Industry and Applications Paper

Diverse User Selection for Opinion Procurement

Yael Amsterdamer
Bar-Ilan University
first.last@biu.ac.il

Oded Goldreich
Bar-Ilan University
first.last@live.biu.ac.il

ABSTRACT

Many applications maintain a repository of user profiles with semantically rich information on each user. Such repositories have a potential of allowing active *opinion procurement*: reaching out to users to ask for their opinions on different topics. An important desideratum of the procurement process is that it targets a *diverse* set of users.

To realize this potential, we present `Podium`: a first framework, to our knowledge, that supports the selection of diverse representatives in presence of high-dimensional, semantically rich user profiles. We demonstrate that data dimensionality is a challenge for both defining and achieving diversification. We address these challenges by proposing a lightweight, flexible notion of diversity that in turn allows explanations and customization of diversification results. We show that the problem of finding an optimally diverse user subset is intractable, and provide a greedy algorithm that computes an approximate solution. We have implemented our solution in a system prototype and tested it on real-world crowdsourcing platform data. Our experimental results show that `Podium` is effective in selecting users with diverse properties, and in turn that the opinions of these users are diverse according to multiple metrics.

1 INTRODUCTION

Multiple applications involve active procurement of opinions from users. 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. A recurring desideratum in such applications is that procured opinions are *diverse*: the restaurant owner may seek users with diverse culinary preferences who live in a certain region, whereas the website manager may seek feedback from users with diverse activity history. Notably, diversity considerations may greatly differ between scenarios, even if users are selected from the same set.

Platforms such as Yelp[1] 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. Yet, to our knowledge, there is no generic solution for selecting diverse representative users accounting for high-dimensional user profiles. In particular, users chosen for opinion procurement should ideally reflect the full range of user properties as observed in the source population – e.g., the full range of opinions on different topics, from positive to negative; the full range of user skills or activity levels, from low to high; etc. Hence, existing diversification solutions that target the overall accuracy of user answers/relevance of items and therefore operate by optimizing properties across multiple axes (e.g., selecting users' highest skills or activity levels) are inapplicable in this context, as explained in Section 2.

[1]Yelp website: https://www.yelp.com

To this end, we introduce `Podium`: a novel tool for the procurement of diverse opinions, utilizing multidimensional user profiles. We next overview our main contributions.

Model. Our model captures user profiles including both personal details provided by the users and their past interactions with the platform. These properties may be associated with a numeric score (reflecting, e.g., rating) and form high-dimensional data. We then provide a formal definition of the diverse user selection problem that is *coverage-based* [1–4]: i.e., the goal is selecting a user subset that in some sense represents or "covers" many of the different, possibly overlapping *groups* within a source population. This class of diversity notions fits typical scenarios of opinion procurement (e.g., surveys, market research), in contrast with *distance-based diversity*, which focuses on maximizing the differences between the members of the selected group [4–7]. As observable from Table 1, our diversity notion fulfills a unique combination of desiderata that arise at an opinion procurement scenario. We overview the desiderata and the compared solutions in Sections 2 and 9. We further propose an operative method for computing user groups from a repository of profiles, along with weight functions to prioritize the coverage of these groups, where the coverage of every group is impossible.

Analysis of the Basic Problem. Based on our model, we develop a solution to the diverse user selection problem. First, we show by a reduction from Set Cover that the decision problem corresponding to user selection in our context is NP-complete, and that finding a user subset of size approximately minimal that covers all the possible groups is also computationally hard. Moreover, in a high-dimensional setting, full coverage would typically require an unrealistically large number of procured opinions. Thus, instead of targeting full coverage and optimizing the subset size, we bound the size according to some budget and aim to select a user subset of that size that maximizes the *total coverage score*, to be defined in Section 3. Fortunately, a user subset whose coverage score is within a constant factor of the optimal can be found in PTIME. We show a simple greedy algorithm that achieves this bound, explain its data structures and optimizations, analyze its time complexity and demonstrate its operation on a sample user repository.

Customization and explanations. The required notion of diversity may vary based on the concrete application and depending on the multiple dimensions of user data, as exemplified above with respect to the different needs of a traveler versus restaurant owner versus website manager. We thus adopt a lightweight solution that facilitates interpretation of the results and in turn allows the clients to interact with the system to customize and fine-tune user selection. This is achieved through a formal definition of *explanations* for how the selected subset covers the population groups and the contribution of each selected user. We then formally define the semantics of a *user feedback* that allows an informed control over the user groups/data dimensions whose coverage is targeted. We extend our problem definition and analysis accordingly.

System	Type		Range	High-Dimension	Explanations	Customizable
Podium	Coverage-based	Intrinsic	✓	✓	✓	✓
Cohen & Yashinski [2]	Coverage-based	Intrinsic	✓*			✓
Stratified sampling (e.g., [8])	Coverage-based	Intrinsic			✓	✓
T-Model [4]	Coverage-based	Predicted	✓*			
APM [3], IA-Select [1]	Coverage-based	Predicted				
Yu et Al. [7]	Distance-based	Intrinsic		✓	**	
S-Model [4], DiRec [5]	Distance-based	Intrinsic	✓***	✓***		
DivRSci [6]	Distance-based	Predicted	✓***	✓***		

Table 1: Comparison of selected diversification solutions, according to the aspects discussed in Section 2. A diversity notion fulfills **Range** if it can diversify along a range of values (low to high) rather than just among categories, and **High-Dimension** if every candidate may be associated with a high number of properties. See Section 9 for more details on these solutions. Remarks: * Can support range coverage on a single dimension/property. ** Explanation for item relevance rather than subset diversity. *** Depends on the choice of distance function.

Implementation and experiments. We have implemented our solution in Podium, a prototype system including back-end implementation of our diverse user selection algorithm and a front-end that provides visualizations for our notions of explanations and user-friendly means of providing customization feedback (see Figure 1 for its architecture). We use this prototype to examine our approach over data from large-scale real-life user repositories. We first study the performance of our approximation algorithm, showing that it is effective in achieving diversity in terms of the selected user profiles according to the target function it approximates as well as multiple other diversity metrics. Next, we simulate opinion procurement using our algorithms (using ground truth user opinions), and test the diversity of procured opinions according to different metrics. Finally, scalability tests support the practicality of our algorithm for real-world data.

Paper Outline. In Section 2 we describe and motivate the desiderata from a diversification system in our context. Section 3 presents our model and basic problem definition, without customization, and in Section 4 we develop and analyze our solution for this basic problem. Next, we extend the basic solution to support the explanation and customization of the selection results in Sections 5 and 6 respectively. We describe our implementation of Podium in Section 7 and the experimental study conducted over it in Section 8. Section 9 discusses related work and we conclude in Section 10.

2 DESIDERATA

Diversification has been extensively studied in multiple contexts; we claim that diversification in the concrete context of opinion procurement has a *unique combination of traits*, which are not accounted for by previous work. We compare several representative previous solutions under the prism of these traits in Table 1. Next, we explain these features as well as the desiderata of diversification that follow; further detailed comparison with related work is given in Section 9.

Coverage vs. distance-based. A prominent approach for diversification is to quantify the (dis)similarity between items, and to then aim at finding items that optimize some aggregate function over the similarity scores, for instance, maximizing the minimal pairwise distance (e.g., [4–7]). Such an approach is valid in our setting, yet its sensitivity to skews in group sizes may yield less meaningful results for real-life datasets, as observed in our experimental results for the Yelp dataset in Section 8.

When it comes to gathering user opinions, a natural desideratum is that opinions are collected from users that in some sense faithfully represent the characteristics of the full population. Such representativeness is targeted by *coverage-based* approaches in different selection contexts – e.g., retrieving documents that cover the topics in a repository, or users that represent predefined groups within a source population (e.g., [1–4]). In contrast with distance-based approaches, coverage-based approaches can in particular be agnostic of the similarities within the selected subset.

We next define the *proportionate-allocation* user subset.

Definition 2.1. Let $\mathcal{G} \subseteq \mathrm{P}(\mathcal{U})$ be a set of user groups. A user subset $U \subseteq \mathcal{U}$ is a *proportionate allocation* of $\mathcal{G}$ if for every $g \in \mathcal{G}$, it holds that

$$\frac{|g \cap U|}{|U|} = \frac{|g|}{|\mathcal{U}|}$$

A user subset for which this definition holds faithfully represents the source population in the sense that it has a number of selected representatives from each group that is proportionate to their number in the population. This trait is used by surveyors in *stratified sampling* to guarantee that certain inferences from the survey are statistically sound (e.g., [8, 9]). For that, surveyors and domain experts carefully define a small set of non-overlapping population groups to be represented (in particular, $|U| \geq |\mathcal{G}|$). See further discussion on surveys in Section 9.

However, in this work we consider user repositories that often form a huge number of highly overlapping user groups, *making proportionate allocation infeasible.* A user subset of size $|U| << |\mathcal{G}|$ with every group even roughly proportionally represented is unlikely to exist. We therefore develop, in the following sections, solutions for a relaxed problem formulation, in particular, aiming to avoid under-representation of groups but allowing over-representation and prioritizing the coverage of certain groups over others.

Intrinsic vs. predicted. Intrinsic diversity is computed based only on known properties (e.g., [2, 4, 5, 7]), whereas *predicted* diversity utilizes a function predicting unknown values for each selected item (e.g., a probabilistic distribution of the answer to some question) [1, 3, 4, 6]. Thus, predicted diversity notions

typically optimize an expected target function (e.g., the expected number of different answers to be obtained).

In opinion procurement scenarios, the intrinsic approach, i.e., relying on user profiles rather than prediction of their opinions, is often preferable. First, the representation of different groups in the population may be the main client need, regardless of what their opinions are (e.g., having representatives for as many genders, age groups, nationalities, etc. as possible). Second, obtaining a reliable prediction of user opinions may be impractical – at least as hard as the original opinion procurement task. When this is the case, users with diverse profiles may provide a good alternative, since they are likely to provide relatively diverse opinions (as demonstrated in our experimental results in Section 7 and in [4]).

Diversification along a range of opinions. Diversification for opinion procurement is characterized by the need to diversify *along ranges of property values* – for example, one has to represent the full range of user opinions, from negative to positive; the full range of user activity or expertise levels, from low to high; users of all ages; etc. In contrast, diversification solutions that target the maximization of user skill or item relevance in diverse categories (as in, e.g., [1, 3]) are not applicable for capturing the full range of (skill/relevance) values in each category.

Support of high data dimensionality. In large-scale user repositories, each profile may consist of hundreds to thousands of properties (e.g., up to 2189 properties per user in the TripAdvisor dataset used in Section 8). Using such properties along with ranges of values associated with them (e.g., frequencies of some activity from lowest to highest), allows defining a huge number of meaningful population groups, larger by orders of magnitude from the number of selected representatives. A practical diversification solution should address this dimensionality problem either by significantly reducing the number of considered groups and/or by adopting a diversity notion and implementation that scale with the problem dimension.

Explanations and customization. Last, we have already noted (in the Introduction) that there is no one-size-fits-all solution for diversification and that different clients may have different diversification needs. To be able to fine-tune the diversification results, the clients must first be able to understand them - via some notion of *explanations* – and then have user-friendly *customization* mechanisms of modifying them according to their needs. The use of intricate optimization problems and/or interdependencies between selected items, which often makes sense the context of diversification, as well as the high scale and dimension make this desideratum nontrivial to achieve. Here, we address this challenge by adopting a simple diversification notion based on profile properties, which in turn are human-understandable, and then explanations and customization pertain to (modifying) how these properties are represented by the selected subset. (See Sections 5-6.)

In the following sections we describe our model and algorithmic solutions, achieving these desiderata.

3 MODEL

We next describe how user profiles are modeled in our framework. We then formally define the problem of diverse user selection with respect to this model.

Property	Alice	Bob	Carol	David	Eve
livesIn	Tokyo$^{(2)}$	NYC$^{(1)}$	Bali$^{(1)}$	Tokyo	Paris$^{(1)}$
ageGroup	50-64$^{(2)}$	–	50-64	–	–
avgRating Mexican	0.95$^{(3)}$	0.3$^{(1)}$	–	0.75	0.8
visitFreq Mexican	0.8$^{(1)}$	0.25$^{(1)}$	–	0.6$^{(2)}$	0.45
avgRating CheapEats	0.1$^{(1)}$	0.9$^{(1)}$	0.45$^{(2)}$	–	0.6
visitFreq CheapEats	0.6$^{(1)}$	0.85$^{(1)}$	0.2$^{(2)}$	–	0.3

Table 2: Example user profiles

3.1 User Profiles

Let $\mathcal{U}$ be a population of users and $\mathcal{P}$ be some domain of property labels. Following [10], we define the profile of a user $u \in \mathcal{U}$ as a tuple $D_u = \langle P_u, S_u \rangle$ where $P_u \subseteq \mathcal{P}$ includes all the properties known for u and $S_u : P_u \rightarrow [0, 1]$ maps each property to a score (normalized to $[0, 1]$). We use the notation $|p| = |\{u \in \mathcal{U} \mid p \in P_u\}|$, where $\mathcal{U}$ is assumed to be clear from the context. Property scores 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 3.1. Table 2 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 $[0, 1]$. Not every property is 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 may contain many properties – e.g., we have constructed from TripAdvisor[2] a user repository with up to 665 properties per user (Section 7). This is due to various activities of a user in the system (e.g., providing opinions about many destinations, each with many different features), due to various types of analysis performed over the data (e.g., one can compute the average rating, maximum rating…) and so on.

Using taxonomies to enrich profiles. To allow for an informed selection of users based on their profiles, these profiles should be as complete as possible. To this end, we perform a preprocessing step and apply *inference rules* on Boolean properties or on the raw data from which properties are derived. Such inference rules can be pre-specified as in RDF languages [11, 12] or derived via rule mining techniques [13]. A particularly useful type of inference rules is *generalization rules*, as exemplified next.

Example 3.2. The property `avgRating Mexican` in Table 2 is derived by averaging over the ratings given by each user to restaurants labelled as "Mexican Cuisine". On this raw data, we can apply a generalization rule if we know, e.g., by a cuisine taxonomy, that Mexican cuisine is a particular type of Latin cuisine. This will enable us to derive properties such as `avgRating Latin` for existing user profiles.

As another example, if `livesIn` is known to be a function, i.e., each person can only have one residence location in our repository, we can infer the falsehood of residence locations other

[2]TripAdvisor website: https://www.tripadvisor.com

than the one specified. Thus, by $S_{\text{Alice}}(\texttt{livesIn Tokyo}) = 1$ we can infer that $S_{\text{Alice}}(\texttt{livesIn X}) = 0$ for every $\texttt{X} \neq \texttt{Tokyo}$.

Having inferred all possible properties, we consider all other properties by the *open world assumption*: missing properties may be either false or true. For instance, if no frequency of visiting Mexican restaurants is known for Carol, this does not mean she has not been to such restaurants.

3.2 Weight-based Diversification

We next define a generic, weight-based approach to coverage-based diversification. We exemplify different choices of weights and show their usefulness for capturing user selection strategies.

Definition 3.3. A *diversification instance* is a tuple $(\mathcal{G}, \text{wei}, \text{cov})$ where $\mathcal{G} \subseteq 2^{\mathcal{U}}$ is a set of (possibly overlapping) user groups of interest, $\text{wei} : \mathcal{G} \to \mathbb{R}^+$ captures the weight of each group, and $\text{cov} : \mathcal{G} \to \mathbb{N}$ captures the number of users required so that a group is said to be covered.

Given a diversification instance and a selected user set $U \subseteq \mathcal{U}$, we define the score of U as $\text{score}_{\mathcal{G}}(U) = \sum_{G \in \mathcal{G}} \text{wei}(G) \cdot \min\{|U \cap G|, \text{cov}(G)\}$.

Finally, given a diversification instance and a budget $B \in \mathbb{N}$, we define BASE-DIVERSITY as the problem of finding a subset $U \subseteq \mathcal{U}$ such that $|U| \leq B$ and $\text{score}_{\mathcal{G}}(U)$ is maximized.

Note that if groups in $\mathcal{G}$ are overlapping, each user may contribute multiple group weights to the total score. This definition accounts for diverse subset selection in the sense that the score increases as more groups in $\mathcal{G}$ have (more) representatives in U. *Excessive* representation ($|U \cap G| > \text{cov}(G)$) is not rewarded but also not penalized.

The problem is defined in a generic way with the diversification instance given as input. We next discuss and exemplify the three parts of this instance.

Groups. Our diversification solution can support any set of groups input by the client, including manually crafted groups as typically defined by surveyors [8, 9].

To support large-scale, high-dimensional user repositories we develop here a concrete group definition that is efficiently computable for such repositories on the one hand, and effective in identifying meaningful groups for diversification on the other hand. Recall that user profiles comprise of properties from $\mathcal{P}$ with scores in $[0, 1]$.

Definition 3.4. Let $p \in \mathcal{P}$ be a property and $b \subseteq [0, 1]$ be a (continuous) range of scores. A *simple user group* is the subset of users whose score for p falls in b, formally,

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

For the ranges of scores, we split the range of scores of each property $p \in \mathcal{P}$ into a set of *non-overlapping buckets* $\beta(p)$. The rationale is, e.g., to group Mexican food lovers and dislikers separately. The computation of $\beta(p)$ is done by partitioning the 1-d data into intervals (clusters). There are several methods for 1-d interval splitting that are more effective than general clustering since the data is ordered (e.g., Jenks optimization [14], K-means, Expectation Maximization and by kernel density).

Simple user groups can be used to define more complex ones as the intersection or union of a few simple groups.

We note that the simplicity of our group definition is key for allowing explanations (see Section 5). There are more complex alternatives to splitting ranges into groups, such as multidimensional clustering (in our case, over multiple properties); however, these generally do not facilitate explainability. For instance, multidimensional clusters have no intuitive "label" or meaning.

Example 3.5. Reconsider Table 2. Let p be the property livesIn Tokyo and $b = [1, 1]$; then $G_{p,b} = \{\texttt{Alice}, \texttt{David}\}$ (group of "Tokyo residents"). Let p' be the property avgRating Mexican and $b' = (0.65, 1]$; then $G_{p',b'} = \{\text{Alice, David, Eve}\}$ (group of "Mexican food lovers"). One can also define, e.g., $G_{p,b} \cap G_{p',b'} = \{\texttt{Alice}, \texttt{David}\}$ ("Tokyo Residents who are also Mexican food lovers").

Our default definition of $\mathcal{G}$ consists only of simple groups, and we examine its effectiveness in Section 8. In particular, we empirically show that this approach also implicitly accounts for more complex groups in the population (such as 'Tokyo Residents who are also Mexican food lovers" from the example above).

Group functions. Similarly to group definition, the group weights (wei) and cover sizes (cov) functions can in principle be manually tailored for a specific domain and diversification context. As a more practical alternative, we next propose a few general-purpose choices, which can be fine-tuned by clients via our customization mechanism (see Section 6).

Definition 3.6. Weights are used to prioritize groups. The following are three examples of $\text{wei}(G)$:

- *Identical Group Importance (Iden):* $\text{wei}(G) \coloneqq 1$ (constant function).
- *Group Importance Linearly By Size (LBS):* $\text{wei}(G) \coloneqq |G|$.
- *Group Importance Enforced By Size (EBS):* define $\text{ord}(\cdot)$ as an ordering of the groups from smallest to largest, [3] then define $\text{wei}(G) \coloneqq (|U| + 1)^{\text{ord}(G)}$

Iden is the most "diverse" choice in the sense that it does not distinguish between groups, which by our problem definition will maximize the number of groups that are covered. However, in cases where only a small fraction of the groups can be represented/covered, one may choose to prioritize certain groups – e.g., large groups. Using LBS, the group importance is linear with its size, thus, e.g., the total weight of two groups of size X equals the weight of one group of size $2X$. This roughly corresponds to maximizing the number of groups represented *per user*. In EBS group importance by size is enforced, meaning that representing larger groups is always preferred over smaller ones. The latter requirement may apply to some diversification contexts, e.g., political surveys may aim to have at least one representative for each of the largest population groups.

Definition 3.7. The coverage function $\text{cov}(G)$ is used to guide how many users will be selected from each group. Examples include

- *Single Representative (Single):* $\text{cov}(G) \coloneqq 1$ (constant function).
- *Proportional Representation (Prop):* $\text{cov}(G) \coloneqq \max\{\lfloor |U| \cdot |G| / |\mathcal{U}| \rfloor, 1\}$ where $|U|$ is the size of the subset to be selected.

Here, Single is the most "diverse" definition in the sense that it requires only one representative from a group to consider it covered. In contrast, Prop rewards a representation that is proportional to the group size in the population.

We next exemplify the effect of using different functions on the resulting user choices.

Example 3.8. Reconsider the user profiles in Table 2 and assume that we define, for each property, three groups of users:

[3] Ties, i.e., groups of equal size, are broken arbitrarily.

those with scores in $[0.65, 1]$ ("high"), in $[0.4, 0.65)$ ("medium") and in $[0, 0.4)$ ("low"). The numbers in superscript at the table show the weights according to LBS – i.e., number of users – on the first user of each group. E.g., the only group with 3 users is `avgRating Mexican high`. The diverse user subset of size 2 that would be selected is {Alice, Eve} with total score 17. Single and Prop behave similarly here, and EBS would yield the same result with different scores. If instead we use Iden, then {Alice, Bob} will be selected with total score 11 (number of represented groups). This exemplifies the tendency of Iden to select more eccentric users, in this case Bob who is the only member of his groups, where LBS and EBS prioritize representatives of larger groups, in this case leading to a larger overlap (Alice and Eve are both Mexican food lovers).

Having defined our model, we next address the computational problem of BASE-DIVERSITY.

4 SOLVING BASE-DIVERSITY

We next consider the computation of a diverse subset of users according to Def. 3.3 of the BASE-DIVERSITY problem. We start by analyzing the complexity of the problem.

Unsurprisingly, we show that achieving an optimal solution is intractable in the subset size B unless P = NP, even for simple weight functions and even without customization. The decision problem DEC-DIVERSITY corresponding to BASE-DIVERSITY is that of the existence of a subset U with $|U| \leq B$ such that the sum of (customized) weights of covered groups exceeds a threshold T. We can then show:

Proposition 4.1. DEC-DIVERSITY *is* NP*-complete in* B.

Proof. Membership is immediate, since computing the total weight of a given user subset is in PTIME.

Hardness is proved by a reduction from Set Cover: Given a universe $\{1 \dots N\}$, a set of subsets $\{S_1, \dots, S_m\}$ and an integer k, we define $B = k$, $\mathcal{G} = \{G_1, \dots, G_N\}$ and $\mathcal{U} = \{u_1, \dots, u_m\}$, such that iff $i \in S_j$, then $u_j \in G_i$. Finally, we set $T = \sum_{G \in \mathcal{G}} \text{wei}(G) \cdot \min\{\text{cov}(G), B\}$, where $\text{wei}(G)$ can be any legal function and we set $\text{cov}(G)$ as the constant function 1 (Single, as we need only one set to cover each element). Since T is the maximum total score achievable, by covering every group in $\mathcal{G}$, it will be achieved by and only by a user subset that corresponds to a Set Cover. □

Approximate solution. The reduction from Set Cover implies not only the intractability of an exact solution but also of a constant-factor approximation in terms of the size of the *covering group*. To formalize this, given an instance of BASE-DIVERSITY and a threshold score T, let $\text{opt}(T)$ be the minimal size of a subset $U \subseteq \mathcal{U}$ whose score exceeds T. We then have, based on [15] inapproximability result for set cover:

Proposition 4.2. *Assuming* $P \neq NP$*, there is no PTIME algorithm for* BASE-DIVERSITY *that given a threshold score* T*, finds a user subset* U *of size* $(1-O(1)) \cdot ln(|\mathcal{G}|) \cdot \text{opt}(T)$ *with* $\text{score}_{\mathcal{G}}(U) \geq T$.

Fortunately, this does not exclude the possibility of approximation in the second axis, namely achieving a near-optimal score while conforming to the given budget. Indeed, a simple greedy algorithm achieves a constant approximation ratio in this sense.

Algorithm 1 outlines this greedy selection. Its input is a repository of users, a bound B on the number of users and a diversification instance (groups, weight function and coverage function). The algorithm starts by initializing an empty U (line 1) and computing, for each user the value $\text{marg}_{u,U}$, which stands for the potential marginal contribution of u to the total score if added to U (line 2). The algorithm then iteratively selects B users. Unless $\mathcal{U}$ is empty (line 4), the user maxUser with the greatest marginal contribution is selected (line 5) and moved from $\mathcal{U}$ to U (line 6). For each group G covered by maxUser, its required coverage $\text{cov}(G)$ decreases by 1 (line 8), and if no more representatives are required to cover G ($\text{cov}(G) = 0$) then G should have no effect on the selection of the following users. We thus, subtract $\text{wei}(G)$ from the marginal contribution of its other members (line 10). After B iterations (or earlier, if $|\mathcal{U}| < B$) the algorithm returns U.

Algorithm 1: Greedy User Selection

Input: $\mathcal{U}$, B, $\mathcal{G}$, wei, cov
Output: U (a set of $\leq B$ users)

1 $U \leftarrow \emptyset$;
2 **foreach** $u \in \mathcal{U}$ **do** $\text{marg}_{u,U} \leftarrow \sum_{G \in \mathcal{G} | u \in G} \text{wei}(G)$;
3 **for** $i \in 1..B$ **do**
4 **if** $\mathcal{U}$ *is empty* **then** break;
5 maxUser $\leftarrow \arg\max_{u \in \mathcal{U}} \text{marg}_{u,U}$;
6 $U \leftarrow U \cup \{\text{maxUser}\}$, $\mathcal{U} \leftarrow \mathcal{U} - \{\text{maxUser}\}$;
7 **foreach** *Group* G *such that* maxUser $\in G$ *and* $\text{cov}(G) > 0$ **do**
8 $\text{cov}(G) \leftarrow \text{cov}(G) - 1$;
9 **if** $\text{cov}(G) = 0$ **then**
10 **foreach** $u \in G$ **do** $\text{marg}_{u,U} \leftarrow \text{marg}_{u,U} - \text{wei}(G)$;
11 **return** U

Data Structures. For efficiency, we represent both the groups and the users as lists, each group $G \in \mathcal{G}$ with its current $\text{wei}(G)$ and $\text{cov}(G)$ values, and each user $u \in \mathcal{U}$ with $\text{marg}_{u,U}$. We further keep links in both directions between the lists, from groups to their members and vice versa. Whenever we (re)compute $\text{marg}_{u,U}$ we can remove the links from the user to groups with weight 0 or coverage size 0, which are not (or no longer) relevant for the user selection, to improve the performance of subsequent computations.

Example 4.3. We next exemplify the execution of Algorithm 1 for the user selection scenario in Example 3.8, using LBS and Single. After executing line 2 the marginal contributions of Alice, Bob, Carol, David and Eve, namely, the sum of weights of their properties, are 10, 5, 7, 6 and 10 respectively. Assume that at the first iteration of the external loop Alice is chosen and removed from $\mathcal{U}$ to U (ties are arbitrarily broken; in this example, selecting Eve happens to lead to the same output). Then the coverage of each of Alice's groups is set to 0. For each such update, the marginal contribution of other members of the groups is reduced: first, the contribution of David is reduced by 2 due to the `livesIn Tokyo` group; next, the contributions of David and Eve are reduced by 3 due to the `avgRating Mexican high` group; and so on. At the end of the first iteration, the contributions of Carol, David and Eve are updated to 5, 2 and 7 respectively. Thus, Eve is chosen at the next iteration, and {Alice, Eve} would be the output, which in this case is also the optimal solution.

Proposition 4.4. *Algorithm 1 computes a* $(1-1/e)$*-approximation of* BASE-DIVERSITY*, i.e. achieves a score that within a multiplicative factor of at least* $\geq (1-1/e)$ *of the optimal for the given budget, in time* $O(B \cdot \max_{G \in \mathcal{G}} |G| \cdot \max_{u \in \mathcal{U}} |\{G' \in \mathcal{G} \mid u \in G'\}|)$.

Proof. The complexity of Algorithm 1 is $O(B \cdot |\mathcal{U}| \cdot |\mathcal{G}|)$ due to the updates of the marginal user contributions (line 10). This

line is nested within three loops. The loop line 3 repeats $O(B)$ times, the loop at line 7 repeats $O(\max_{u \in \mathcal{U}} |\{G' \in \mathcal{G} \mid u \in G'\}|)$ times, namely, bounded by the maximal number of groups per user, and the innermost loop (line 10) repeats at most $O(\max_{G \in \mathcal{G}} |G|)$ times, namely, bounded by the size of the largest group. We assume constant complexity for arithmetic computations and for getting the next group of a given user/next user of a given group (as links in both directions are maintained).

As for the approximation ratio, observe that the score function satisfy the following properties *regardless of the choice of wei, cov*:

- *Submodularity.* For any $U \subseteq U' \subseteq \mathcal{U}$ and $u \in \mathcal{U}$ we have $\text{score}_{\mathcal{G}}(U \cup \{u\}) - \text{score}_{\mathcal{G}}(U) \geq \text{score}_{\mathcal{G}}(U' \cup \{u\}) - \text{score}_{\mathcal{G}}(U')$.
- *Non-negativity.* $\text{score}_{\mathcal{G}}(\cdot) > 0$ since $\text{wei}(G)$ and $\text{cov}(G)$ are positive.
- *Monotonicity.* If $U \subseteq U'$ then $\text{score}_{\mathcal{G}}(U) \leq \text{score}_{\mathcal{G}}(U')$.
- *Bounded input.* The size of a selected subset 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 $\text{score}_{\mathcal{G}}(U \cup \{u\})$ is known to guarantee the stated approximation ratio [16]. □

Clearly, $\max_{G \in \mathcal{G}} |G| = O(|\mathcal{U}|)$ and $\max_{u \in \mathcal{U}} |\{G' \in \mathcal{G} \mid u \in G'\}| = O(|\mathcal{G}|)$. If we use only simple groups, the complexity bound of Prop. 4.4 may be simply written as $O(B \cdot \max_{G \in \mathcal{G}} |G| \cdot \max_{u \in \mathcal{U}} |P_u|)$.

5 EXPLANATIONS

We have proposed a simple generic framework for diverse user selection. We next consider notions of *explanations* of the diversification results, allowing clients to understand why certain users were selected and how certain groups were covered. This, in turn will enable the clients to use customization (see the next section) to refine these results.

Recall first that we have defined user profiles based on support values with respect to properties. We will use the set of property names in $\mathcal{P}$ to define *labels*; in practice, this entails that we will keep them in a human-readable form, and their combination will be used in presented explanations.

We further introduce labeling to simple groups, as follows. Each bucket is given a label, e.g. "low scores", "medium scores" and "high scores". Then, the *label* $G_{p,b}$ of each group can be constructed from the property name p and the label corresponding to the bucket b, e.g., "high scores for Mexican cuisine (average rating)".

We then define the notion of explanation to be presented to the client alongside the computed user subsets. Such explanations may be practically shown to users by visual means (see Section 7).

Definition 5.1. We introduce three types of explanations.

- *Group explanations.* Let $G \in \mathcal{G}$ be a group labeled l_G, we define its explanation as $\text{expl}(g) = \langle l_G, \text{wei}(G), \text{cov}(G) \rangle$, namely the property and bucket that defines it, along with its weight and required coverage.
- *User explanation.* The explanation of a selected user $u \in U \subseteq \mathcal{U}$ is defined as $\text{expl}(u) = \{G \in \mathcal{G} \mid u \in G\}$, namely, the groups which u represents.
- *Subset-group explanation.* Let $U \subseteq \mathcal{U}$ and $G \in \mathcal{G}$ be a selected user subset and a group. The explanation of how U covers G is the pair $\langle \text{cov}(G), |U \cap G| \rangle$, which represents the required versus actual coverage.

These explanations are complementary in the sense that they provide intuition about different aspects of the diverse selection: respectively, of the group meaning and importance; of why a given user was selected; and on how the selected user subset, as a whole, covers a certain group.

Example 5.2. Reconsider the selection of {Alice, Eve} in Example 3.8 in our running example. Assume that each property is given a human readable label, and we are further given labels for the buckets of Boolean properties and properties with a score. Group explanations may then be ⟨"high average rating for Mexican Cuisine", 3, 1⟩, since the weight of this group reflects its size, 3, and we use Single – one user to cover each group. "High" is the label of the bucket in range (0, 65, 1]. Similarly, we may have ⟨"lives in Tokyo", 2, 1⟩, where the label of the bucket [1, 1] is empty for Boolean properties, and "lives in Tokyo" is the property label. Next, an explanation for Alice would be the groups she represents, "high average rating for Mexican Cuisine", "lives in Tokyo" and so on. The explanation for {Alice, Eve} with respect to the former group would be ⟨1, 2⟩, meaning both selected users belong to this group, exceeding the required coverage.

6 CUSTOMIZATION

Given the user selection results and their explanations, clients may fine-tune the algorithms if the results do not fit their needs. Specifically, we introduce customization at the level of individual groups (which, in a sense, correspond to the granularity of explanations that are shown). This customization is applied "on top" of the high-level decisions of how weights are assigned, which would typically not be made at the group level.

Definition 6.1. A *customization feedback* of the user is composed of four subsets of $\mathcal{G}$.

- $\mathcal{G}_+$: "must have" groups, each selected user must belong to all of them.
- $\mathcal{G}_-$: "must not" groups, each selected user must belong to none of them.
- $\mathcal{G}_d$: "priority coverage" groups, whose coverage is prioritized over others.
- $\mathcal{G}_{d?}$: "standard coverage" groups, whose coverage is of a lower priority with respect to the priority coverage groups.

Intuitively, the first two types of feedback serve to filter the repository of users. To avoid contradictions, if $\mathcal{G}_+$ contains more than one bucket of some property p, users need only belong to one of them. By default, $\mathcal{G}_+ = \mathcal{G}_- = \emptyset$. The priority and standard coverage group definitions allow to prioritize the coverage of certain groups, or completely ignore them in terms of coverage (groups in $\mathcal{G} - (\mathcal{G}_d \cup \mathcal{G}_{d?})$). By default, $\mathcal{G}_d = \emptyset$ and $\mathcal{G}_{d?} = \mathcal{G}$.

Example 6.2. Assume that for a particular application, the client prefers users from diverse locations and who are familiar with Mexican food. This may be captured by the following customization feedback:

- The "must have" groups consists of the three buckets of `AvgRating Mexican`, thereby requiring that the selected users have provided some rating for some Mexican restaurant.
- The "priority coverage" groups $\mathcal{G}_d$ consists of the multiple `livesIn <city>` properties.
- Finally, $\mathcal{G}_- = \emptyset$ and $\mathcal{G}_{d?} = \mathcal{G} - \mathcal{G}_d$.

We will demonstrate below how these choices guide user selection.

The effect of a customization feedback on the chosen groups is formalized as follows.

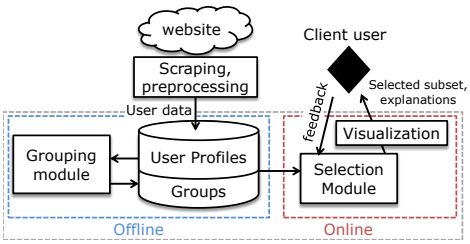


Figure 1: System Architecture

Definition 6.3. Given a customization feedback $\mathcal{G}_+, \mathcal{G}_-, \mathcal{G}_d$ and $\mathcal{G}_{d?}$, define the refined set of users as

$$\mathcal{U}' = \{u \in \mathcal{U} \mid \forall G_{p,b} \in \mathcal{G}_+,\ \exists b' \in \beta(p) :\ u \in G_{p,b'} \wedge G_{p,b'} \in \mathcal{G}_+\} \cap \{u \in \mathcal{U} \mid \forall G_{p,b} \in \mathcal{G}_- :\ u \notin G_{p,b}\}$$

The customized diversity problem CUSTOM-DIVERSITY is then to select new subset $U \subseteq \mathcal{U}'$, of size $\leq B$, that maximizes $\text{score}_{\mathcal{G}_d}(U)$, namely, the sum of weights over covered groups from $\mathcal{G}_d$, breaking ties by $\text{score}_{\mathcal{G}_{d?}}(U)$.

Example 6.4. Reconsider the problem of selecting a user subset of size 2 from Example 3.8. We now incorporate the customization feedback of Example 6.2. The refined user set will exclude Carol who did not rate Mexican food. The best user subsets using Single and LBS functions is still {Alice, Eve}: first, it maximizes the sum of weights over `livesIn <city>` properties (to 3). Among other subsets that achieve this maximum (e.g., {Alice, Bob}), the selected subset further maximizes the sum of weights over other properties (to 14). Note that a different customization feedback would yield a different result; e.g., if we set $\mathcal{G}_{d?} = \emptyset$ then any subset maximizing the sum of weights over `livesIn <city>` properties may be selected.

Results revisited. CUSTOM-DIVERSITY is NP-complete, as an easy consequence of the NP-completeness of BASE-DIVERSITY. Further, the counterpart of Proposition 4.4 holds:

Proposition 6.5. CUSTOM-DIVERSITY *may be approximated within a multiplicative factor of at least* $(1 - 1/e)$ *in time* $O(B \cdot \max_{G \in \mathcal{G}} |G| \cdot \max_{u \in \mathcal{U}'} |\{G' \in \mathcal{G} \mid u \in G'\}|)$

Proof. The approximation algorithm is an adaptation of Algorithm 1 to account for customization feedback, as follows.

We first change the weights of the total score function to simulate a primary order by "priority coverage" groups and secondary order by "standard coverage" groups. The $\widetilde{\text{score}}(U) = \text{score}_{\mathcal{G}_d}(U) \cdot \text{MAX-SCORE} + \text{score}_{\mathcal{G}_{d?}}(U)$ where MAX-SCORE is a value greater than the maximum value of $\text{score}_{\mathcal{G}_{d?}}(U)$.

It now holds that:

Lemma 6.6. *The* $\widetilde{\text{score}}(U)$ *function is submodular, non-negative and monotone.*

We further refine the user repository to be $\mathcal{U}'$ of Definition 6.3, by filtering out user profiles that do not satisfy the conditions.

Last, we change Algorithm 1 so that instead of greedily selecting from $\mathcal{U}$ based on $\text{score}_{\mathcal{G}}(U)$, it selects from $\mathcal{U}'$ based on $\widetilde{\text{score}}(U)$. Following Lemma 6.6, the refined algorithm satisfies the approximation guarantees.

□

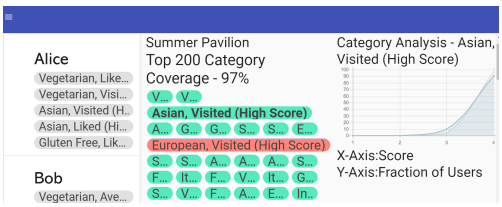


Figure 2: Screenshot of Podium UI: selection explanation

Explanations. The explanations defined in Section 5 can also be used for explaining customized results. The set of users and weights of groups may be different; in particular priority coverage groups will have a higher weight indicating a higher priority. Clients may not be able to interpret the values of weights, but they will be able to compare weights between groups to understand their relative importance.

7 IMPLEMENTATION

We developed Podium as a prototype system, implemented in Python using Flask[4]. Its architecture is depicted in Figure 1. The input to Podium is a set of user profiles, as explained in Section 3.1, in JSON format. 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 allows an administrator to feed in an *initial set of diversification configurations* with associated textual descriptions.

The Graphical User Interface of Podium was created using AngularJS 1.6.4[5]. Given a user selection 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 2 shows the explanation page for the initial configuration titled "Summer Pavilion", which only considers properties related to a restaurant in that name. The labels of the groups in this page are taken from the group explanations of Def. 5.1. The left pane displays the names of selected users, along with the top-weight groups that were covered by each (corresponding to user explanations of Def. 5.1). The middle pane uses the subset-group explanations of Def. 5.1 to show the percentage of top-weight relevant groups 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.[6] When clicking any group, the right pane displays a graph comparing the score distribution for the relevant property between the entire population and the selected subset (in Figure 2 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").

8 EXPERIMENTAL STUDY

We have examined the performance of our system, first, by evaluating the intrinsic diversity of the selected subset, i.e., how well it represents the source population (as explained in Section 2, proportional allocation is generally impossible in our setting). While an intrinsically diverse subset is sufficient in some user selection scenarios, in others one cares also for the eventual diversity of procured opinions. In order to examine this aspect, we

[4] Flask. http://flask.pocoo.org
[5] AngularJS. https://angularjs.org
[6] For space constraints, some group names in Figure 2 are truncated.

have selected datasets *with known ground truth*, i.e., where user opinions are already recorded. We have used these to simulate opinion procurement from the selected user subset and evaluate the diversity of collected opinions.

8.1 Datasets

The datasets used in our experimental study are real-world user repositories, focusing on the domain of restaurant reviews. The raw data is pre-processed to obtain aggregated scores for different categories based on user activity, as explained below.

The first dataset that we use consists of a sample of TripAdvisor [17] restaurant reviews data. This dataset contains data from 4475 users reviewing a total of 50K restaurants, and 11749 different groups. The raw data contains both user submitted data (e.g. age, residence) and user activity data (e.g. visited destinations), pre-processed and enriched as explained in Section 3.1, to generalize, e.g., Mexican cuisine to Latin cuisine.

The second dataset is the Yelp Open Dataset [18], which contains businesses, reviews, and user data for use in academic purposes. In our experiments we have used a subset of the data: for compatibility with the TripAdvisor dataset we used only restaurant-related data and took the 60K users with most reviews – reviewing a total of 52K restaurants and forming 8491 different groups. This limit was used in our *qualitative* experiments (see Section 8.4) due to memory limitations of some of the other baselines – recall that each user belongs to many groups. In comparison with the TripAdvisor dataset, the Yelp dataset has more users, but less groups due to its simpler semantics.

The datasets include two types of properties: ones that appeared explicitly in the original data, such as age and address; and ones that we have derived based on aggregation of user activities, as follows.

- *Average Rating*. The average rating given by a user to restaurants of a certain category (e.g. French cuisine), normalized by the overall average rating of that user.
- *Visit Frequency.* The fraction, among all the restaurants visited by a user, of restaurants from a certain category.
- *Enthusiasm Level.* A combination of rating and visit frequency, computed as the fraction of rating points given by the user to restaurants of a certain category.

8.2 Metrics

We next introduce metrics for algorithm performance, in three categories. *Intrinsic diversity metrics* are computed from the known properties of the selected user subset. *Opinion diversity metrics* are computed from opinions of the user subset, which are unknown to the user selection algorithms as explained in the beginning of this section. Finally, we evaluate the *scalability* of the algorithms.

Intrinsic diversity metrics. We consider a few complementary metrics, including our definition of total score – since our algorithm only approximates its optimal value – but also metrics of coverage that are not targeted directly by `Podium`.

- *Selection total Score.* According to Def. 3.3. We focus on the LBS weights and Single coverage functions, which our algorithm aims to approximate. This score can give us an intuition about alternative algorithms, since it reflects the number of groups and users within them that are represented by the subset.
- *Top-k groups coverage.* There are thousands of groups within the source population, which cannot be covered even by one representative in a small selected subset. We consider whether the top-k largest groups have selected representatives. In our experiments we have set $k = 200$.
- *Intersected-Property Coverage.* This metric is similar to the previous one, but now we consider intersections of simple groups that are at least as large as the k-th largest simple group.
- *Distribution Similarity.* This metric examines the similarity of user distribution between the source population and the selected subset, according to Def. 8.1 below.

The last metric aims at testing whether the number of representatives selected for groups is proportional to their number in the population, even if the coverage size is Single. Intuitively, our algorithm is likely to choose more representatives for larger groups without targeting it explicitly. However, standard distribution similarity metrics (such as Kolmogorov-Smirnov goodness of fit test) are not adequate for this purpose: to enhance coverage, small groups *must be over-represented.* We therefore define a distribution similarity metric that only taxes the selected user subset for under-representation of groups.

Definition 8.1. Let $B = b_1, \ldots, b_k$ be a discrete set of values. Let $f_{\text{subset}}, f_{\text{all}} : B \to [0, 1]$ be two functions over B, intuitively applied to the entire population and the selected subset respectively. We define the *coverage-oriented distribution similarity* (CD-sim, for short), as cd-sim$(f_{\text{subset}}, f_{\text{all}})$ =

$$1 - \frac{1}{k} \sum_{f_{\text{subset}}(b_i) < f_{\text{all}}(b_i)} \frac{(f_{\text{all}}(b_i) - f_{\text{subset}}(b_i))}{f_{\text{all}}(b_i)}$$

Note that this definition sums only over values of the domain for which the subset (f_{subset}) returns a lower result than the full population (f_{all}), corresponding to under-representation. Normalizing by the size of the full population guarantees that under-representations of larger groups are preferred, since the relative tax each missing user incurs is smaller.

For the group bucket distribution similarity, for a given property $p \in \mathcal{P}$, we set $B = \beta(p)$ (i.e., the set of buckets computed for p) and for $b \in \beta(p)$, we define $f_{\text{all}}(b) \mapsto \frac{\text{wei}(G_{p,b})}{\sum_{b' \in \beta(p)} \text{wei}(G_{p,b'})}$ (the fraction of the weight that falls in the b bucket, which corresponds to the fraction of the users that belongs to this group). Similarly, we define $f_{\text{subset}}(b) \mapsto \frac{\text{wei}(G_{p,b} \cap U)}{\sum_{b' \in \beta(p)} \text{wei}(G_{p,b'} \cap U)}$ for a selected subset $U \subseteq \mathcal{U}$. For the overall distribution score, we average CD-sim for the top-20 largest groups.

Example 8.2. An example user distribution for the property "Mexican Food Average Rating" could be [0.23,0.4,0.37], meaning 23% of the population rate Mexican food poorly, etc. A selection distribution of [0.4,0.5,0.1] would receive a CD-sim score of 0.76, reflecting a penalty solely for the under-representation of the third sub-group, and not for the over-representation of the others.

Diverse opinion metrics. Thus far, the diversity metrics we considered were defined over user profiles. We next introduce metrics that consider the diversity of procured opinions. For that, we split the data into profiles used for selection, and data that simulates the procured opinions. For instance, we can select users from TripAdvisor based on their profiles excluding the data related to some destination, then evaluate diversity of the selected subset reviews on the excluded destination.

To measure diversity of opinions we have used complementary metrics that relate to the rating provided by the selected subset

and their reviews' contents. Importantly, user opinions range not only over sentiment (positive or negative), but also over the facets that interest them with respect to the object in review.

- *Topic+Sentiment Coverage.* We measure content coverage using a list of prevalent topics extracted by TripAdvisor from each destination's reviews. We measure the fraction of topics that appear in the selected subset reviews. We also consider the review sentiment, such that 100% coverage means every topic appears in both a positive and a negative review.
- *Usefulness.* Available only for Yelp dataset, based on user feedback to reviews. A review is more useful when it is well-written, but also when a larger group of users agree or can relate to its contents. In this sense, the review is more likely to represent the opinions of large population groups, which is what we target in coverage-based diversity. We compute this metric by summing over individual reviews usefulness levels.
- *Rating Distribution Similarity.* Reusing our distribution similarity metric CD-sim, we measure the similarity in *rating distribution* between the selected subset and the entire population. For a given destination we set $B = \{1, \ldots, k\}$ (i.e., the set of possible rating values) and for $i \in \{1, \ldots, k\}$, let $R_i \subseteq \mathcal{U}$ be the set of users that gave this destination a rating of i. We define $f_{\text{all}}(i) \mapsto \frac{|R_i|}{\sum_{j=1}^{k}|R_j|}$. Similarly, for a selected subset $U \subseteq \mathcal{U}$ we define $f_{\text{subset}}(i) \mapsto \frac{|R_i \cap U|}{\sum_{j=1}^{k}|R_j \cap U|}$
- *Rating variance.* Variance of the rating given by the selected subset to a given destination.

All of the above metrics are defined per destination, to obtain an overall score we average over all destinations.

Scalability. We have tested the system execution times and scalability with respect to the number of users and profile size.

8.3 Baselines

We consider the following alternatives algorithms for diverse user selection.

- `Podium`. Our implementation as described above. By default, we use no customization feedback, LBS weights (Def. 3.6), the Single coverage function (Def. 3.7) and a budget $B = 8$, which also applies to the other baselines.
- *Random Selection.* An algorithm that selects a subset of the users uniformly at random. This method is a common practice in user selection for opinion procurement in the context of e.g. surveys, and under certain conditions there are reasons to assume the selected set of users is likely to be diverse. However, it has already been observed that explicitly managing diversity is often helpful in improving the results [4], which we will demonstrate in our setting.
- *Clustering.* Splitting the entire user repository into clusters, and choosing one representative from each – assuming each cluster represents a community. This approach has an inherent drawback as the clusters may have no intuitive explanation or customization; yet here we compare its performance to ours on other metrics. There are many options for clustering algorithms and representative choice. We have tested several options and show here one generally practical choice: computing B clusters using k-means (Scikit-Learn implementation[7]), then taking the near-mean user as the representative per cluster. k-means is particularly suitable to our settings: large, high-dimensional normally-distributed data, easy parametrization and is known to achieve comparatively high quality and low execution times (see, e.g., a comparison of clustering solutions in [19]).
- *Distance-based diversity.* While the distance-based approach for diversification has a different goal than coverage-based diversity (as explained in Section 2), it is still interesting to compare its performance to ours. As a representative distance-based baseline we use the S-Model of [4] via a greedy algorithm that maximizes the pairwise Jaccard distances between the properties of the selected subset.
- *Optimal Selection.* Naïve iteration over all user subsets of size B to obtain the optimal total score. This baseline is naturally applicable only for small values of B, and used to examine how good is the approximation achieved by our algorithm in practice, compared to the theoretical bound.

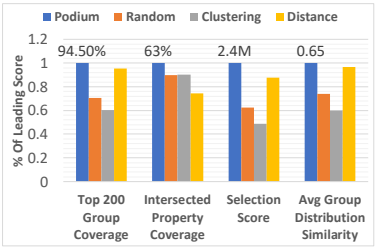
[7] Scikit-Learn. http://scikit-learn.org

8.4 Qualitative Results

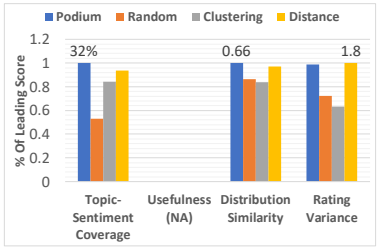
We next describe our experimental results regarding the achieved diversity. All experiments have been conducted on a Windows 10 machine powered by an Intel Core i7 7500U processor with a 16 GB of DDR4 memory.

Intrinsic diversity results. We depict the intrinsic diversity comparison between baselines for the TripAdvisor and Yelp datasets in Figures 3a and 3c, respectively. For showing different metrics on a similar scale, all scores are *normalized relative to the leading algorithm's score*; the value of the leading score is denoted on the relevant bar. Our main findings are summarized as follows.

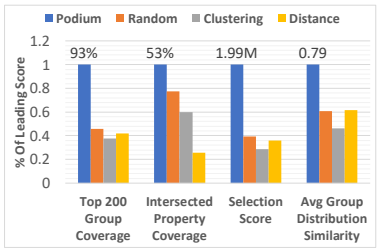
- `Podium` outperforms its alternatives in every tested diversity metric.
- Yelp is a more difficult dataset than TripAdvisor, since the former has less properties and less "room for maneuver"; for this dataset our results are better than the baselines by a significantly larger gap.
- Results for top-200 coverage and intersected property coverage indicate that our algorithm implicitly accounts for representing a high percentage of the largest groups, including complex ones – suggesting that selection based on simple groups may be sufficient for coverage purposes.
- The distance-based baseline performs poorly in covering complex groups, since it explicitly avoids intersections with overlapping properties between users.
- Surprisingly, our algorithm achieves a high similarity to the group distribution in the source population, although we do not optimize this directly.
- Our algorithm achieves the best total selection score by a large gap - this is expected, since our algorithm approximates the optimal value for this function.
- We were only able to test the optimal selection algorithm on a restricted source population and very small subset sizes due to the exponential runtime, hence it is omitted from the graphs. Generally, the total score achieved by `Podium` greatly exceeded the approximation bound and was near-optimal in all of our experiments. E.g., for selecting 5 out of 40 users `Podium` provided a .998 approximation ratio of the optimal.
- Since each user belongs to many groups, we can achieve high coverage even with a small B. As B increases, all the



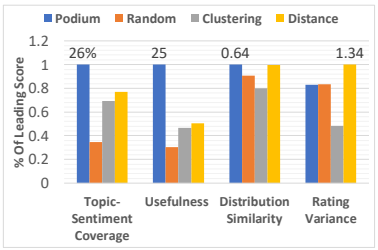
(a) TripAdvisor intrinsic diversity



(b) TripAdvisor opinion diversity



(c) Yelp intrinsic diversity



(d) Yelp opinion diversity

Figure 3: Quality experimental results

quality metric improve and the gaps between the baselines slightly decrease, but the general trends are preserved.

These results indicate that it is able to select good representatives of the sources population in different respects, covering most large groups and leaving few under-represented groups. Regarding the competitors, we observe that clustering is inferior in almost every metric; this indicates that the splitting of population into cluster is probably unable to identify meaningful groups, and is outperformed even by random sampling.

The results also indicate that distance-based selection is less able to represent groups not explicitly defined in the dataset. Generally, the main difference between the distance-based approach and ours is the pairwise intersection in user properties – e.g., 2 versus tens on average that we get for the Yelp dataset. Consequently, when there are a few prevalent categories that are shared by many users, the distance-based approach tends to seek the few users that do not have these categories, which comes at the expense of coverage and distribution similarity.

Opinion diversity results. We now consider whether indeed the selected user subset, by `Podium` and its alternatives, provides diverse opinions, according to the metrics defined in Section 8.2. Naturally, the considered groups in $\mathcal{G}$ may affect the opinion diversity for algorithms that rely on groups. In these experiments, we have chosen to consider groups that are defined from properties related to cuisine and location, as a client seeking opinions about a restaurant might have chosen.

For the TripAdvisor dataset (Figure 3b) we have examined 50 destinations with an average of 90 reviews per destination.

For the Yelp experiment (Figure 3d) we have considered 130 destinations with an average of 1730 reviews per destination.

Concluding both experiments, our main findings are:

- `Podium` achieves the best results in any tested metric for each dataset, with the exception of rating variance.
- Distance-based is the strongest competitor of `Podium` in this set of experiments; however, in the Yelp dataset we still see a significant gap w.r.t. `Podium` in topic coverage and usefulness.
- `Podium` achieves a good balance in the tradeoff between attaining dissimilar ratings/sentiments (as reflected in rating variance and distribution similarity) – which tends to the selection of "eccentric" users – and attaining representative opinions that cover prominent topics (as reflected in topic coverage, usefulness) – which tends to the selection of "mainstream" users.
- Random achieves a comparatively better performance in "dissimilarity" metrics (rating variance and distribution similarity), although still inferior to `Podium` and distance-based, and inferior results in "representativeness" metrics (topic-sentiment, usefulness), as expected.
- Clustering shows the opposite trends to those of Random, probably due to selection of near-mean users as representatives, which reduces the randomness of their selection but increases their representativeness.

These results reconfirm the assumption, proposed in previous work, that diverse users provide diverse opinions [4]. We have been able, by selecting a small user subset, to capture prominent topics and the ratings of the source population – even though `Podium` is not explicitly calibrated to predict opinions.

The effect of customization. We next consider the effect of customization on the selected user subset, with respect to the intrinsic quality metrics of the selected subset. We focus on the effect of "priority coverage" feedback from Def. 6.1. For that, we have selected from the Yelp dataset with 30K users, uniformly at random, four subsets $\mathcal{G}_{20} \subseteq \mathcal{G}_{40} \subseteq \mathcal{G}_{60} \subseteq \mathcal{G}_{80} \subseteq \mathcal{G}$ such that $|\mathcal{G}_i| = i$. Each subset was, in turn, fed into `Podium` as the set of priority coverage groups $\mathcal{G}_d$. Then, we have selected a user subset of size 8 in the customized setting. We have repeated this process 20 times and recorded the average for each metric.

The results are detailed in Figure 4, along with the intrinsic diversity metrics for the setting without customization, for comparison. Notably, all the quality metrics slightly decrease with

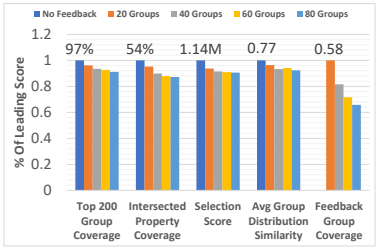


Figure 4: Yelp intrinsic diversity with customization

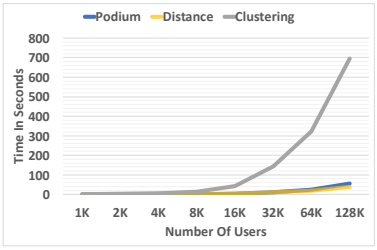


Figure 5: The effect of $|\mathcal{U}|$ on execution time.

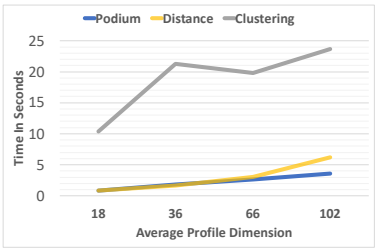


Figure 6: The effect of profile size on execution time.

every increase of the subset size, indicating that covering the priority groups restricts Podium's ability to cover standard priority groups – surprisingly, not by a significant gap. The newly-added *Feedback Group Coverage* metric measures the percentage of priority groups that were covered. Note that the groups are randomly selected with equal probability and are thus likely to be small and non-overlapping. Hence, there may not be 8 users who cover all of them. As expected, we can observe that the more priority groups are defined their coverage significantly decreases.

8.5 Scalability Results

We have examined the scalability of our algorithm w.r.t. the number of users and size of user profiles, which affect the number of groups. Here we only compare results with the clustering and distance-based baselines (random is immediate).

Scalability in number of users. In these experiments we have used user profiles with up to 200 properties. Following the complexity analysis in Section 4 we expect to witness a linear growth in the running time of the algorithm with accordance to the change in population size.

Scalability in profile size. The number of users has been set at 8K, and we varied the properties assembling the user profiles thus affecting their size. Again, we expect the running time to be linear to the average profile size.

Figures 5 and 6 depict the running times achieved by the algorithms. Our main findings are:

- Podium and distance-basedare ~9 times faster than the clustering alternative.
- Execution time for Podium scales linearly in the size of the population as well as the number of properties.
- The Optimal baseline, due to its exponential complexity, demonstrated poor scalability. E.g., for $|\mathcal{U}| = 40$ and B=5 its execution time was 443 seconds, and for $|\mathcal{U}| = 100$ we have terminated its execution after an hour. It is therefore omitted from the graphs.

9 RELATED WORK

A comparison between diversification approaches is given in Table 1. We now elaborate more on these solutions and others.

Diversity in crowdsourcing. A few studies (e.g., [2–4]) have considered the selection of diverse users in the context of crowdsourcing, namely performing tasks with the collaborative help of Web users/workers. The work of [4] is the most relevant to ours since it also studies diverse opinion procurement. They present two approaches for diversification: S-Model is distance-based, where pairwise distance is assumed to be known; and T-Model is coverage-based on predicted data, i.e., targets the selection of a user subset with a certain opinion distribution, but only in a single category. Other studies consider the selection of diverse crowd workers in order to improve the overall accuracy. In [3] the authors study the selection of diverse users by modeling the dependence of error rates within access paths (corresponding to non-overlapping user groups), and optimizing the information gain by the selected subset. This, however, does not apply to opinion procurement where there are no errors and every opinion should be accounted for. The recent [2] resembles ours in considering coverage-based diversity and supporting customization. However, they consider only a single group per worker.

Diverse search results. Search results diversification has been extensively studied in the field of information retrieval (e.g., [1, 6, 20, 21]). Apart from solving query ambiguity, diversification is used to avoid over-personalization of search results [22]. The classification of diversity definitions as coverage-based versus distance-based is also considered in this context [23, 24]. In contrast with our approach, IR solutions generally target relevance and therefore are inadequate for diversifying along different axes and accounting for positive and negative opinions.

Diversity in recommender systems. Diversification has also been studied in the context of recommender systems. Diversity can be computed based on item properties [6] or collaborative filtering, namely, the ratings of similar users to similar items [5, 7]. Specifically, in [7] a notion of explanation-based diversity is presented, but is different than ours – certain item properties are identified as recommendation-relevant and these are used for diversification. In contrast, we do not assume that relevant properties are predefined but rather derive explanations from the actual diversification results. Moreover, to our knowledge, coverage-based approaches have not been considered in the context of recommender systems.

User sampling in survey research. The selection of people representing some population has been vastly studied in the context of surveys. While also concerned with opinion procurement, the focus of this research field is different. Specifically, as explained

in Section 2, the dimensionality of user profiles in surveys is typically, by design, much lower than ours. This is because the goal of surveys is to ensure the statistical soundness of specific inferences from the participants' answers to larger populations [8, 9]. Statistical soundness may require the selected participants to be *proportionally allocated* (Def. 2.1), which, as explained in Section 2, is impossible in our high-dimensional setting due to the presence of many overlapping groups. Our approach involves a different problem formulation suitable for the high-dimensional setting. Also in contrast to surveys, which require a careful design and thereby a heavy load of manual curation, our solution applies to a given user repository as-is and may be easily executed multiple times, e.g., to incorporate data updates.

User selection. Various studies have considered the selection or filtering of users who undertake a task in crowdsourcing platforms or social networks. This includes assessment of crowd worker skill and filtering of low-skill workers [25]; filtering of low trust or spammer users [26]; filtering of slow or inefficient users [27]; expert finding [28–30]; and general-purpose declarative crowd selection [10, 31–33]. In general, these works are orthogonal to ours, since we can view the scores they derive as additional user properties that can be used for diversification.

A particular line of work considers *team formation* (or group formation), namely the selection of a set of workers that in some sense function as a team, by having e.g. complementary skills, similar properties, and/or better collaboration means [2, 34–37]. Among these, [2] is the most relevant to ours in targeting worker diversification, as discussed above. [37] uses coverage and diversity notions that our quite different than ours and thus render the problem and solution techniques quite different: diversity is considered between formed groups and is distance-based; and coverage is considered with respect to items rather than groups and does not support dimensionality.

10 CONCLUSION AND FUTURE WORK

In this work, we presented a framework for the selection of diverse user subsets for opinion procurement. We define a generic diversity notion that, while simple, satisfies a unique combination of desiderata that arise in presence of high-dimensional user profiles. In particular, as we showed, this notion admits efficient near-optimal computation and allows explanations and customization by the client. Our experimental study, on real user data, examines different metrics for diverse selection and shows that our algorithm outperforms a variety of baselines.

In future work, we plan to investigate further enhancement of the usability of our system, by methods of proposing relevant refinements for the user and by additional visualizations of the selection results. Another direction involves foundational study of the statistical properties of our algorithm: we have empirically shown that it performs well with respect to various measures other than our total score, e.g., distribution similarity and coverage of complex groups; the next step is formulating the guarantees for the algorithm performance in these metrics. The framework we have proposed is deterministic in choosing the (near-)optimal user subset by our definition, and is shown to outperform a fully random algorithm. Our implementation adds some randomness in randomly breaking ties, and we plan to further incorporation of randomness in our solution, e.g., adding noise to group weights, and its effect on the output diversity.

ACKNOWLEDGEMENTS

This work was funded 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] S. Cohen and M. Yashinski, "Crowdsourcing with diverse groups of users," in *WebDB*, 2017.
[3] B. Nushi, A. Singla, A. Gruenheid, E. Zamanian, A. Krause, and D. Kossmann, "Crowd access path optimization: Diversity matters," in *HCOMP*, 2015.
[4] 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*, vol. 8, no. 5, 2015.
[5] R. Boim, T. Milo, and S. Novgorodov, "Diversification and refinement in collaborative filtering recommender," in *CIKM*, 2011.
[6] M. Servajean, E. Pacitti, S. Amer-Yahia, and P. Neveu, "Profile diversity in search and recommendation," in *WWW*, 2013.
[7] C. Yu, L. V. S. Lakshmanan, and S. Amer-Yahia, "It takes variety to make a world: diversification in recommender systems," in *EDBT*, 2009.
[8] J. C. Helton and F. J. Davis, "Latin hypercube sampling and the propagation of uncertainty in analyses of complex systems," *Rel. Eng. & Sys. Safety*, vol. 81, no. 1, 2003.
[9] P. H. Rossi, J. D. Wright, and A. B. Anderson, *Handbook of survey research*. Academic Press, 2013.
[10] Y. Amsterdamer, T. Milo, A. Somech, and B. Youngmann, "Declarative user selection with soft constraints," in *CIKM*, 2019, to appear.
[11] G. Klyne, J. J. Carroll, and B. McBride, "Resource description framework (RDF): Concepts and abstract syntax," *W3C rec.*, vol. 10, 2004.
[12] D. L. McGuinness, F. Van Harmelen *et al.*, "OWL web ontology language overview," *W3C rec.*, vol. 10, p. 10, 2004.
[13] L. Galárraga, C. Teflioudi, K. Hose, and F. M. Suchanek, "Fast rule mining in ontological knowledge bases with AMIE+," *VLDB J.*, vol. 24, no. 6, 2015.
[14] G. F. Jenks, "The data model concept in statistical mapping," *International yearbook of cartography*, vol. 7, 1967.
[15] I. Dinur and D. Steurer, "Analytical approach to parallel repetition," in *STOC*, 2014.
[16] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, "An analysis of approximations for maximizing submodular set functions—I," *Math. Prog.*, vol. 14, no. 1, 1978.
[17] "Tripadvisor website," 2018, https://tripadvisor.com/.
[18] "Yelp open dataset," 2018, https://yelp.com/dataset/.
[19] M. Z. Rodriguez, C. H. Comin, D. Casanova, O. M. Bruno, D. R. Amancio, F. A. Rodrigues, and L. da F. Costa, "Clustering algorithms: A comparative approach," *CoRR*, vol. abs/1612.08388, 2016.
[20] J. Carbonell and J. Goldstein, "The use of MMR, diversity-based reranking for reordering documents and producing summaries," in *SIGIR*, 1998.
[21] K. D. Onal, I. S. Altingovde, and P. Karagoz, "Utilizing word embeddings for result diversification in tweet search," in *AIRS*, 2015.
[22] F. Radlinski and S. T. Dumais, "Improving personalized web search using result diversification," in *SIGIR*, 2006.
[23] M. Drosou and E. Pitoura, "Search result diversification," *SIGMOD*, 2010.
[24] W. Zheng, X. Wang, H. Fang, and H. Cheng, "Coverage-based search result diversification," *Inf. Retr.*, vol. 15, no. 5, 2012.
[25] P. G. Ipeirotis, F. Provost, and J. Wang, "Quality management on amazon mechanical turk," in *HCOMP*, 2010.
[26] V. C. Raykar and S. Yu, "Eliminating spammers and ranking annotators for crowdsourced labeling tasks," *JMLR*, vol. 13, 2012.
[27] D. Haas, J. Wang, E. Wu, and M. J. Franklin, "Clamshell: Speeding up crowds for low-latency data labeling," *PVLDB*, vol. 9, no. 4, 2015.
[28] A. Bozzon, M. Brambilla, S. Ceri, M. Silvestri, and G. Vesci, "Choosing the right crowd: expert finding in social networks," in *EDBT*, 2013.
[29] J. Tang, J. Zhang, R. Jin, Z. Yang, K. Cai, L. Zhang, and Z. Su, "Topic level expertise search over heterogeneous networks," *Machine Learning*, 2011.
[30] J. Zhang, J. Tang, and J. Li, "Expert finding in a social network," in *DASFAA*, 2007.
[31] Y. Amsterdamer, T. Milo, A. Somech, and B. Youngmann, "December: A declarative tool for crowd member selection," *PVLDB*, vol. 9, no. 13, 2016.
[32] M. Martın, C. Gutierrez, and P. Wood, "SNQL: A social networks query and transformation language," in *AMW*, 2011.
[33] R. Ronen and O. Shmueli, "SoQL: A language for querying and creating data in social networks," in *ICDE*, 2009.
[34] M. Kargar, A. An, and M. Zihayat, "Efficient bi-objective team formation in social networks," in *PKDD*, 2012.
[35] T. Lappas, K. Liu, and E. Terzi, "Finding a team of experts in social networks," in *SIGKDD*, 2009.
[36] H. Rahman, S. B. Roy, S. Thirumuruganathan, S. Amer-Yahia, and G. Das, "Optimized group formation for solving collaborative tasks," *VLDB J.*, vol. 28, no. 1, 2019.
[37] B. Omidvar-Tehrani, S. Amer-Yahia, P. Dutot, and D. Trystram, "Multi-objective group discovery on the social web," in *PKDD*, 2016.