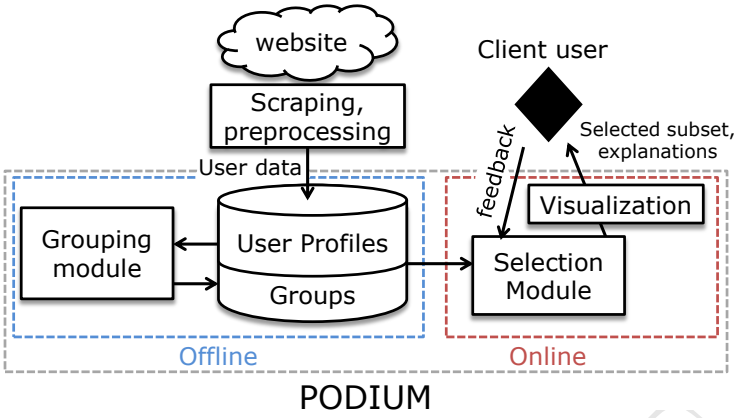

Figure 1: Architecture and auxiliary components

people familiar with Mexican food. This translates to a feedback where $\mathcal{G}_+$ *consists of the two buckets* `high` *and* `low` *of* `AvgRating Mexican` *(requiring the users to have any rating for Mexican food), and* $\mathcal{G}_d$ *consists of the different* `livesIn <city>` *properties.* $\mathcal{G}_-$ *and* $\mathcal{G}_{d?}$ *would be* $\emptyset$ *and* $\mathcal{G} - \mathcal{G}_d$*, respectively. Then, the refined user set* $\mathcal{U}_c$ *will exclude Carol who did not rate Mexican food. The best user subsets will now be* $\{Alice, Bob\}$ *or* $\{Bob, Eve\}$*, which maximize the sum of weights over* `livesIn <city>` *properties (to 3) and among other subsets that achieve this maximum (e.g.,* $\{Alice, David\}$*) the former two subsets further maximize the sum of weights over other properties (to 14).*

2.3 Diversity Computation

We next consider the computation of a diverse subset of users.

Computing $\mathcal{G}$. The set of properties $\mathcal{P}$ is assumed to be given (derived from the user profiles). To compute the buckets $\beta(p)$ for any property $p \in \mathcal{P}$ we first determine the number of buckets heuristically as $\lceil \log \text{user}(p) \rceil$, where $\text{user}(p)$ is the number of users with property p in their profiles. For sanity, we bound this number from above by $\text{uniq}(p)$, the number of unique scores obtained for p. We then use one-dimensional clustering (based on k-means) to split the score range $[0 : 1]$ into buckets.

Solving MAX-DIVERSITY Unsurprisingly, achieving an optimal solution is intractable unless P=NP, even for simple weight functions. The decision problem `DEC-MAX-DIVERSITY` corresponding to `MAX-DIVERSITY` is that of the existence of a subset of a given cardinality B such that the sum of weights of covered groups exceeds a threshold T. We can then show:

Proposition 2.5 *DEC-MAX-DIVERSITY is NP-complete in* B*.*

Despite this intractability result, our problem enables efficient approximation, due to properties of $W_{\mathcal{G}}(U)$. First, $W_{\mathcal{G}}(U)$ is *submodular*, namely, for any $U \subseteq U' \subseteq \mathcal{U}$ and $u \in \mathcal{U}$ it holds that $W_{\mathcal{G}}(U \cup \{u\}) - W_{\mathcal{G}}(U) \geq W_{\mathcal{G}}(U' \cup$

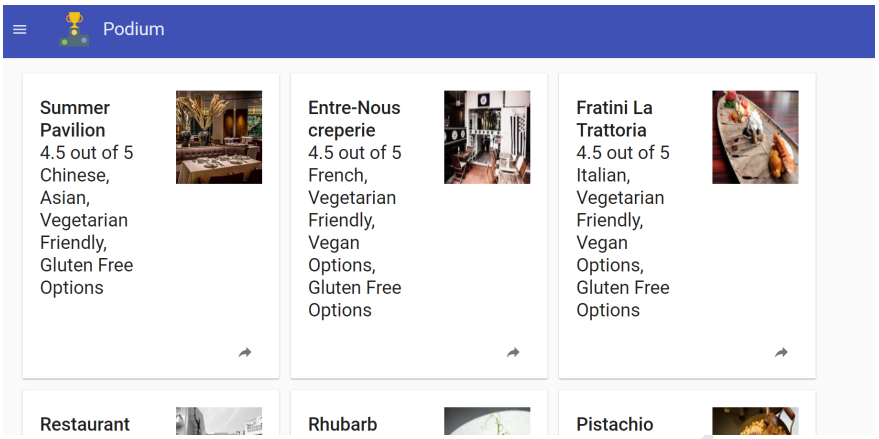

Figure 2: Search results

$\{u\}) - W_{\mathcal{G}}(U')$. Since the group weight function W is non-negative, we get that $W_{\mathcal{G}}(U)$ is also non-negative ($W_{\mathcal{G}}(U) \geq 0$) and *monotonous* (if $U \subseteq U'$ then $W_{\mathcal{G}}(U) \leq W_{\mathcal{G}}(U')$). The size of the groups that we consider is bounded by B. For such functions, a greedy algorithm that iteratively adds one user u to the selected subset U so as to maximize $W_{\mathcal{G}}(U \cup \{u\})$ is known to guarantee a good approximation ratio $(1 - 1/e)$ [7]. We implement such a greedy algorithm in PODIUM.

2.4 Related Work

The selection/retrieval of a diverse subset has been studied in many contexts, including but not limited to search engines (e.g., [1, 8]), recommender systems (e.g., [3, 4]) and crowdsourcing (e.g., [6, 9]). The present work is motivated by the procurement of diverse *opinions*, and thus considers the *full range* of scores assigned to any property, accounting for e.g. low and high ratings. This is in contrast with the coverage of document topics [1, 4] or skill coverage in task assignment [6]. The recent work of [9] is the most relevant to ours since it also studies diverse opinion procurement. However, they do not consider multi-dimensional data nor customization. While our approach explicitly relies on a predefined set of properties for the grouping of users, other approaches may attempt to *compute* the "best" groups using clustering methods (e.g. [3]). However, for such approaches, the explanation and refinement of the groups may be highly cumbersome to a client, and thus they are not practical for customization. Previous work has studied customizable user selection in different setting (e.g., [2, 5]), and is complementary to our present study which focuses on diversification.

3 System Overview

PODIUM is implemented in Python, using Flask (`http://flask.pocoo.org`). Its architecture is depicted in Figure 1. The input to PODIUM is a set of user profiles, as explained in Section 2.1, in JSON format.

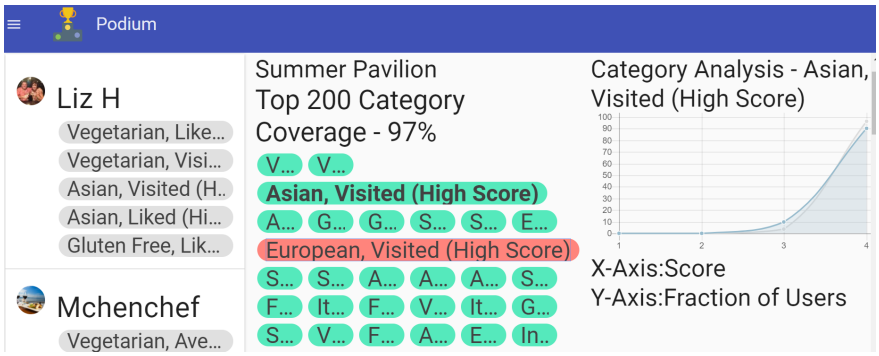

Figure 3: Selected group screen

Given a set of user profiles, the *Grouping Module* computes the bucketing of properties and the weights of groups in an offline process. PODIUM also enables the administrator to feed in an *initial set of diversification configurations* with associated textual descriptions, which the client can search, preview and refine. For example, Figure 2 displays the search results for "restaurants in Singapore" over a set of initial configurations defined on TripAdvisor data. The result titled X is a configuration based on the properties of restaurant X, e.g., `avgRating <Y>` where `Y` is a cuisine of X.

The Graphical User Interface of PODIUM was created using AngularJS 1.6.4 (https://angularjs.org). Given a client request, the *Selection Module* executes the user selection algorithm and returns the selected subset and its explanations to the client via the *Visualization* module. Figure 3 shows the result page for the initial configuration "Summer Pavilion". The left pane displays the names of selected users, along with the top-weight groups that were covered by each. The middle pane shows the percentage of top-weight relevant groups (in $\mathcal{G}_d$ and $\mathcal{G}_{d?}$) covered by the selected subset (in this case, 97%). The list of groups ordered by decreasing weight is displayed below, with covered groups in green and the others in red.[1] When clicking any group, the right pane displays a graph comparing the score distribution between the entire population and the selected subset for the relevant property (in Figure 3 the distributions are almost identical). Users can browse the different groups and refine the selection by adding groups to $\mathcal{G}_+$ and $\mathcal{G}_-$ ("selected users must / not have this property"); and to $\mathcal{G}_d$ and $\mathcal{G}_{d?}$ ("Do not / diversify on this property") via the user-friendly interface.

Dataset for demonstration In addition to the inherent components of PODIUM, Figure 1 shows an auxiliary module we have developed to create the dataset for the demonstration. To extract user data from TripAdvisor we have developed a scraper using the Selenium (`http://www.seleniumhq.org`) library. The raw data contains both user submitted data (e.g. age, residence) and user activity data (e.g. reviews and ratings). It is then pre-processed and normalized to the required format. We enrich the categories in the raw data using the Foursquare (`https://foursquare.com/`) taxonomy, to generalize, e.g., Mexican cuisine to South American cuisine. Example properties include "average

[1]For space constraints, some group names in Figure 3 are truncated.

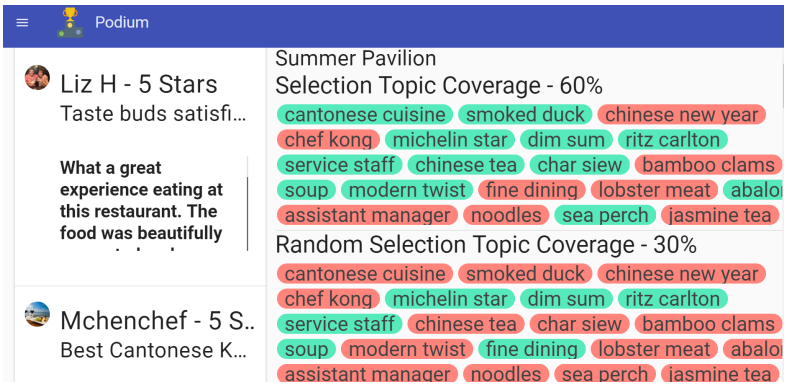

Figure 4: Prediction evaluation screen

rating of Mexican Cuisine" whose score is the average rating, and "visited Asian cuisine" whose score is the fraction of reviews by this user on Asian restaurants.

4 Demonstration

We will demonstrate the use of PODIUM for the procurement of travel-related diverse opinions from TripAdvisor users, using the previously described dataset (Section 3). To enable an online evaluation of PODIUM's *prediction*, namely, the diversity of actual opinions procured from the selected users, we have enhanced the UI of PODIUM with an evaluation screen (Figure 4). During the demonstration, PODIUM will only use TripAdvisor data up to Nov. 2016 to select users. Later reviews by the selected users will serve as the procured opinions – data that is not apriori available in a typical use of PODIUM. The left pane in Figure 4 shows the real review/rating of each user for the selected travel destination, and the middle pane shows different (visualizations of) useful statistics. These include the coverage of topics (identified by TripAdvisor for the selected destination) by the reviews of selected users (on top) compared with a random subset of users (on the bottom). For instance, In Figure 4, the reviews of ten users selected by PODIUM covered twice as many topics.

We will begin the demonstration by asking a volunteer from the audience to choose an initial configuration. For instance, the volunteers can ask about the hotel they currently reside at, or about a site they plan to visit. PODIUM compute a diverse set of TripAdvisor users among the ones who wrote a review about the chosen destination after November 2016 (the *reference population*). For fairness, PODIUM will ignore previous reviews/ratings about this destination by the reference population, if exist.

We will then view the selected subset of users and its visual explanation. The explanation lists many properties that are relevant for the chosen destination as covered and uncovered groups. The comparison of score distribution provides intuition on how well the selected subset represents the opinions in the entire population, and we will use it to inspect the results for properties of interest.

Then, we will ask the volunteers to experiment with the controls and change the user selection according to their personal preferences. For example, if the "Vegetarian" category is highly weighted in the data, but the clients are not

interested in vegetarian food, they may use the UI controls to exclude the corresponding category from consideration by the algorithm, or enforce choosing users that give low scores to vegetarian restaurants. As another example, the client will be able to obtain better diversification of certain groups in the population, e.g., gender, age, residence, etc.

Ongoing and Future Work This demonstration focuses on a use case and data which are particularly intuitive for the conference participants. Evaluating the quality of our system over various real-life scenarios is the aim of our ongoing and future work.

Acknowledgments

This work was supported in part by the BIU Center for Research in Applied Cryptography and Cyber Security in conjunction with the Israel National Cyber Bureau in the Prime Ministers Office; and by the Israel Science Foundation (grant No. 1157/16).

References

[1] R. Agrawal, S. Gollapudi, A. Halverson, and S. Ieong. Diversifying search results. In *WSDM*, 2009.

[2] Y. Amsterdamer, T. Milo, A. Somech, and B. Youngmann. December: A declarative tool for crowd member selection. *PVLDB*, 9(13), 2016.

[3] R. Boim, T. Milo, and S. Novgorodov. Diversification and refinement in collaborative filtering recommender. In *CIKM*, 2011.

[4] K. El-Arini, G. Veda, D. Shahaf, and C. Guestrin. Turning down the noise in the blogosphere. In *SIGKDD*, 2009.

[5] J. Fan, G. Li, B. C. Ooi, K. Tan, and J. Feng. iCrowd: An adaptive crowdsourcing framework. In *SIGMOD*, 2015.

[6] P. Mavridis, D. Gross-Amblard, and Z. Miklós. Skill-aware task assignment in crowdsourcing applications. In *Int. Sym. Web Algo.*, 2015.

[7] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functionsI. *Math. Prog.*, 14(1), 1978.

[8] M. Servajean, E. Pacitti, S. Amer-Yahia, and P. Neveu. Profile diversity in search and recommendation. In *WWW*, 2013.

[9] T. Wu, L. Chen, P. Hui, C. J. Zhang, and W. Li. Hear the whole story: Towards the diversity of opinion in crowdsourcing markets. *PVLDB*, 8(5), 2015.