

# Algorithmic Cartography: Placing Points of Interest and Ads on Maps

Mohammad Mahdian   Okke Schrijvers   Sergei Vassilvitskii  
Google   Stanford University   Google  
{mahdian, sergeiv}@google.com   okkes@cs.stanford.edu

## ABSTRACT

We study the problem of selecting a set of points of interest (POIs) to show on a map. We begin with a formal model of the setting, noting that the utility of a POI may be discounted by (i) the presence of competing businesses nearby as well as (ii) its position in the set of establishments ordered by distance from the user. We present simple, approximately optimal selection algorithms, coupled with incentive compatible pricing schemes in case of advertiser supplied points of interest. Finally, we evaluate our algorithms on real data sets and show that they outperform simple baselines.

## 1. INTRODUCTION

Accurate maps used to be treated as state secrets, but high quality geographic data is now readily available at our fingertips, and online mapping services boast a large user base, both on mobile and desktop. Google Maps alone claimed 150 million mobile users in 2011 [20] and more than one billion monthly active users in total in 2012 [19].

One of the challenges in algorithmic cartography is sifting through all of the data available and deciding what to place on the map (at a given zoom level). One must decide which natural features, like rivers, lakes, and mountains should be labeled, which roads should be rendered, which political features, such as city and neighborhood names, should appear, and, finally, which points of interest (POIs), such as restaurants, schools, and museums, should be highlighted.

In addition to the organic results, one may wish to place advertisements on the map as well. Both the quantity and the quality of page views on online map services make them an attractive target for advertising, since an online map user reveals the location she is interested in, and sometimes the type of business she is looking for. These are valuable pieces of information that can be used to target relevant advertisements. In short, many of the same dynamics that make sponsored search advertising a lucrative business are at play here as well. Furthermore, advertising on maps enables local and small business advertisers that might not be able to

compete on platforms like search advertising where larger advertisers are present [6].

In this work we focus on the POI selection problem, taking into account both organic and sponsored POIs. We must balance many competing objectives: the overall utility of the map, density and diversity of results, as well as the overall aesthetics. We will discuss the critical parameters that need to be taken into account for this problem, propose models that capture some of these issues, give optimization algorithms, and, for the case of ads, pricing mechanisms for these models. An additional consideration that we do not consider here is that of zoom stability, see, for example, [18] for complex optimizations that emerge with this constraint.

We distinguish between two distinct and important scenarios in our models: the location-aware case, where the location of the user is known (e.g., when the map is used on a mobile app with access to GPS data), and the location-unaware case, where only the boundaries of the map viewport are known and the POIs need to be placed without reference to a specific user location (e.g., when the map is used on a desktop computer with no GPS). We define these models in the next two sections, and then study the location-unaware problem followed by the location-aware one.

## 2. CONSIDERATIONS FOR POI SELECTION ON MAPS

Map making is an ancient tradition: one of the oldest known maps comes from Babylon, and is dated to 5th century BC [21]; in addition, there is some indication that maps may be even older [15]. Although maps today look nothing like the ornate hand-drawn maps from the Middle Ages, they are also not purely utilitarian and design continues to play a large role in annual map making competitions [7]. Similar design, aesthetic, and pragmatism considerations are important factors for the placement of POIs on maps, as we will discuss in this section.

**Aesthetics.** One desire that is often repeated is that of being selective in choosing items that should appear on the map, thereby avoiding clutter. In many cases, *less is more*: for example, one need not label every business in a commercial area, or every museum in a city. By discriminating, the map maker accentuates the relevant parts of the area, and gives the viewer the right information to make his or her decision.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the Owner/Author(s). Copyright is held by the owner/author(s).

KDD'15, August 10-13, 2015, Sydney, NSW, Australia.

ACM 978-1-4503-3664-2/15/08.

DOI: <http://dx.doi.org/10.1145/2783258.2783375>.

Capturing the overall aesthetics formally is a challenging task well beyond the scope of this work.<sup>1</sup> We begin by formalizing what is probably the most obvious form of clutter: placing POIs too close to each other, versus optimizing for geographic diversity. As it turns out, this is well aligned with another constraint motivated by negative externalities that POIs and advertisements impose on each other.

**Negative Externalities.** It is well understood that content, as well as ads, delivered simultaneously to a user compete for the user’s attention. Therefore, placing an additional POI on a page decreases the value of other POIs on the same page.<sup>2</sup> In economics terminology, POIs impose *negative externalities* on each other. This phenomenon is well studied in the literature for enforcing diversity among search results [2, 10], as well as among various forms of advertising [1, 8, 9, 11, 12, 13].

In the specific context of advertising, externalities are probably stronger, since a search for restaurants in downtown Palo Alto is a strong indication that the user is looking to pick *one* restaurant in that neighborhood to visit in a not too distant future, and the value of ads is mostly based on their immediate influence on this one decision. Furthermore, the externality effect for ads on a map is clearly a location-dependent effect: both the distance between potentially competing businesses, as well as the distance from the user play a large effect in valuation of the ad.

### 3. MODEL

We model negative externalities differently depending on whether the user’s location is known or not. In the *location-unaware* model, we discount the utility gained from a POI by a factor dependent on the distance to the nearest establishment (of a similar type) also placed on the map. Such an establishment is in direct competition, and the closer it is, the likelier that both are in the consideration set of the user. As mentioned above, this is also well-aligned with our aesthetic preference to avoid clutter in the map.

When the precise user location is known, which we will refer to as the *location-aware* setting, previous work on geographic choice [14, 17] has shown the rank of the establishment in a list ordered by distance to the user plays a large role on the user’s final choice. That is, all else being equal, the fact that one business is closest and the other is second closest, leads to the former getting a larger number of engagements than the latter. We adapt the empirical findings to define an economically-grounded model for negative externalities in this setting.

In both models, we are given a number of candidate POIs,  $A = \{a_1, a_2, \dots, a_n\}$ , each associated with the location where it should be shown, if selected. Throughout this paper we will assume that candidate POIs correspond to a single point on the map. This assumption is reasonable when they are relatively small markers. We will use this assumption in our algorithms, but the model is well-defined if each POI is allowed to be a geometric shape (e.g., a rectangle) that occupies a non-trivial area.

Each POI  $a_i$  has a standalone value  $v_i$ . For organic results this is the predicted utility of the POI to the user.

<sup>1</sup>See Section 8 for a discussion of various aspects of aesthetics not covered here.

<sup>2</sup>Other models, where different POIs might complement each other, are not considered in the present work.

For ads, this represents the value of the ad to the advertiser, if this were the only ad being shown. This value takes into account all of the features associated with the user session, such as the POI’s relevance, quality, and the distance between the location of the POI and the “focal area” of the map (e.g., points closer to the user’s location in the location-aware model have higher value, or when the user has asked for a route between two points on the map, points that are closer to this route have higher value, everything else being the same).

The final value of placing a POI  $a_i$  is the product of  $v_i$  and a discount term  $\delta_i(S)$  that depends on the set  $S$  of other POIs that are displayed and captures the effect of negative externalities. This part is modeled differently depending on whether the location of the user is known or not, as described below. In either case, the final objective of the optimization problem is to select a set  $S$  of POIs with the maximum total value.

We now state the problem formally:

**DEFINITION 1 (POI SELECTION PROBLEM).** *Given a set of POIs  $A = \{a_1, a_2, \dots, a_n\}$ , where each POI  $a_i$  has an associated value  $v_i$ ; and a discount function  $\delta_i$ , find a set  $S \subseteq A$  that maximizes the total utility:*

$$\sum_{a_i \in S} v_i \delta_i(S).$$

#### 3.1 Location-unaware model

In this model, we focus on a discount function defined as the minimum of pairwise discount factors between  $a_i$  and other POIs in  $S$ , i.e.,  $\delta_i(S) = \min_{a_j \in S \setminus \{a_i\}} \{\delta_{ij}\}$  where for  $a_j \in S \setminus \{a_i\}$ ,  $\delta_{ij}$  is the pairwise discount factor between  $a_i$  and  $a_j$ .<sup>3</sup>

The pairwise discount factors are, in general, functions of the similarity of the two POIs as well as the distance between their corresponding locations, i.e.,

$$\delta_{ij} = f(w_{ij}, d_{ij}),$$

where  $w_{ij}$  is the relative similarity of POIs  $a_i$  and  $a_j$  (e.g., it has a high value if both  $a_i$  and  $a_j$  are sushi places, moderately high value if one is a sushi place and the other is a burger joint, and low value if one is a sushi place and the other is a car mechanic), and  $d_{ij}$  is the distance between the locations associated with  $a_i$  and  $a_j$ . The discount function,  $f(w, d)$ , is a decreasing function of  $w$  and an increasing function of  $d$ , and is always between 0 and 1. In this work, we first focus on the uniform-relevance case, where  $w_{ij}$ ’s are all the same and  $\delta_{ij} = f(d_{ij})$  for an increasing function  $f$ . This is a reasonable assumption in cases where the candidates considered are similar, for example, when we want to only display restaurant markers. We will then turn to the more complex case of general similarity functions.

**Well-behaved discount functions.** Not all discount functions are appropriate for modeling the map POI selection problem: as we show below, under some functions the optimal solution may get arbitrarily dense.

<sup>3</sup>We also considered a number of alternative discount functions, e.g.,  $\prod_j \delta_{ij}$ , but settled on the minimum function because of its simplicity as well as a number of desirable properties it exhibits. For example, the minimum function is local, and therefore is not affected by factors such as the screen size that affect the total number of POIs that can be displayed.

Consider an instance of the POI selection problem with infinitely many candidates, one at each position in the unit square  $[0, 1]^2$ , and each having a value of 1. We call the discount function  $f$  a *bounded-solution* function, if the POI selection problem on this instance, using  $f$  as the discount function, has an optimal solution of bounded value.

For example, for  $f(x) = 1 - e^{-x}$ , the solution that consists of  $(k+1)^2$  points  $\{i/k : 0 \leq i \leq k\}^2$  has a value of

$$(k+1)^2 \cdot (1 - e^{-1/k}) \geq (k+1)^2 \cdot \frac{1}{2k} > \frac{k}{2},$$

which is unbounded. Such functions are obviously undesirable for modeling the POI selection problem.

To exclude these functions, we introduce the notion of *planar-density* of  $f$ , denoted by  $\lambda(f)$ , and show that bounded planar density implies that  $f$  is bounded solution.

LEMMA 1. *Call*

$$\lambda(f) = \sup_{x>0} \frac{f(x)}{x^2},$$

the planar-density of  $f$ . For every bounded-solution function  $f$ , the planar-density,  $\lambda(f)$ , is finite.

PROOF. Assume this is not the case. This means that for every constant  $B$ , there is a value of  $x$  such that  $\frac{f(x)}{x^2} > B$ . Using this, we construct a solution for an instance in  $[0, 1]^2$  as follows: the solution selects POIs at locations  $\{ix : 0 \leq i < 1/x\}^2$ . This is a collection of precisely  $\lceil 1/x \rceil^2$  points, with the distance between each point and its nearest neighbor exactly  $x$ . Therefore, the value of this solution is  $\lceil 1/x \rceil^2 f(x) \geq \frac{f(x)}{x^2} > B$ . Since  $B$  can be arbitrarily large, this contradicts the assumption that  $f$  is bounded-solution.  $\square$

### 3.2 Location-aware model.

To model the discounts in the location-aware setting, we adapt the findings in [14, 17] which showed, that, given a set of choices to a user, the probability that a user chooses an establishment can be explained by the rank of the establishment among the alternatives (controlling for other factors that depend on the individual establishment and not the set of alternatives, like the quality of the establishment or its distance to the user). For example, the attractiveness of a cafe decreases if there is a different cafe that is strictly closer. To formalize the model, let  $\pi$  denote a permutation of all candidates in increasing order of their distance to the user. In this model, we assume the location of the user is known, either deterministically, or as a distribution. If the location of the user is known as a distribution,  $\pi$  will be a random variable selected from a distribution  $P$  over the set of permutations. Let  $\pi(i, S)$  denote the rank of  $a_i$  in  $S$  under  $\pi$ , i.e.,  $\pi(i, S) = 1$  if  $a_i$  is the element of  $S$  that is closest to the user. The rank discounts are specified by a non-increasing sequence  $\gamma_1, \gamma_2, \dots$  of discount values. The discount of POI  $a_i$  in the set of displayed ads  $S$  is given by  $\delta_i(S) = \gamma_{\pi(i, S)}$ . We assume without loss of generality that  $\gamma_1 = 1$ .

In the case that the user location and therefore the ordering  $\pi$  is uncertain, we can represent the location information as a distribution over a set of permutations  $\Pi = \{\pi_1, \pi_2, \dots, \pi_k\}$ , with permutation  $\pi_i$  occurring  $p_i$  fraction of the time. It is easy to show that the total number of possible orderings of a set of  $n$  points in  $\mathbb{R}^2$  in increasing order

of their distance to a reference point is bounded by  $n^4$ , and therefore a compact representation of the input in the above form always exists. The aim in the non-deterministic case is to select a set  $S$  that maximizes the expected (over  $\pi$  drawn from  $P$ ) value of the objective function.

## 4. LOCATION-UNAWARE SELECTION

### 4.1 The Optimization Problem

Below we give an algorithm for the optimization problem for location-unaware POI selection. At a high level we show that a large family of discount functions can be approximated by a parametrized step function that jumps from 0 (full discount) to 1 (no discount) at a specific distance  $R$ . Using this simpler discount function, we compute an allocation that approximately maximizes the social welfare (for a judicious choice of  $R$ ), but is much easier to optimize. In fact, with this choice of discount function, the maximization problem is equivalent to finding a maximum weight independent set in unit disk graphs.<sup>4</sup> This means that the objective is to select a set of points with the maximum total weight subject to the constraint that no two selected points are within distance  $R$  of each other. A simple approximation algorithm for this problem is the greedy algorithm that iteratively picks the point of maximum weight and eliminates all of the candidates within distance  $R$  of the selected point. We call the algorithm PICKANDREMOVE and show its pseudocode in Algorithm 1.

---

#### Algorithm 1 Map POI Selection Algorithm

---

```

1: function PICKANDREMOVE( $R$ )
2:    $C \leftarrow$  Set of candidate POIs,  $A$ 
3:    $S \leftarrow \emptyset$ 
4:   while  $C \neq \emptyset$  do
5:     Let  $a_i \leftarrow \arg \max_{a_i \in C} v_i$ 
6:      $S \leftarrow S \cup \{a_i\}$ 
7:      $C \leftarrow C \setminus \{a_j : d_{ij} < R\}$ 
8:   end while
9:   return set  $S$  of
10: end function

```

---

To analyze the algorithm, we define the following function for a set  $S$  of points on the plane:

$$h(S) = \sum_{a_i \in S} f\left(\min_{a_j \in S \setminus \{a_i\}} \{d_{ij}\}\right).$$

Here, if  $S \setminus \{a_i\}$  is empty,  $f(\min_{a_j \in S \setminus \{a_i\}} \{d_{ij}\})$  is defined as 1. The core of our proof is the next lemma, which gives an upper bound on the value of  $h(S)$  for any set of points contained in a ball of given radius. To state the lemma, we need to restrict  $f$  to a class of “well-behaved” functions as defined below.

DEFINITION 2. *We say the function  $f$  is well-behaved if:*

- $f$  is an increasing function with  $f(0) = 0$  and  $f(\infty) = 1$ .

<sup>4</sup>This also places the problem in the realm of geometric set packing, e.g. [4]. We express the approximation with respect to the *original* discount function differentiating it from traditional set packing problems.

- $f$  has a bounded planar density,  $\lambda(f)$ .
- There is a threshold value  $\theta_f \geq 0$  such that  $f(\sqrt{x})$  is a convex function of  $x$  for  $x \leq \theta_f$  and a concave function of  $x$  for  $x \geq \theta_f$ .

The last requirement intuitively states that the discount function is S-shaped, with the value of  $\delta$  rising slowly at first, then sharply, and then slowly again. One example of such function is the double logistic function,  $f(x) = 1 - \exp(-x^2)$ .

For every well-behaved function we can bound the value of  $h(S)$  for any set  $S$  of points in a ball of radius  $R$ .

LEMMA 2. Assume  $f$  is a well-behaved function. For every set  $S$  contained in a ball of radius  $R$ ,  $h(S) \leq 16R^2\lambda(f) + 1$ .

Given the above lemma, the approximation factor of the algorithm can be bounded as follows:

THEOREM 1. Assume  $f$  is a well-behaved function. Then, the approximation ratio of PICKANDREMOVE( $R$ ) is at most  $(16R^2\lambda(f) + 1)/f(R) = O(1)$ .

PROOF. Fix an instance of the problem and an optimal solution  $OPT$  for this instance. We use a charging argument to show that the value of  $OPT$  is at most  $c$  times the value of the solution  $ALG$  produced by PICKANDREMOVE( $R$ ). Consider any point  $p$  in  $OPT$ . The algorithm PICKANDREMOVE( $R$ ) terminates when the set  $C$  is empty, and therefore the point  $p$  must be removed from  $C$  during the execution of the algorithm. Assume  $p$  is removed in an iteration where the point  $a_i$  is added to  $ALG$ . We charge  $p$  to  $a_i$ . By definition of the algorithm, it is clear that  $v_p \leq v_i$ .

We now argue that for every  $a_i \in ALG$ , the total value that  $OPT$  gets from the points charged to  $a_i$  is at most  $(1 + 16R^2\lambda(f))/f(R)$  times the value that  $ALG$  gets from  $a_i$ . It is clear that the value that  $ALG$  derives from  $a_i$  is at least  $v_i \cdot f(R)$ . Since  $v_p \leq v_i$  for every point  $p$  charged to  $a_i$ , the total value that  $OPT$  derives from such points is at most  $v_i \cdot h(S)$ , where  $S$  is the set of points in  $OPT$  charged to  $a_i$ . Since every such point is within distance  $R$  of  $a_i$ , by Lemma 2,  $h(S) \leq 16R^2\lambda(f) + 1$ .

Moreover, since  $f$  is well behaved, there is a finite value of  $R$  such that  $f(R) \geq 1/2$ , which implies that the approximation ratio is always bounded by a constant.  $\square$

All that remains is to prove Lemma 2.

PROOF OF LEMMA 2. Assume  $S$  is a set of points contained in a ball of radius  $R$  that maximizes the value of  $h(S)$ . If  $S$  is a singleton, then  $h(S) = 1$  and we are done. So, assume  $|S| \geq 2$ . For every  $a_i \in S$ ,  $r_i := \min_{a_j \in S \setminus \{a_i\}} \{d_{ij}\}$  is well-defined and finite. For each  $a_i \in S$ , consider a ball  $B_i$  of radius  $r_i/2$  around  $a_i$ . These balls are all disjoint, since if two balls  $B_i$  and  $B_j$  intersect, the distance between  $i$  and  $j$  must be less than  $r_i/2 + r_j/2 \leq \max(r_i, r_j)$ , which contradicts the definition of  $r_i$ 's. Furthermore, since  $r_i \leq 2R$  for all  $i$ , all  $B_i$ 's are contained in a ball of radius  $2R$ . This means that the total volume of  $B_i$ 's cannot exceed the volume of a ball of radius  $2R$ . In other words,

$$\sum_i \pi(r_i/2)^2 \leq \pi(2R)^2$$

or  $\sum_i r_i^2 \leq 16R^2$ . On the other hand,  $h(S) = \sum_i f(r_i)$ . Therefore,  $h(S)$  is bounded from above by the maximum of

$\sum_i f(r_i)$  subject to  $\sum_i r_i^2 \leq 16R^2$ . Denoting  $x_i := r_i^2$ , this reduces to the maximum of  $\sum_i f(\sqrt{x_i})$  subject to  $\sum_i x_i \leq 16R^2$ . Consider the optimal solution of this maximization program. For any two  $i$  and  $j$  where  $x_i, x_j \geq \theta_f$ ,  $x_i$  and  $x_j$  must be equal, since otherwise we can increase the objective function by increasing the smaller one and decreasing the larger one (due to the concavity of  $f$  in this range). Therefore, all  $x_i$ 's such that  $x_i \geq \theta_f$  are equal. Similarly, there can be at most one  $x_i$  in  $(0, \theta_f)$ , since if there are two, the objective function can be increased by increasing the larger one and decreasing the smaller one (due to the convexity of  $f$  in this range). Let  $k$  denote the number of  $x_i$ 's that are non-zero and at least  $\theta_f$ . The total value of these  $x_i$ 's is at most  $16R^2/k$ . The value of the  $x_i$  that is less than  $\theta_f$  (if any) is also bounded by the same value. Therefore, the total value  $\sum_i f(\sqrt{x_i})$  is at most

$$(k+1)f(4R/\sqrt{k}) = kf(4R/\sqrt{k}) + f(4R/\sqrt{k}) \leq 16R^2\lambda(f) + 1.$$

$\square$

Theorem 1 implies that the approximation factor of the algorithm depends on the function  $f$ . As an example, we compute this approximation factor with the discount function  $f(x) = 1 - e^{-x^2}$ , also known as the double logistic function.

COROLLARY 1. The map selection problem with a double logistic discounting function  $f(x) = 1 - e^{-x^2}$  is approximable within a factor of 22.35.

PROOF. For every  $x$ , we have  $e^{-x^2} \geq 1 - x^2$ . Therefore,

$$\lambda(f) = \frac{f(x)}{x^2} = \frac{1 - e^{-x^2}}{x^2} \leq 1.$$

Hence, the approximation factor can be bounded as:

$$\alpha^* \leq \min_R \frac{16R^2 + 1}{f(R)}$$

Numerical calculation shows that the above expression achieves its minimum at  $R \approx 0.578$ , achieving a value of  $\alpha^* \approx 22.342$ .  $\square$

## 4.2 Pricing for Ads

The optimization algorithm in the previous section is agnostic to the provenance of the candidate POIs: they can be organic results, or potential ads. In the latter case, an important part of the overall system is the pricing scheme: we want to set the prices such that each advertiser has an incentive to truthfully reveal the value of the ad.

We show how the algorithm in the previous section can be coupled with an efficient pricing scheme to obtain such an incentive-compatible mechanism. In general, the key point that determines if an approximation algorithm can be turned into an incentive compatible mechanism is whether the allocation function is monotone [3]. In the case of our problem, the allocation function is the function that for a fixed set of bids  $v_{-i}$  of advertisers other than  $i$ , maps the bid  $v_i$  of bidder  $i$  to  $\delta_i(S)$ , where  $S$  is the solution computed with the bid vector  $(v_i, v_{-i})$ . If this function is monotonically non-decreasing, then there is a payment function that is incentive compatible. In general, computing this payment function involves integrating over the allocation function, which can

be algorithmically inefficient. We show not only that the allocation rule is monotone, but also that there is a simple algorithm for computing truthful payments that has the same asymptotic running time as the allocation algorithm.

The pricing algorithm is presented in Algorithm 2. The algorithm proceeds as in the last section, processing points in decreasing order of their values, and, for each ad  $a_i$ , picks it unless another point within distance  $R$  is picked before. In addition, whenever the latter event occurs, it attributes the fact that ad  $a_i$  is blocked to the *unique* picked point  $a_j$  that is within distance  $R$  of  $a_i$  (if such a unique ad exists; if it does not, the event is not attributed to any ad). The price for a picked ad  $a_i$  is set to the maximum of the values of the ads that are blocked because of  $a_i$ , times the appropriate discounting coefficient.

---

**Algorithm 2** Map Ad Pricing Algorithm

---

```

1: function PICKANDREMOVEANDPRICE( $R$ )
2:   Sort candidate ads so that  $v_1 \geq v_2 \geq \dots \geq v_n$ 
3:   for  $i = 1 \rightarrow n$  do
4:      $p_i \leftarrow 0$ 
5:   end for
6:    $S \leftarrow \emptyset$ 
7:   for  $j = 1 \rightarrow n$  do
8:      $b \leftarrow 0$   $\triangleright$  number of other ads blocking  $a_j$ 
9:     for  $k = 1 \rightarrow j - 1$  do
10:      if  $a_k \in S$  and  $d_{kj} < R$  then
11:         $i \leftarrow k$ 
12:         $b \leftarrow b + 1$ 
13:      end if
14:    end for
15:    if  $b = 0$  then  $\triangleright a_j$  is not blocked by any ad
16:       $S \leftarrow S \cup \{a_j\}$ 
17:    else if  $b = 1$  then  $\triangleright a_j$  is blocked by unique ad
18:       $p_i \leftarrow \max\{p_i, v_j\}$ 
19:    end if
20:  end for
21:  return set  $S$  of winners, price  $p_i \cdot f(\min_{a_j \in S \setminus \{a_i\}} \{d_{ij}\})$ 
    for  $i$ .
22: end function

```

---

**THEOREM 2.** *The mechanism defined by Algorithm 2 is incentive-compatible.*

**PROOF SKETCH.** The main insight behind the proof is the following: For each ad  $a_i$  that is picked by the algorithm PICKANDREMOVE( $R$ ), if we compare the execution of this algorithm with the execution of the same algorithm when ad  $a_i$  is removed, the two executions look exactly the same up until the point where the execution without  $a_i$  picks an ad  $a_j$  that is within distance  $R$  of  $a_i$ . This means that if we change the value of ad  $a_i$  to  $v$  and run the algorithm, for  $v > v_j$ , the algorithm picks the exact same set of ads (except that it picks  $a_i$  later if  $v$  is smaller), and if  $v < v_j$ , it will pick the ad  $a_j$ , and after that, cannot pick  $a_i$ . Therefore, the value  $v_j$  is the threshold value above which ad  $a_i$  will be picked in the solution, and the discount factor  $f(\min_{a_j \in S \setminus \{a_i\}} \{d_{ij}\})$  is the same when  $a_i$  is picked.

What remains is to prove that the threshold value  $v_j$  described above is equal to the  $p_i$  computed by the algorithm. Note that since the algorithm initially sorts ads in decreasing order of values, the value  $p_i$  at the end of the algorithm is either zero (if we never reach line 18 with this value of

$i$ ), or is equal to the first  $v_j$  for which line 18 is reached. We argue that this  $j$  is the same as the  $j$  defined above. To clarify the notation, let  $j_1$  denote the first value of  $j$  for which line 18 is reached with  $i$  and  $j$ , and  $j_2$  denote the first  $j$  so that  $a_j$  is within distance  $R$  of  $a_i$  that is picked by PICKANDREMOVEANDPRICE( $R$ ) on an input without  $a_i$ . By definition the condition in line 10 is satisfied for  $j = j_2$  and  $k = i$ . Furthermore, this condition should not be satisfied for  $j = j_2$  and any other value of  $k$ , since if it is,  $j_2$  would not be picked by PICKANDREMOVEANDPRICE( $R$ ) on the instance without  $a_i$ . Thus, line 18 must be reached with this value of  $i$  and  $j = j_2$ . Next, we prove that this line is not reached for any  $j < j_2$  (and the same value of  $i$ ). Assume, for contradiction, that it is reached for one such  $j$ . Consider the execution of PICKANDREMOVEANDPRICE( $R$ ) on the instance without  $a_i$ . It is easy to see that in this scenario the algorithm must pick  $a_j$ , since the only selected ad blocking it is  $a_i$ , which is removed. This contradicts with  $j_2$  being the *first* ad within distance  $R$  of  $a_i$  picked in this scenario.  $\square$

### 4.3 Non-Uniform Weights

The algorithms we presented so far work in the regime where the discount function is indifferent to the similarity between the POIs. This is equivalent to assuming  $w_{ij} = 1$  for all pairs  $(i, j)$ . In this section we explore the setting of non-homogenous weights. We show that in general the optimization problem is NP-complete, and very hard to approximate. We then turn our attention to a more limited setting and show that the optimization algorithm presented before remains approximately welfare optimizing.

We consider the special family of discount functions  $\{g\}$ , which multiply the effects of similarity and distance, namely:  $f(w_{ij}, d_{ij}) = g(d_{ij}/w_{ij})$ . Note that just like  $f$ ,  $g$  is increasing in distance and decreasing in similarity. We prove that even for this special class of discount functions, and even fixing any set of pairwise distances  $d_{ij}$ , the problem of finding the optimal set of POIs given the weights  $w_{ij}$  is NP-complete.

For this result, we need the discount function  $g$  to satisfy a mild non-triviality condition: For every  $n$ , there has to be a  $c > 0$  and  $0 < t_{\min} < t_{\max}$  such that  $g(x) < c/n$  for  $x \in (0, t_{\min})$  and  $g(x) > c$  for  $x \in [t_{\max}, \infty)$ . Roughly speaking the condition requires the function  $g$  to make full use of the input range. We say a function  $g$  is *non-trivial* if it satisfies the above condition.

**THEOREM 3.** *For any non-trivial discount function  $g$  and every set of distances  $\{d_{ij}\}$ , the problem of optimal POI selection given the weights  $w_{ij}$  is NP-complete.*

**PROOF.** We reduce from the maximum independent set problem. Given a graph  $G = (V, E)$  with  $n$  vertices, we construct a set of weights  $\{w_{ij}\}$  such that the resulting instance of the POI selection problem (with vertex  $i$  corresponding to POI  $a_i$ ) captures the independent set problem in  $G$ .

Let  $n$  be the number of vertices in  $G$ . By non-triviality of  $g$ , there are  $c > 0$  and  $0 < t_{\min} < t_{\max}$  such that  $g(x) < c/n$  for  $x \in (0, t_{\min})$  and  $g(x) > c$  for  $x \in [t_{\max}, \infty)$ . Associate each node  $v_i \in V$  with POI  $a_i$  with value 1. For any two points  $a_i$  and  $a_j$ , let

$$w_{ij} = \begin{cases} d_{ij}/t_{\min}, & \text{if } (v_i, v_j) \in E \\ d_{ij}/t_{\max}, & \text{otherwise.} \end{cases}$$

Now consider any optimal solution to the welfare optimization problem,  $S^*$ . We show that the nodes of  $G$  corresponding to  $S^*$  must be an independent set. Assume, for contradiction, that a node  $v$  in  $S^*$  has an edge to another node  $u \in S^*$ . We construct another solution by removing all such  $u$ 's. By the definition of  $w_{ij}$ 's,  $g(d_{ij}/w_{ij})$  is less than  $c/n$  for  $(i, j) \in E$  and is greater than  $c$  otherwise. Therefore, removing all nodes  $u$  that have an edge to  $v$  removes at most  $n$  terms of value less than  $c/n$  and adds a term of value greater than  $c$ . This means that the resulting set has higher total value than  $S^*$ , contradicting the optimality of  $S^*$ . This means that solving the POI selection problem on this instance is equivalent to solving the independent set problem on  $G$ .  $\square$

The lower bound presented above may seem limiting, but the similarity weights  $w_{ij}$  usually have additional structure. For example, consider the following simple model, where we partition the points into  $k$  distinct classes  $A = A_1 \cup A_2 \cup \dots \cup A_k$ . In this setting,  $A_1$  may represent restaurants,  $A_2$ , mechanics,  $A_3$ , dentists, and so on. We impose a binary structure on the similarity weights: any two POIs in the same class have a high similarity of  $w_H$ , whereas any two POIs in different classes have a low similarity of  $w_L < w_H$ .

Now consider the intuition behind Algorithm 1. When the weights are identical, upon selecting a POI, the algorithm removes all of the other competing advertisers within radius  $R$ . When the similarity weights are not uniform, we adapt the algorithm as follows. There are two radii to consider:  $R_H$  for points in the same cluster, and  $R_L$  for points across clusters. The algorithm then iteratively picks POIs of maximum value, and every time a POI is picked, it removes all points in the same class within radius  $R_H$ , and all points in other classes within radius  $R_L$ .

**PROPOSITION 1.** *Assume  $f$  is a well behaved function, and  $w_H > w_L$  as above. Then the approximation ratio of the modified PickAndRemove algorithm is at most twice the ratio of the algorithm with uniform weights.*

**PROOF SKETCH.** We proceed as before, charging every point  $a$  in OPT to the POI selected by the algorithm that resulted in the removal of  $a$  from consideration. There are now two sets of points that are removed, those from the same class and those from a different class. By the same logic as in Theorem 1, we can bound the total value of points in the same class to  $16R_H^2(\lambda_f) + 1$ . Moreover, since  $R_L \leq R_H$ , the total value of points in the different class is at most  $16R_L^2(\lambda_f) + 1 < 16R_H^2(\lambda_f) + 1$  and the Proposition follows.  $\square$

## 5. LOCATION-AWARE SELECTION

We now turn to the POI selection problem in the location-aware model, where we have either exact or approximate information on the user's location. As we observe in Section 5.1, if the user location is known without any uncertainty, corresponding to the case of a single ranking  $\pi$  of the POIs based on their distance to the user, the problem can be solved exactly using a dynamic programming algorithm. The more technically challenging case is when the user location is uncertain, requiring an optimization with respect to a convex combination of multiple permutations. We study this case in Section 5.2, where we first give a general  $O(\log n)$ -approximation algorithm, and then improve this factor to

$O(1)$  for important special cases such as geometric and concave discount functions.

### 5.1 Known User Location

When the user location is known exactly, the distances between the user location and POI locations induce a single ordering  $\pi$  on the items. We give a simple dynamic program that solves the optimization problem.

Order the POIs according to  $\pi$ , (i.e.,  $a_1$  is the POI closest to the user, and so on). Let  $U[i, j]$  be the value of the optimal solution that only picks points from among those in  $i, \dots, n$  and allocates them to slots  $j, \dots, n$  in the permutation (i.e., getting discounts  $\gamma_j, \dots, \gamma_n$ ). The base case has  $U[\cdot, n+1] = U[n+1, \cdot] = 0$ . The recursive definition considers whether POI  $a_i$  should be allocated to slot  $j$ :

$$U[i, j] = \max(U[i+1, j], \gamma_j \cdot v_i + U[i+1, j+1]).$$

The value of the optimal solution is the value in cell  $U[1, 1]$ . In case of advertisements, this algorithm yields a simple mechanism: for the payments of each winner, we run the dynamic program without that winner to determine their VCG payment.

### 5.2 Uncertain User Location

If the location of the user is uncertain, and we are only given a prior over possible locations, we can phrase the optimization problem as that of finding the maximum expected social welfare, where the expectation is over the prior. We recall the notation below.

Each possible user location induces a permutation  $\pi$  over the points of interest ordered by distance to the user. Let  $\Pi = \{\pi_1, \pi_2, \dots, \pi_k\}$  be the set of possible permutations, with permutation  $\pi_i$  occurring  $p_i$  fraction of the time. For a set of selected POIs  $S$ , and a set of permutations  $\Pi$ , let  $U(\Pi, S)$  denote the expected social welfare,

$$U(\Pi, S) = \sum_{i=1}^k p_i \sum_{j \in S} v_j \gamma_{\pi_i(j, S)}.$$

We first give a simple greedy algorithm that achieves an  $O(\log n)$  approximation to the optimum solution for general non-increasing discounts  $\gamma$ . We show in Section 5.4 that the same algorithm gives a constant approximation ratio when the discounts are linear, concave, or geometric.

#### 5.2.1 The LARGESTVALUEPREFIX Algorithm

Algorithm LARGESTVALUEPREFIX first orders the POIs in order of decreasing value  $v_1 \geq v_2 \geq \dots \geq v_n$ . Let  $a_i$  be the POI corresponding to the value  $v_i$ . Define the prefix sets,  $X_1 = \{a_1\}$ ,  $X_2 = \{a_1, a_2\}$ , and generally  $X_i = \{a_1, a_2, \dots, a_i\}$ . The algorithm returns the best set among all of these,  $S = \arg \max_{X_i} U(\Pi, X_i)$ .

**LEMMA 3.** LARGESTVALUEPREFIX gives an  $\mathcal{H}_n = O(\log n)$  approximation to the optimal solution.

**PROOF.** Let  $ALG$  be the value returned by the algorithm, and  $OPT$  be the value of the optimal solution. We first show a simple upper bound on  $OPT$ , denoted by  $\overline{OPT}$ . When the POIs are ordered in order of non-increasing values,  $v_1 \geq v_2 \geq \dots \geq v_n$ , a simple swapping argument shows that:

$$OPT \leq v_1 \gamma_1 + v_2 \gamma_2 + \dots + v_n \gamma_n = \sum_{i=1}^n v_i \gamma_i = \overline{OPT}$$

The value of the set selected by the algorithm can be lower bounded as follows:

$$\begin{aligned} \text{ALG} &\geq \gamma_1 \cdot v_1 \\ \text{ALG} &\geq \gamma_2 \cdot (v_1 + v_2) \geq 2 \cdot \gamma_2 \cdot v_2 \\ &\dots \\ \text{ALG} &\geq \gamma_n \cdot (v_1 + \dots + v_n) \geq n \cdot \gamma_n \cdot v_n. \end{aligned}$$

So for every  $i$ :  $v_i \gamma_i \leq \text{ALG}/i$ . Therefore:

$$\text{OPT} \leq \overline{\text{OPT}} = \sum_{i=1}^n v_i \gamma_i \leq \sum_{i=1}^n \frac{1}{i} \text{ALG} = \mathcal{H}_n \cdot \text{ALG}$$

where  $\mathcal{H}_n = O(\log n)$  is the  $n^{\text{th}}$  harmonic number.  $\square$

The analysis presented above is almost tight.

LEMMA 4. *There exist instances where the solution returned by LARGESTVALUEPREFIX is an  $\Omega(\frac{\log n}{\log \log n})$  approximation to the optimum.*

PROOF. The hard instance applies even when there is a single permutation  $\pi$ . Let  $k$  be a constant that we fix later. For a schematic depiction of the following description, refer to Figure 1. The POIs come in a set of  $2k$  blocks. Block  $2i$  has  $M_i$  POIs, each of value  $v_i$ . Block  $2i-1$  has  $M_i - M_{i-1}$  POIs, each with value  $v_i^+ = v_i + \eta$  for some small value of  $\eta$ . Finally, we fix an  $\epsilon$ , and set values with  $v_i = \epsilon^{i-1}$ , and the block size  $M_0 = 0, M_1 = 1$ , and  $M_i = \epsilon^{-2(i-1)}$  for  $i > 1$ .

To set the discounts, let  $\gamma_1 = 1$ ,  $\gamma_i = \epsilon$  for  $i \in (M_1, M_1 + M_2]$ ,  $\gamma_i = \epsilon^2$  for  $i \in (M_1 + M_2, M_1 + M_2 + M_3]$ , and more generally  $\gamma_i = \epsilon^\ell$  for  $i \in (\sum_{j=1}^\ell M_i, \sum_{j=1}^{\ell+1} M_i]$ .

One solution is to pick the blocks of value  $v_i$ , and leave all of the blocks with value  $v_i^+$ . This solution has value:

$$\sum_{i=1}^k M_i \gamma_i v_i = \sum_{i=1}^k 1 = k \leq \text{OPT}.$$

The LARGESTVALUEPREFIX algorithm takes all of the elements. For the sake of analysis, we can increase all discounts to be at least  $\epsilon^{k-1}$ . Then the utility achieved by the algorithm is at most:

$$\begin{aligned} &\gamma_1 v_1^+ + \sum_{i=2}^k [M_{i-1} v_{i-1}^+ \gamma_i + (M_i - M_{i-1}) \gamma_i v_{i+1}] + M_k \gamma_k v_k \\ &\leq (1 + \eta)(\epsilon + (k-1) \cdot 2\epsilon) + 1 \\ &\leq 3k\epsilon + 1 \end{aligned}$$

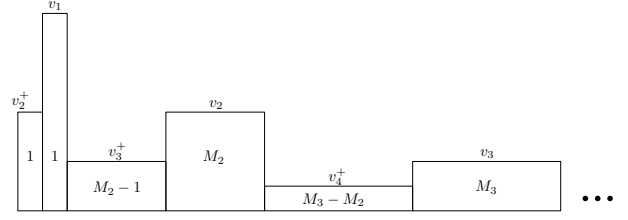
Therefore, the approximation ratio is at least  $\frac{k}{3k\epsilon+1}$ . We set  $\epsilon = 1/k$ , therefore the numerator is exactly 4. To compute the total number of POIs,

$$n = \sum_{i=1}^k M_i = \sum_{i=0}^{k-1} k^{2^i}.$$

Therefore we have that  $k^k \leq n \leq k^{2^k}$ . Then,  $\frac{1}{2} \log n \leq k \log k \leq \log n$  and  $k \geq \frac{\log n}{2 \log k}$ . Combining that, we have

$$\frac{1}{3k\epsilon + 1} \cdot k \geq \frac{1}{4} \cdot \frac{\log n}{2 \log k} \geq \frac{\log n}{8 \log \log n}$$

Therefore, the LARGESTVALUEPREFIX algorithm will give an  $\Omega(\frac{\log n}{\log \log n})$  approximation ratio on the instance.  $\square$



**Figure 1: The lower bound example for LARGESTVALUEPREFIX.** We have  $v_i = \epsilon^{i-1}$  and  $M_i = 1/\epsilon^{2(i-1)}$ .  $v_i^+$  is just barely larger than  $v_i$  and therefore LARGESTVALUEPREFIX will pick it earlier. At each discount level  $\gamma_i$  there can be  $M_i$  POIs.

### 5.3 Pricing

Again, for the case of advertisement selection, we give a pricing scheme that leads to an incentive compatible mechanism. While the welfare-maximizing allocation is always monotone, this is not generally true for approximately maximizing allocation rules. However, the following Lemma shows that in the case of the LARGESTVALUEPREFIX algorithm, the monotonicity property holds. (Due to space constraints, we defer the proof to the full version of the paper.)

LEMMA 5. *LARGESTVALUEPREFIX is a monotone allocation rule.*

For mechanisms with a monotone allocation rule, there is a unique payment rule (where losers pay nothing) that makes the mechanism dominant-strategy incentive compatible [3, 16]. The payment is in terms of critical values  $z_j$  where the allocation jumps (as the value of the bidder changes), and the magnitude of those jumps. It is easy to see that there are polynomially many such critical points and they can be computed in polynomial time. This gives us a polynomial-time dominant-strategy incentive compatible mechanism.

### 5.4 Better Bounds for Special Discount Functions

We now consider some important special cases of discount functions for which LARGESTVALUEPREFIX is a constant-factor approximation. The general proof structure is as follows: for the special discount functions we focus on a particular set  $X_j$  that the algorithm will consider. By using information on the discount function, we show that picking  $X_j$  gives a constant-factor approximation to the optimal social welfare. Since LARGESTVALUEPREFIX yields a solution that is at least as good as  $X_j$ , this shows that LARGESTVALUEPREFIX performs well for these discount functions. Note that we do not need to change the algorithm; the same algorithm will provably perform better through a better analysis. **Geometric Discounts.** We define geometric discounts to be those of the form  $\gamma_i = \alpha^{i-1}$  with  $\alpha \in (0, 1)$ .

LEMMA 6. *Let  $\gamma$  be a geometric discount function, and let  $j$  be the largest index such that  $\gamma_j \geq \frac{1}{2}$ . Then  $X_j$  is a 3-approximation to  $\text{OPT}$ .*

PROOF. Recall that  $\overline{\text{OPT}} \geq \text{OPT}$  selects all POIs  $A$  for an optimal value of  $\sum_{a_i \in A} v_i \gamma_i$  where  $v_i$  is in decreasing order. We split up the value of  $\overline{\text{OPT}}$  into that which is

gained from POIs in  $X_j$ , and that which is gained from the remaining POIs  $A \setminus X_j$ . Let  $ALG$  denote the value of picking  $X_j$  as solution.

Let  $U^*(X_j)$  be the contribution of elements in  $X_j$  to  $\overline{OPT}$ . For the elements that are in  $X_j$ , their discount is at least  $1/2$  hence their contribution to  $ALG$  is at least half of that in  $\overline{OPT}$ :  $U^*(X_j) \leq 2 \cdot ALG$ .

For the remaining elements we show that  $U^*(A \setminus X_j) \leq ALG$ . In the following let  $v_j$  be the value of the smallest element of  $X_j$ . Recall that for any  $\alpha \in (0, 1)$ :  $\sum_{i=0}^{j-1} \alpha^i = \frac{1-\alpha^j}{1-\alpha}$  and  $\sum_{i=j}^{\infty} \alpha^i = \frac{\alpha^j}{1-\alpha}$ . Using this and  $\gamma_j \geq \frac{1}{2}$ , we have:

$$U^*(A \setminus X_j) \leq v_j \cdot \sum_{i=j+1}^{\infty} \gamma_i \leq v_j \cdot \sum_{i=1}^j \gamma_i \leq ALG$$

where the first inequality comes from  $v_j$  being a lower bound on values of POIs in  $A \setminus X_j$ , the second inequality from the bounds on geometric sequences and  $\gamma_j \geq 1/2$ , and the last inequality because  $v_j$  is a lower bound on all values of POIs in  $ALG$ . Combining this we get:

$$\begin{aligned} OPT &\leq \overline{OPT} = U^*(X_j) + U^*(A \setminus X_j) \\ &\leq 2 \cdot ALG + ALG \\ &= 3 \cdot ALG. \end{aligned}$$

Since LARGESTVALUEPREFIX performs at least as well as picking  $X_j$ , this completes the proof.  $\square$

**Linear and Concave Discounts.** We define concave discount functions as follows: for a number  $0 < k \leq n$  a concave discount function satisfies:  $\gamma_{i-1} + \gamma_{i+1} - 2\gamma_i \leq 0$ , for all  $i \in [2, k-1]$  and  $\gamma_i = 0$  for  $i \geq k$ .

When the above inequality is tight, the function is linear and has  $\gamma_i = 1 - \frac{i-1}{k}$ . We will prove the bound for linear discounts first, and then extend it to all concave discounts. We defer both proofs to the full version of the paper.

**LEMMA 7.** *Let  $\gamma$  be a linear discount function, and let  $j$  be  $\lceil (k+1)/2 \rceil$  such that  $j$  is the largest index with  $\gamma_j \geq \frac{1}{2}$ .  $X_j$  is a  $\frac{7}{3}$ -approximation to  $OPT$*

This result holds more generally for concave discount functions (of which linear discounts are a special case).

**LEMMA 8.** *Let  $\gamma$  be a concave discount function, and let  $j$  be the largest index with  $\gamma_j \geq \frac{1}{2}$ . Then  $X_j$  is a  $\frac{7}{3}$ -approximation to  $OPT$*

## 6. COMBINATION OF THE TWO MODELS

In this section we give a natural algorithm for the model which combines the pairwise discounts described in Section 4 and the rank based discounts described in Section 5. This captures the situation where we know the location of the user, but would still like to spread out markers for aesthetic reasons. In the combined model the discount function is the product of the discounts in the two models. More specifically, for an POI  $a_i$  in a set  $S$  of displayed establishments, we have

$$\delta_i(S) = \min_{j \in S \setminus a_i} \{\delta_{ij}\} \cdot \gamma_{\pi(i,S)}, \quad (1)$$

where  $\delta_{ij} = f(w_{ij}, d_{ij})$  is a function of the relevance  $w_{ij}$  of POIs  $a_i$  and  $a_j$  and their distance  $d_{ij}$ , and  $\pi(i, S)$  is the

ranking of  $a_i$  in the ordering of  $S$  according to the distance to the user location. The goal is again to pick the set  $S$  that maximizes the total discounted value.

### 6.1 Algorithm

To select such a set  $S$ , we first run PICKANDREMOVE without the rank discounts. Intuitively, the resulting set is “well-separated” in that any two POIs are far apart from each other. We then run the LARGESTVALUEPREFIX algorithm on this set of POIs, further pruning down the list of POIs to show on the map. Call this algorithm LARGEST-PRUNEDPREFIX.

For the setting of the problem let  $\alpha$  be the approximation ratio of the PICKANDREMOVE algorithm, recall that this ratio depends on the planar density  $\lambda$  of the pairwise discount function  $f$ , as well as the choice of  $R$  used by the algorithm. Similarly, let  $\beta$  be the approximation ratio of LARGESTVALUEPREFIX. The value of  $\beta$  depends on the shape of the rank discount function  $\gamma$ , and ranges from constant to  $O(\log n)$ .

**THEOREM 4.** *LARGESTPRUNEDPREFIX has an approximation ratio of  $\alpha \cdot \beta$ , for  $\alpha, \beta$  as defined above. For pairwise discount functions  $f$  with bounded planar density, and geometric or concave rank discount functions  $\gamma$ , this is a constant competitive algorithm.*

**PROOF.** Let  $A$  be the full set of POIs,  $T$  be the set of POIs that are returned by the call to PICKANDREMOVE, and let  $S$  be the set that is returned by calling LARGESTVALUEPREFIX on  $T$ . Order the POIs in descending order of value,  $v_i$ . Let  $OPT_i$  be the set of elements in  $OPT$  that are charged to  $a_i$ , as in the proof of Theorem 1. Let  $\delta(i, S) = \min_{a_j \in S \setminus a_i} \{d_{ij}\}$  the discount due to competition in  $S$ . Let  $c_f(R) = 16R^2\lambda(f)+1$  from the proof of Theorem 1. Then:

$$\begin{aligned} OPT &= \sum_{a_i \in T} \sum_{a_j \in OPT_i} v_j \cdot \delta(j, OPT) \cdot \gamma_{\pi(j, OPT)} \\ &\leq \sum_{a_i \in T} \left( \sum_{a_j \in OPT_i} v_j \cdot \delta(j, OPT) \right) \cdot \max_{a_j \in OPT_i} \gamma_{\pi(j, OPT)} \\ &\leq \sum_{a_i \in T} c_f(R) \cdot v_i \cdot \max_{a_j \in OPT_i} \gamma_{\pi(j, OPT)} \quad (2) \end{aligned}$$

$$\begin{aligned} &\leq c_f(R) \cdot \sum_{a_i \in T} v_i \cdot \max_{a_j \in OPT_i} \gamma_{\pi(j, OPT)} \\ &\leq c_f(R) \cdot \sum_{a_i \in T} v_i \cdot \gamma_{i'} \quad (3) \end{aligned}$$

$$\leq c_f(R) \cdot \beta \cdot \sum_{a_i \in S} v_i \cdot \gamma_{\pi(i, S)} \quad (4)$$

$$\begin{aligned} &\leq \frac{c_f(R)}{f(R)} \cdot \beta \cdot \sum_{a_i \in S} v_i \cdot \delta(i, S) \cdot \gamma_{\pi(i, S)} \quad (5) \\ &= \alpha \cdot \beta \cdot ALG. \end{aligned}$$

Here (2) follows from the proof of Theorem 1. For (3), let  $i'$  be the rank of  $a_i$  in  $T$  in descending order by values, the statement then follows because the sum is maximized when the  $i^{\text{th}}$  highest POI in  $T$  has the  $i^{\text{th}}$  largest discount. Line (4) follows from the analysis of LARGESTVALUEPREFIX; and finally (5) follows from all POIs being spaced at least  $R$  from each other.  $\square$



LEMMA 9. LARGESTPRUNEPREFIX yields a monotone allocation rule.

PROOF SKETCH. The proof relies on two main ideas: first, if a POI is selected by PICKANDREMOVE, it will always be selected as part of the same set  $T$ , hence the input to LARGESTVALUEPREFIX is always the same. We can then extend the monotonicity proof of LARGESTVALUEPREFIX with respect to (1). We defer the formal proof to the full version of the paper.  $\square$

COROLLARY 2. LARGESTPRUNEDPREFIX is a dominant-strategy incentive compatible mechanism that obtains an  $\alpha\beta$  approximation of the social welfare, with  $\alpha$  the approximation ratio of PICKANDREMOVE and  $\beta$  the approximation ratio of LARGESTVALUEPREFIX.

## 7. EXPERIMENTAL EVALUATION

In this section, we present experimental evaluation of the algorithms presented and analyzed theoretically in previous sections, and show that they also perform well on real data. We start by describing the data sets we use to evaluate the algorithms.

### 7.1 Data

We use a dataset of restaurants publicly available from the crawl done by [5]. From this dataset, we have extracted the list of restaurants in six cities, ranging from large to suburban: Berkeley, Brooklyn, Chicago, Mountain View, Palo Alto, and San Francisco. The number of restaurants in these data sets are 340, 1894, 3047, 304, 171, and 2705. These restaurants will be our POIs. We take the value of each POI to be its rating, which is a number between 1 and 10. To avoid ranking restaurants with a single good review higher than the ones with many reviews, we calculate the rating of each restaurant as  $(\text{avg. rating} \cdot \text{num. reviews} + 10)$  divided by  $(\text{number of reviews} + 2)$ . This corresponds to assuming prior with a beta distribution with parameters  $\alpha = \beta = 10$ .

### 7.2 Location-unaware model

For the discount function, we use the double logistic function  $f(x) = 1 - e^{-x^2}$  discussed in Corollary 1. This function discounts a POI at distance 0.8km from another by roughly a factor of  $\frac{1}{2}$ . As a baseline, we run two simple algorithms that keep a subset of POIs of a certain target size, either uniformly at random (RANDOMTHINNING) or with probabilities proportional to the value of the POI (PROPRANDOMTHINNING). The algorithms we intend to evaluate are PICKANDREMOVE( $R$ ) for  $R$  ranging from 0.1 to 2 Km, and a greedy algorithm (GREEDY) that sorts POIs in decreasing order of their values, and iteratively picks points from this list if the marginal value of the point is positive. While the GREEDY algorithm may seem appealing, it is possible to construct examples that show that in the worst case its approximation ratio is unbounded. Moreover, while GREEDY performs comparably to PICKANDREMOVE, our approach is more efficient as GREEDY requires updating all of the marginal values at every iteration.

The value of the objective function on the solution given by PICKANDREMOVE( $R$ ) is plotted as a function of  $R$  in Figure 2 (all values are scaled to fit on the same plot). As can be seen in this figure, the optimal value of  $R$  ranges from 0.6 to 1.05. Curiously, this is not far from the value of  $R$  that optimizes the worst-case performance (0.578).

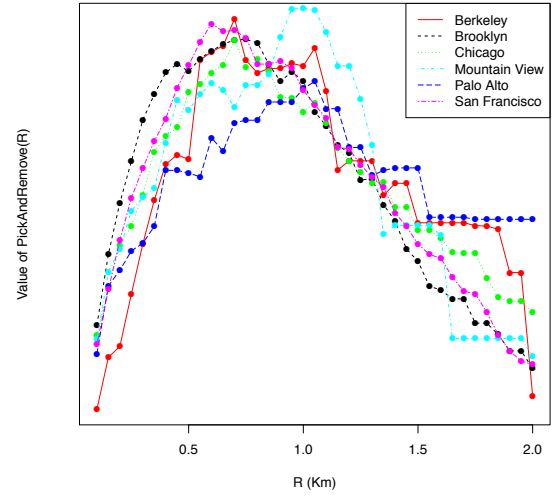


Figure 2: PICKANDREMOVE( $R$ ) as a function of  $R$

City	PickRem	Greedy	Rand	Prop
Berkeley	77.52	71.08	23.39	23.62
Brooklyn	272.83	271.19	90.84	92.53
Chicago	244.96	239.31	70.22	73.29
Mountain View	73.62	64.72	23.55	23.04
Palo Alto	62.38	60.72	20.12	22.57
San Francisco	372.58	370.26	111.22	109.04

Table 1: Value in the location-unaware model

Table 1 shows the value of the solutions computed by various algorithms on the different data sets. For PICKANDREMOVE, RANDOMTHINNING, and PROPRANDOMTHINNING, the values shown correspond to the value obtained from the best choice of  $R$  or the target size. As can be seen in this table, PICKANDREMOVE is the winner in all cases, and GREEDY is a close second. The value achieved by these algorithms is close to 3 times higher than the baselines. Interestingly, PROPRANDOMTHINNING does not perform significantly better than RANDOMTHINNING and sometimes even performs worse. We believe the reason for this effect is that good restaurants tend to be located close to each other, and therefore PROPRANDOMTHINNING is more likely to pick POIs that are close to each other.

### 7.3 Location-aware model

For the discount function in this model, we use exponential discounts  $\gamma_r = 0.8^{r-1}$ . For the user location, we pick a random location in the corresponding city and add a number of random perturbations of this point to model uncertainty. We use the same baselines, RANDOMTHINNING and PROPRANDOMTHINNING, and the algorithm we evaluate is LARGESTVALUEPREFIX.

The results are presented in Table 2. As can be seen in this table, LARGESTVALUEPREFIX performs significantly better than both baselines, although the margin is not as large as it was in the case of the location-unaware model. The size of the solutions picked by LARGESTVALUEPREFIX for these data sets are 18, 30, 35, 20, 24, and 35, respectively.

City	LargestValuePrefix	Rand	Prop
Berkeley	46.12	34.92	36.04
Brooklyn	48.35	35.27	36.73
Chicago	97.83	70.12	74.11
Mountain View	45.68	35.83	37.02
Palo Alto	43.64	34.78	36.02
San Francisco	49.18	35.97	37.15

**Table 2: Value in the location-aware model**

## 8. CONCLUSION AND OPEN PROBLEMS

In this paper, we initiated the study of the problem of selecting the set of points of interest to place on the map, and finding incentive compatible mechanisms in the case some of these are advertisements. We studied two scenarios, those where the user location is unknown, with a focus on externalities due to competing POIs, and those with known user location, with a focus on rank based externalities. For both problems, as well as the combination of the two information models, we give simple greedy approximation algorithms coupled with incentive-compatible pricing schemes.

There are still many open questions remaining. In particular, it would be useful to be able to accommodate additional constraints, such as explicit density restrictions (i.e., upper bound on the number of POIs that can be placed on each tile of the map), consistency constraint on the set of selected POIs at different zoom levels (see [18] for a treatment of this constraint), or POIs that have positive externalities on each other. Finally, improving the approximation ratio of our algorithms (or coming up with other practical algorithms with better ratios) is an interesting open problem.

## 9. REFERENCES

- [1] Gagan Aggarwal, Jon Feldman, S. Muthukrishnan, and Martin Pál. Sponsored search auctions with markovian users. In Christos Papadimitriou and Shuzhong Zhang, editors, *Internet and Network Economics*, volume 5385 of *Lecture Notes in Computer Science*, pages 621–628. Springer Berlin Heidelberg, 2008.
- [2] Rakesh Agrawal, Sreenivas Gollapudi, Alan Halverson, and Samuel Ieong. Diversifying search results. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining*, WSDM '09, pages 5–14, 2009.
- [3] Aaron Archer and Éva Tardos. Truthful mechanisms for one-parameter agents. *Proceedings of 42th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 482–491, 2001.
- [4] Moshe Babaioff and Liad Blumrosen. Computationally-feasible truthful auctions for convex bundles. *Games and Economic Behavior*, 63(2):588–620, 2008.
- [5] Saeideh Bakhshi, Partha Kanuparth, and Eric Gilbert. Demographics, weather and online reviews: A study of restaurant recommendations. In *Proceedings of the 23rd International Conference on World Wide Web*, WWW '14, pages 443–454, New York, NY, USA, 2014. ACM.
- [6] Tony Bradley. Google maps ads are a big opportunity for local businesses. PC World (August 9, 2013). [www.pcworld.com/article/2046292/google-maps-ads-are-a-big-opportunity-for-local-businesses.html](http://www.pcworld.com/article/2046292/google-maps-ads-are-a-big-opportunity-for-local-businesses.html).
- [7] Cartography and Geographic Information Society. Cagis map design competition. <http://www.cartogis.org/awards/contest.php>, 2013.
- [8] Arpita Ghosh and Mohammad Mahdian. Externalities in online advertising. In *Proceedings of the 17th international conference on World Wide Web*, pages 161–168. ACM, 2008.
- [9] Arpita Ghosh and Amin Sayedi. Expressive auctions for externalities in online advertising. In *Proceedings of the 19th international conference on World wide web*, pages 371–380. ACM, 2010.
- [10] Sreenivas Gollapudi and Aneesh Sharma. An axiomatic approach for result diversification. In *WWW '09: Proceedings of the 18th international conference on World wide web*, pages 381–390, New York, NY, USA, 2009. ACM.
- [11] Renato Gomes, Nicole Immorlica, and Evangelos Markakis. Externalities in keyword auctions: An empirical and theoretical assessment. In Stefano Leonardi, editor, *Internet and Network Economics*, volume 5929 of *Lecture Notes in Computer Science*, pages 172–183. Springer Berlin Heidelberg, 2009.
- [12] Samuel Ieong, Mohammad Mahdian, and Sergei Vassilvitskii. Advertising in a stream. In *Proceedings of the 23rd International Conference on World Wide Web*, WWW '14, pages 29–38, New York, NY, USA, 2014. ACM.
- [13] David Kempe and Mohammad Mahdian. A cascade model for externalities in sponsored search. In *Internet and Network Economics*, pages 585–596. Springer, 2008.
- [14] Ravi Kumar, Mohammad Mahdian, Bo Pang, Andrew Tomkins, and Sergei Vassilvitskii. Modeling geographic choice. In *Proceedings of the 8th International Conference on Web Search and Data Mining (WSDM)*, 2015.
- [15] Stephanie Meece. A bird's eye view of a leopard's spots. the Çatalhöyük 'map' and the development of cartographic representation in prehistory. <https://www.repository.cam.ac.uk/handle/1810/195777>, 2008.
- [16] Roger Myerson. Optimal auction design. *Mathematics of Operations Research*, 6(1):58–73, 1981.
- [17] Anastasios Noulas, Salvatore Scellato, Renaud Lambiotte, Massimiliano Pontil, and Cecilia Mascolo. A tale of many cities: Universal patterns in human urban mobility. *PLoS ONE*, 7(5):e37027, 05 2012.
- [18] Anish Das Sarma, Hongrae Lee, Hector Gonzalez, Jayant Madhavan, and Alon Halevy. Consistent thinning of large geographical data for map visualization. *ACM Trans. Database Syst.*, 38(4):22:1–22:35, December 2013.
- [19] Greg Sterling. Google introduces offline maps for mobile, claims a billion users globally for maps, earth. Search Engine Land weblog (June 6, 2012). <http://searchengineland.com/live-blogging-the-google-maps-next-dimension-event-123617>, 2012.
- [20] Jennifer van Grove. Marissa Mayer: Google will connect the digital and physical worlds through mobile. Mashable weblog (March 11, 2011). <http://mashable.com/2011/03/11/mayer-sxsw-talk/>, 2011.
- [21] Wikipedia. Babylonian map of the world. [http://en.wikipedia.org/wiki/Babylonian\\_Map\\_of\\_the\\_World](http://en.wikipedia.org/wiki/Babylonian_Map_of_the_World), 2014.