Trajectory-driven Influential Billboard Placement

Ping Zhang
Wuhan Univeristy
pingzhang@whu.edu.cn

Zhifeng Bao
RMIT University
zhifeng.bao@rmit.edu.au

Yuchen Li
Singapore Management University
yuchenli@smu.edu.sg

Guoliang Li
Tsinghua University
liguoliang@tsinghua.edu.cn

Yipeng Zhang
RMIT University
s3582779@student.rmit.edu.au

Zhiyong Peng
Wuhan Univeristy
peng@whu.edu.cn

ABSTRACT

In this paper we propose and study the problem of trajectory-driven influential billboard placement: given a set of billboards U (each with a location and a cost), a database of trajectories $\mathcal{T}$ and a budget L, find a set of billboards within the budget to influence the largest number of trajectories. One core challenge is to identify and reduce the overlap of the influence from different billboards to the same trajectories, while keeping the budget constraint into consideration. We show that this problem is NP-hard and present an enumeration based algorithm with $(1 - 1/e)$ approximation ratio. However, the enumeration-based method is costly when $|U|$ is large. By exploiting the locality property of billboards' influence, we propose a partition-based framework PartSel. PartSel partitions U into a set of small clusters, computes the locally influential billboards for each cluster, and merges them to generate the global solution. Since the local solutions can be obtained much more efficient than the global one, PartSel can reduce the computation cost greatly; meanwhile it achieves a non-trivial approximation ratio guarantee. Then we propose a LazyProbe method to further prune billboards with low marginal influence, while achieving the same approximation ratio as PartSel. Experiments on real datasets verify the efficiency and effectiveness of our methods.

CCS CONCEPTS

• **Mathematics of computing** → **Combinatorial optimization**; *Enumeration*; • **Applied computing** → *Marketing*;

KEYWORDS

Outdoor Advertising, Influence Maximization, Trajectory

ACM Reference Format:
Ping Zhang, Zhifeng Bao, Yuchen Li, Guoliang Li, Yipeng Zhang, and Zhiyong Peng. 2018. Trajectory-driven Influential Billboard Placement. In *KDD '18: The 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, August 19–23, 2018, London, United Kingdom.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3219819.3219946

KDD '18, August 19–23, 2018, London, United Kingdom

ACM ISBN 978-1-4503-5552-0/18/08…$15.00
https://doi.org/10.1145/3219819.3219946

1 INTRODUCTION

Outdoor advertising (ad) has a $500 billion global market [2]. Compared to TV and mobile advertising, it delivers a high return on investment, e.g., an average of $5.97 is generated in product sales for each dollar spent[3]. Moreover, it literally drives consumers 'from big screen to small screen' to search, interact, and transact[4]. Billboards are the highest used medium for outdoor ads (about 65%), and 80% people notice them when driving[5].

Nevertheless, existing market research only leverages traffic volume to assess the performance of billboards [15]. Such a straightforward approach often leads to coarse-grained estimations and undesirable ad placement plans. Enabled by the prevalence of positioning devices, tremendous amounts of user/vehicle trajectories are being generated. It enables us to propose a fine-grained model to quantify the billboard influence over a database of trajectories. Intuitively, if a billboard is close to a trajectory along which a user or vehicle travels, the billboard influences the user to a certain degree. When multiple billboards are close to a trajectory, the marginal influence is reduced to capture the property of diminishing returns.

Based on this influence model, we propose and study the Trajectory-driven Influential Billboard Placement (TIP) problem: given a set of billboards (each with a location and non-uniform cost), a database of trajectories and a budget constraint L, it finds a set of billboards within budget L such that the placed ads on the selected billboards influence the largest number of trajectories. The primary goal is to maximize the influence within a budget, which is critical to advertisers because the average cost per billboard is not cheap. For example, the average cost of a billboard is $14000 for four weeks in New York [1]; the total cost of renting 500 billboards is $7,000,000 per month. Since the cost of a billboard is usually proportional to its influence, if we can improve the influence by 5%, we can save about $10,000 per week for one advertiser. The secondary goal is to avoid expensive computation while achieving the same competitive influence value, so that prompt analytic on deployment plans can be conducted with different budget allocations.

There are two challenges to achieve the above goals. First, a user's trajectory can be influenced by multiple billboards, which incurs the influence overlap among billboards. Figure 1 shows 6 billboards ($b_1, \ldots, b_6$) and 6 trajectories ($t_1, \ldots, t_6$). Each billboard is associated with a λ-radius circle as its influence range. If any

[1] Zhiyong Peng is the corresponding author.
[2] https://www.statista.com/topics/979/advertising-in-the-us/
[3] http://oaaa.org/StayConnected/NewsArticles/IndustryRevenue/tabid/322/id/4928
[4] http://www.alloutdigital.com/2012/09/what-are-some-advantages-of-digital-billboard-advertising
[5] http://www.runningboards.com.au/outdoor/relocatable-billboards

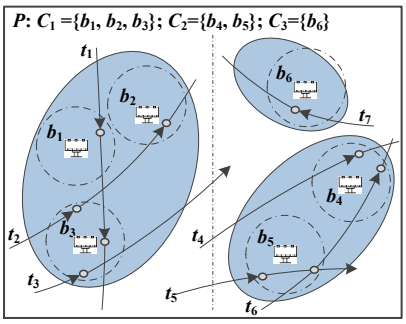


Figure 1: A Motivating Example ($w(b_i) = i$)

point p of a trajectory t lays in the circle of b, t is influenced by b with a certain probability. Thereby, t_1 is first influenced by b_1 and then by b_3. If the selected billboards have a large overlap in their influenced trajectories, advertisers may waste the money for repeatedly influencing the audiences who have already seen their ads. Second, the budget constraint L and non-uniform costs of different billboards make the optimization problem intricate. To our best knowledge, this is the first work that simultaneously takes three critical real-world features into consideration, i.e., budget constraint, non-uniform costs of billboards, and influence overlap of the selected billboards to a certain trajectory (Section 2). It is worth noting that the solution to this problem is useful in any store site selection problem that needs to consider the influence gain w.r.t. the cost of the store under a budget constraint. The only change is a customization of the influence model catered for specific scenarios, while the influence overlap is always incurred whenever the audiences are moving. For example in the electric vehicle charging station deployment, each station has an installment fee and a service range, which is similar to the billboard in TIP. Given a budget limit, its goal is to maximize the deployment benefit, which can be measured by the trajectories that can be serviced by the stations deployed.

To address these challenges, we first propose a greedy method EnumSel by employing the enumeration technique [8], which can provide an $(1 - 1/e)$-approximation for TIP (Section 3). However, this algorithm runs in a prohibitively large complexity of $O(|\mathcal{T}| \cdot |U|^5)$, where $|\mathcal{T}|$ and $|U|$ are the number of trajectories and billboards respectively. To avoid such high computational cost, we exploit the locality property of the billboard influence and propose a partition-based framework PartSel (Section 4). The core idea works as follows: first, it partitions the billboards into a set of clusters with low influence overlap; second, it executes the enumeration algorithm to find local solutions; third, it uses the dynamic programming approach to construct the global solution based on the local solutions maintained by different clusters. We show that the partition based method provides a $\frac{1}{2}^{\lceil \log_{(1+1/\theta)} m \rceil}(1 - 1/e)$-approximate solution, where θ is a user defined parameter to balance between efficiency and effectiveness, ranging between 0 and 1.

To further improve the efficiency of our PartSel framework, we devise a lazy probe approach by pro-actively estimating the upper bound of each cluster and combining the results from a cluster only when its upper bound is significant enough to contribute to the global solution (Section 5). Last, we conduct extensive experiments on real-world trajectory and billboard datasets. Our best method LazyProbe significantly outperforms the traditional greedy approach in terms of quality improvement over the naive traffic volume approach by about 99%, and provides competitive quality against the EnumSel baseline while achieving 30×-90× speedup in efficiency (Section 6). All the proofs of theorems and lemmas, and the time complexity analysis of our solutions are in technical report [26].

2 PRELIMINARIES

2.1 Problem Formulation

In a trajectory database $\mathcal{T}$, each trajectory $t = \{p_1, p_2, ..., p_{|t|}\}$, where p_i consists of the latitude *lat* and longitude *lng*. A billboard b={*loc*, w}, where *loc* and w denote b's location and leasing cost, respectively. Without loss of generality, we assume that a billboard carries either zero or one ad at any time.

Definition 2.1. *We define that b can influence t, if* $\exists p_i \in t$, *such that* $Distance(p_i, b.loc) \leq \lambda$, *where* $Distance(p_i, b.loc)$ *computes a certain distance between* p_i *and b.loc, and* λ *is a given threshold.*

The choice of distance function is orthogonal to our solution, and we choose Euclidean distance for illustration purpose.

Influence of a billboard b_i to a trajectory t_j: $pr(b_i, t_j)$. The influence can be measured in various ways depending on application needs, such as the panel size, the exposure frequency, the travel speed and the travel direction. Our solution of finding the optimal placement is orthogonal to the choice of influence measurement, so long as it can be computed deterministically given a b_i and t_j. By looking into the influence measure of one of the largest outdoor advertising companies LAMAR [1], we observe that panel size and exposure frequency are mainly used. Moreover, they can be obtained from the real data, hence we adopt them in our influence model illustration and experiment. (1) For all $b_i \in U$ and $t_j \in \mathcal{T}$, we set $pr(b_i, t_j)$ as a uniform value (between 0 and 1) if b_i can influence t_j. (2) Let $size(b_i)$ be the panel size of b_i. We set $pr(b_i, t_j) = size(b_i)/A$ for t_j influenced by b_i, where A is a given value that is larger than $\max_{b_i \in U} size(b_i)$.

Influence of a billboard set S to a trajectory t_j: $pr(S, t_j)$. Since different billboards in S may have overlaps when they influence t_j, $pr(S, t_j)$ cannot be simply computed as $\sum_{b_i \in S} pr(b_i, t_j)$. Obviously $pr(S, t_j)$ should be the probability that at least one billboard in S can influence t_j. Thus, we use the following equation to compute the influence of S to t_j.

$$pr(S, t_j) = 1 - \prod\nolimits_{b_i \in S} (1 - pr(b_i, t_j)) \quad (1)$$

Influence of a billboard set S to a trajectory set $\mathcal{T}$: $I(S)$. Let $\mathcal{T}_S$ denote the set of trajectories in $\mathcal{T}$ that are influenced by at least one billboard in S. The influence of a billboard set S to a trajectory set $\mathcal{T}$ is computed by summing up $pr(S, t_j)$ for $t_j \in \mathcal{T}_S$:

$$I(S) = \sum\nolimits_{t_j \in \mathcal{T}_S} pr(S, t_j) \quad (2)$$

Definition 2.2. **(Trajectory-driven Influential Billboard Placement (TIP))** Given a trajectory database $\mathcal{T}$, a set of billboards U to place ads and a cost budget L from a client, our goal is to select

a subset of billboards $S \subset U$, which maximizes the expected number of influenced trajectories such that the total cost of billboards in S does not exceed budget L.

Theorem 2.1. *The TIP problem is NP-hard.*

2.2 Related Work

Maximized Bichromatic Reverse k Nearest Neighbor (MaxBRkNN). The MaxBRkNN queries [5, 16, 24, 27] aim to find the optimal location to establish a new store such that it is a kNN of the maximum number of users based on the spatial distance between the store and users' locations. Different spatial properties are exploited to develop efficient algorithms, such as space partitioning [27], intersecting geometric shapes [24], and sweep-line techniques [16]. Recently, the MaxRKNN query [22] is proposed to find the optimal bus route in term of maximum bus capacity by considering the audiences' source-destination trajectory data. Regarding the usage of trajectory data, most recent work only focus on top-k search over trajectory data [21, 23].

Our TIP problem is different from MaxBRkNN in two aspects. (1) MaxBRkNN assumes that each user is associated with a fixed (check-in) location. In reality, the audience can meet more than one billboard while moving along a trajectory, which is captured by the TIP model. Thus it is challenging to identify such influence overlap when those billboards belong to the same placement strategy. (2) Billboards at different locations may have different costs, making this budget-constrained optimization problem more intricate. However, MaxBRkNN assumes that the costs of candidate store locations are uniform.

Influence Maximization and its variations. The original Influence Maximization (IM) problem aims to find a size-k subset of all nodes in a social network that could maximize the spread of influence [7]. Independent Cascade (IC) model and Linear Threshold (LT) model are two common models to capture the influence spread. Under both models, this problem has been proven to be NP-hard, and a simple greedy algorithm guarantees the best possible approximation ratio of $(1 - 1/e)$ in polynomial time. Then the key challenge lies in how to calculate the influence of sets efficiently, and a plethora of algorithms [2–4, 9, 19] have been proposed to achieve speedups. Some new models are also introduced to solve IM under complex scenarios. IM problems for propagating different viral products are studied in [11, 13]. Recently, the IM problem is extended to location-aware IM (LIM) problems by considering different spatial contexts [6, 10, 15]. Li et al. [10] find the seed users in a location-aware social network such that the seeds have the highest influence upon a group of audiences in a specified region. Guo et al. [6] select top-k influential trajectories based on users' checkin locations. See a recent survey [14] for more details.

Our TIP differs from the IM problems as follows. (1) The cardinality of the optimal set in IM problems is often pre-determined because the cost of each candidate is equal to each other (when the cost is 1, the cardinality is k), thus a theoretically guaranteed solution can be directly obtained by a naive greedy algorithm. However, in our problem, the costs of billboards at different locations differ from one to another, so the theoretical guarantee of the naive greedy algorithm is poor [8]. (2) Since IM problems adopt a different influence model to ours, they mainly focus on how to efficiently and effectively estimate the influence propagation, while TIP focuses on how to optimize the profit of k-combination by leveraging the geographical properties of billboards and trajectories.

Algorithm 1: GreedySel (U, L, S)

1.1 **repeat**
1.2 Select $b \in U \setminus S$ that maximizes $\frac{\Delta(b|S)}{w(\{b\})}$
1.3 **if** $w(S) + w(b) \leq L$ **then**
1.4 $S \leftarrow S \cup \{b\}$
1.5 $U \leftarrow U \setminus \{b\}$
1.6 **until** $U = \phi$ *or* $w(S) \geq L$;
1.7 $H \leftarrow \text{argmax}\{I(\{b\}) | \ b \in U, \text{and } w(\{b\}) \leq L\}$
1.8 **If** $I(H) > I(S)$ **return** H; **otherwise**, **return** S

3 OUR FRAMEWORK

3.1 Baselines

We first present two baselines that are extended from the algorithms for the general Budgeted Maximum Coverage (BMC) problem.

3.1.1 A Basic Greedy Method. A straightforward way (lines 1.1-1.6 of Algorithm 1) is to select the billboard b that maximizes the unit marginal influence, i.e., $\frac{\Delta(b|S)}{w(\{b\})}$, to a candidate solution set S, until the budget is exhausted, where $\Delta(b|S)$ denotes the marginal influence of b to S, i.e., $I(S \cup \{b\}) - I(S)$. However, it cannot achieve a guaranteed approximation ratio. For example, given two billboards b_1 with influence 1 and b_2 with influence x. Let $w(b_1) = 1$, $w(b_2) = x + 1$, $L = x + 1$. The optimal solution is S=$\{b_2\}$ with $I(S)$=x, while the solution picked by the greedy is S=$\{b_1\}$ with $I(S)$=1. In this case the approximation factor is x, which can be arbitrarily large.

Khuller et al. [8] propose GreedySel (lines 1.7-1.8 in Algorithm 1) to overcome this issue by naive greedy. However, it is misclaimed in [8] that GreedySel achieves an approximation factor of $(1 - 1/\sqrt{e})$. In fact, the approximation ratio of GreedySel should be $\frac{1}{2}(1-1/e)$ (in Theorem 3.1). The time complexity of Algorithm 1 is $O(|\mathcal{T}| \cdot |U|^2)$.

Theorem 3.1. *GreedySel achieves an approximation factor of* $\frac{1}{2}(1-1/e)$ *for the TIP problem.*

3.1.2 Enumeration Greedy Algorithm. Since GreedySel is only $\frac{1}{2}(1-1/e)$-approximate, we aim to further boost the influence value, even at the expense of longer run time as compared to GreedySel. It is critical to maximize the influence as it can save real money, while keeping acceptable efficiency. Thus we utilize the enumeration-based solution proposed in [8] to obtain $(1 - 1/e)$-approximation.

Algorithm 2: EnumSel (U, L)

2.1 Let τ be a constant /* τ=2 to achieve the lowest time complexity */
2.2 $H_1 \leftarrow \text{argmax}\{I(S') | \ S' \subseteq U, |S'| \leq \tau, \text{and } w(S') \leq L\}$
2.3 $H_2 \leftarrow \phi$
2.4 **for** *all* $S \subseteq U$, *such that* $|S| = \tau + 1$ *and* $w(S) \leq L$ **do**
2.5 $S \leftarrow$ **GreedySel**$(U \setminus S, L - w(S), S)$
2.6 **if** $I(S) > I(H_2)$ **then**
2.7 $H_2 \leftarrow S$
2.8 **If** $I(H_1) > I(H_2)$ **return** H_1; **otherwise**, **return** H_2

As shown in Algorithm 2, EnumSel runs in two phases. In the first phase (line 2.2), it enumerates all feasible billboard sets whose cardinality is no larger than a constant τ, and adds the one with the largest influence to H_1. In the second phase (lines 2.3-2.7), it enumerates each feasible set of size-$(\tau+1)$ whose total cost does not exceed budget L. Then for each set S, it invokes NaiveGreedy to select new billboards (if any) that can bring marginal influence, then chooses the one that maximizes the influence under the remaining budget $L - w(S)$ and assigns it to H_2. Last, if the best influence of all size-$(\tau+1)$ billboard sets is still smaller than that of its size-τ counterpart (i.e., $I(H_1) > I(H_2)$), H_1 is returned; otherwise, H_2 is returned. The time complexity of Algorithm 2 is $O(|\mathcal{T}| \cdot |U|^{\tau} + |\mathcal{T}| \cdot |U|^{\tau+3}) = O(|\mathcal{T}| \cdot |U|^{\tau+3})$.

Selection of τ. It has been proved in [8] that Algorithm 2 can achieve an approximation factor of $(1 - 1/e)$ when $\tau \geq 2$. Note that (1) the approximation ratio $(1 - 1/e)$ cannot be improved by a polynomial algorithm [8] and (2) a larger τ leads to larger overhead, thus we set $\tau = 2$. Thus, Algorithm 2 can achieve the $(1 - 1/e)$-approximation ratio with a time complexity of $O(|\mathcal{T}| \cdot |U|^5)$.

3.2 A Partition-based Framework

Although EnumSel provides an approximation ratio of $(1 - 1/e)$, it involves high computation cost, because it needs to enumerate all size-τ and size-$(\tau+1)$ billboard sets and compute their influence to the trajectories, which is impractical when $|U|$ and $|\mathcal{T}|$ are large. To address this problem, we propose a partition-based framework. Important notations used are presented in Table 1.

Partition-based Framework. Our problem has a distance requirement that if a billboard influences a trajectory, the trajectory must have a point close to the billboard (distance within λ). All existing techniques neglect this important feature. After a careful investigation, we observe that most trajectories span over a small area in real world. For instance, around 85% taxi trajectories in New York do not exceed five kilometers (see Section 6). It implies that billboards in different areas should have small overlaps in their influenced trajectories. Thereby, we exploit such locality features to propose a partition based method called PartSel. Intuitively, we partition U into a set of small clusters, compute the locally influential billboards for each cluster, and merge them to generate the globally influential billboards of U. Since the local cluster has much smaller number of billboards, this method reduces the computation greatly while keeping competitive influence quality.

Partition. We first partition the billboards to m clusters $C_1, C_2, \ldots, C_m$, where different clusters have no (or little) influence overlap to the same trajectories. Given a budget l_i for cluster C_i, by calling $EnumSel(C_i, l_i)$, we select the locally influential billboard set $S[i][l_i]$ from cluster C_i within budget l_i, where $S[i][l_i]$ has the maximum influence $\xi[i][l_i]$. Next we want to assign a budget to each cluster C_i and take the union of $S[i][l_i]$ as the globally influential billboard set, where $l_1 + l_2 + \ldots + l_m \leq L$. Obviously, we want to allocate the budget to different clusters to maximize

$$\sum_{i=1}^{m} \xi[i][l_i] \quad \text{s.t.} \quad l_1 + l_2 + \ldots + l_m \leq L \tag{3}$$

There are two main challenges in this partition based method. (1) How to allocate the budgets to each cluster to maximize the overall influence? We propose a dynamic programming algorithm to address this challenge (see Section 4). (2) How to partition the billboards to reduce the influence overlap among clusters? We propose a partition strategy to reduce the influence overlap and devise an effective algorithm to generate the clusters (see Section 4).

Table 1: Important Notations in Solutions

Symbol	Description
U	A set of billboards that a user wants to advertise
$I(S)$	The influence of a selected billboard set S
P	A billboard partition
ϑ_{ij}	The overlap ratio between clusters
$\Delta(b\|S)$	The marginal influence of b to S
$\mathbb{I}$	The DP influence matrix: $\mathbb{I}[i][l]$ is the maximum influence of the billboards selected from the first i clusters within budget l ($i \leq m$ and $l \leq L$)
ξ	The local influence matrix: $\xi[i][l]$ is the influence returned by $EnumSel(C_i, l)$, i.e., the maximum influence of billboards selected from cluster C_i within budget l

Lazy Probe. In the partition based method, its dynamic programming process has to repeatably invoke $EnumSel(C_i, l_i)$ to compute the local influence for every cluster in the partition. Thus, we propose a lazy probe method to estimate an upper bound $\xi^{\uparrow}[i][l_i]$ of the local solution for a given cluster C_i and a budget l_i. If using this cluster cannot improve the influence, we do not need to compute the real influence $\xi[i][l_i]$. There are two challenges. (1) How to utilize the bounds to reduce computation cost (i.e., avoid calling $EnumSel(C_i, l_i)$)? (2) How to estimate the upper bounds while keeping the same approximation ratio as PartSel? We devise an incremental algorithm to estimate the bounds (Section 5).

Algorithm 3: PartSel (P θ-partition , L)

```
3.1 Initialize matrices 𝕀 and ξ; m ← |P|
3.2 for i ← 1 to m do
3.3     for l ← 1 to L do
3.4         Invoke EnumSel(C_i, l) to compute ξ[i][l]
3.5         q = arg max_{0≤q≤l} (𝕀[i−1][l−q] + ξ[i][q])
3.6         𝕀[i][l] ← 𝕀[i−1][l−q] + ξ[i][q]
3.7 S ← the corresponding selected set of 𝕀[m][L]
3.8 return S
```

4 PARTITION BASED METHOD

For convenience sake, we first present how to select the billboards based on a given partition scheme, and then discuss how to find the partition efficiently.

4.1 Partition based Selection Method

Definition 4.1. (**Partition**) A partition of U is a set of clusters $\{C_1, ..., C_m\}$, such that $U = C_1 \cup C_2 \cup ... \cup C_m$, and $\forall i \neq j, C_i \bigcap C_j = \phi$. Without loss of generality, we assume that the clusters are sorted by their size, and C_m is the largest cluster.

We follow a divide and conquer framework to combine partial solutions from the clusters. Let S^* denote the billboard set returned

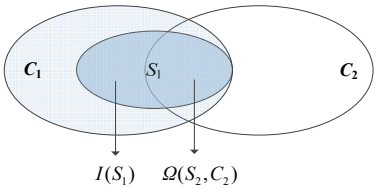


Figure 2: The relationship of S_1, C_2 and $\Omega(S_1, C_2)$

by $EnumSel(U, L)$, $S[i][l]$ denote the billboard set returned by $EnumSel(C_i, l)$, where $l < L$ is a budget for cluster C_i. Let $\xi[i][l]$ be the influence value of the billboard set $S[i][l]$, i.e., $\xi[i][l] = I(S[i][l])$. If $S[i][l]$ for $1 \le i \le m$ has no overlap, we can assign a budget l for each cluster and maximize the total influence based on Equation 3.

We note that the costs for billboards are integers in reality, e.g., the costs from a leading outdoor advertising company are all multiples of 100 [1]. Thereby it allows us to design an efficient dynamic programming method to solve Equation 3. The pseudo code is presented in Algorithm 3. It considers the clusters in P one by one. Let $\mathbb{I}[i][l]$ denote the maximum influence value that can be attained with a budget not exceeding l using up to the first i clusters ($i \le m$ and $l \le L$). Clearly, $\mathbb{I}[m][L]$ is the solution for Equation 3 since the union of the first m clusters is U. To obtain $\mathbb{I}[m][L]$, Algorithm 3 first initializes the matrices $\mathbb{I}$ and ξ (line 3.1), and then constructs the global solution (line 3.2 to 3.8) with the following recursion:

$$\begin{aligned} &\mathbb{I}[0][l] = 0 \\ &\mathbb{I}[i][l] = \max_{0 \le q \le l} (\mathbb{I}[i-1][l-q] + \xi[i][q]) \end{aligned} \tag{4}$$

The time complexity of Algorithm 3 is $\sum_{i=1}^{m} |C_i|^5$ which is bounded by $O(mL \cdot |\mathcal{T}| \cdot |C_m|^5)$. It is more efficient than Algorithm 2 ($O(|\mathcal{T}| \cdot |U|^5)$), since $|C_m|$ is often significantly smaller than $|U|$ and L is a constant.

4.2 θ-partition

In order to reduce the influence overlap between clusters, we introduce the concept of *Overlap Ratio*, which controls the maximum overlap ratio between any subset of a cluster and all the rest clusters.

Definition 4.2. (**Overlap Ratio**) For two clusters C_i and C_j, the ratio ϑ_{ij} of the overlap between C_i and C_j relative to C_i is

$$\vartheta_{ij} = \arg\max_{\forall S_i \subseteq C_i} \{\Omega(S_i, C_j)/I(S_i)\} \tag{5}$$

where S_i is a subset of C_i, and $\Omega(S_i, C_j)$ is the overlap between S_i to C_j (see Figure 2), i.e., $I(S_i) + I(C_j) - I(S_i \cup C_j)$.

Intuitively, the smaller ϑ_{ij} is, the lower influence overlap C_i and C_j have. Although there are other ways to define the overlap ratio, we find that our measure is more reasonable than the other feasible choices: (1) the volume of the clusters' overlap, i.e., $\vartheta_{ij} = \Omega(C_i, C_j)$; (2) the overlap ratio between billboards in a cluster and those that are not in this cluster. The details of those choices are discussed in our technical report [26].

Given the overlap ratio, we present the concept of θ-partition to trade-off between cluster size and the overlap of clusters, where θ is a user-defined parameter to control the granularity of partitions.

Definition 4.3. (θ-**partition**) Given a threshold $\theta \in [0, 1]$, we say a partition $P = \{C_1, .., C_m\}$ is a θ-partition, if $\forall i, j \in [1, m]$ the overlap ratio ϑ_{ij} between any pair of clusters $\{C_i, C_j\}$ is less than θ.

Lemma 4.1. *Let P be a θ-partition of U. Given any set $S \subseteq U$, and the billboards in S belong to k different clusters of P in total. When $k \le (1/\theta + 1)$, we have $I(S) \ge 1/2 \sum_{S_i \in S} I(S_i)$, where $S_i = S \cap C_i$.*

Proof. To facilitate our proof, we assume $S = \{S_1, S_2 ... S_k\}$ and $I(S_1) \ge, ..., \ge I(S_k)$. Let $\overline{I(S)}$ denote the average influence among all $S_i \in S$, i.e., $\overline{I(S)} = \frac{1}{k} \sum_{j=1}^{k} I(S_j)$. According to Definition 4.3, we observe that $I(S_i \cup S_j) \ge I(S_i) + (1-\theta) I(S_j)$, as each subset of S_j has at most θ percent of influence overlapping with the elements of S_i, or vice versa. Then for all subsets of S, we have:

$$\begin{aligned} I(S) &\ge I(S_1) + (1-\theta) I(S_2) + (1-2\theta) I(S_3) ... + [1-(k-1)\theta] I(S_k) \\ &= \sum_{i=1}^{k} I(S_i) - \theta [I(S_2) + 2I(S_3) + ... + (k-1) I(S_k)] \\ &= \sum_{i=1}^{k} I(S_i) - \theta [\sum_{i=2}^{k} I(S_i) + \sum_{i=3}^{k} I(S_i) + ... + I(S_k)] \\ &\ge \sum_{i=1}^{k} I(S_i) - \theta [\overline{I(S)} + 2\overline{I(S)} + ... + (k-1)\overline{I(S)}] \\ &= \sum_{i=1}^{k} I(S_i) - \theta \frac{k(k-1)}{2} \overline{I(S)} \end{aligned}$$

The second inequality above follows from the fact that $\overline{I(S)} \ge \frac{1}{k-1} \sum_{i=j}^{k} I(S_i)$ for $j = 2, 3 ... k$, because we have assumed $I(S_1) \ge , ..., \ge I(S_k)$. As $k \le \frac{1}{\theta} + 1$, we have $\theta \frac{k(k-1)}{2} \overline{I(S)} \le \frac{k}{2} \overline{I(S)}$ and

$$I(S) \ge \sum_{i=1}^{k} I(S_i) - \frac{k}{2} \overline{I(S)} = 1/2 \sum_{S_i \in S} I(S_i)$$

□

Based on Lemma 4.1, we proceed to derive the approximation ratio of Algorithm 3 in Theorem 4.2.

Theorem 4.2. *Given a θ-partition $P = \{C_1, .., C_m\}$, Algorithm 3 obtains a $\frac{1}{2}^{\lceil \log_{(1+1/\theta)} m \rceil}(1 - 1/e)$-approximation to the TIP problem.*

Note that when θ or m are small, this ratio is close to $(1 - 1/e)$.

4.3 Finding a θ-partition

It is worth noting that there may exist multiple θ-partitions of U (e.g., U is a trivial θ-partition). Recall Section 4.1, the time complexity of the partition based method (Algorithm 3) is $O(mL \cdot |\mathcal{T}| \cdot |C_m|^5)$, where $|C_m|$ is the size of the largest cluster in a partition P. Therefore, $|C_m|$ is an indicator of how good a θ-partition is, and we want to minimize $|C_m|$. Unfortunately, finding a good θ-partition is not trivial, since the it can be modeled as the balanced k-cut problem where each vertex in the graph is a billboard and each edge denotes two billboards with influence overlap, which is found to be NP-hard [20]. Therefore, we use an approximate θ-partition by employing a hierarchical clustering algorithm [17]. It first initializes each billboard as its own cluster, then it iteratively merges these two clusters into one, if their overlap ratio (Equation 5) is larger than θ. That is, for each pair of clusters $C_i, C_j \subseteq U$, if ϑ_{ij} is larger than θ, then C_i and C_j will be merged. By repeating this process, an approximate θ-partition is obtained when no cluster in U can be merged.

Note that how to efficiently get a θ-partition is not the key point of this paper and it can be processed offline; instead our focus is how to find the influential billboards based on a θ-partition.

5 LAZY PROBE

5.1 The Lazy Probe Algorithm

Recall that $\mathbb{I}[i][l]$ is the maximum influence value that can be attained with a budget not exceeding l using up to the first i clusters (in Section 4.1), and all clusters are processed in an order of their size (from the smallest to the largest by Definition 4.1). As mentioned in Section 4, $\mathbb{I}[i][l] = \max_{0 \le q \le l}(\mathbb{I}[i-1][l-q] + \xi[i][q])$, we need to find a q ($0 \le q \le l$) to maximize this influence. Note that $\mathbb{I}[i-1][l-q]$ has already been obtained in the $(i-1)$th iteration, but it is expensive to compute $\xi[i][q]$ by calling algorithm EnumSel. To address this issue, instead of computing the exact influence $\xi[i][q]$ in cluster C_i, we can estimate an upper bound of $\xi[i][q]$ for $0 \le q \le l$ (denoted by $\xi^{\uparrow}[i][q]$), and then prune q that cannot get larger influence by bound comparison.

Algorithm 4: LazyProbe(P, L)

4.1 Initialize two matrices $\mathbb{I}$ and ξ
4.2 **for** $i = 1$ *to* m **do**
4.3 **for** $l = 1$ *to* L **do**
4.4 $\mathbb{I}^{\downarrow}[i][l] \leftarrow \mathbb{I}[i-1][l]$
4.5 **for** $q = 1$ *to* l **do**
4.6 $\xi^{\uparrow}[i][q] \leftarrow$ **EstimateBound**(C_i, q)
4.7 **if** $\mathbb{I}^{\downarrow}[i][l] \le \mathbb{I}[i-1][l-q] + \xi^{\uparrow}[i][q]$ **then**
4.8 **if** $\xi[i][q]$ *has not been computed* **then**
4.9 Invoke **EnumSel**(C_i, q) to compute $\xi[i][q]$
4.10 **Update** $\mathbb{I}^{\downarrow}[i][l]$ **by** $\mathbb{I}[i-1][l-q] + \xi[i][q]$
4.11 **else**
4.12 continue;
4.13 $\mathbb{I}[i][l] \leftarrow \mathbb{I}^{\downarrow}[i][l]$
4.14 $S \leftarrow$ the corresponding selected set of $\mathbb{I}[i][l]$
4.15 **return** S

Function 5: EstimateBound(U, L)

5.1 S' = GreedySel (U, L, ϕ);
5.2 b_{k+1} is the next billboard with the largest unit marginal influence;
5.3 $\xi^{\uparrow}[i][q] = I(S') + \frac{\Delta(b_{k+1}|S')}{L - w(S')}$;
5.4 **return** $\xi^{\uparrow}[i][q]$;

Algorithm 4 describes how our method works. Similar to PartSel, we employ a dynamic programming approach to compute the selected billboard set and its influence value for each cluster i and each cost l. However, the difference is that we first compute the lower bound $\mathbb{I}^{\downarrow}[i][l]$ of $\mathbb{I}[i][l]$. Obviously $\mathbb{I}^{\downarrow}[i][l] = \mathbb{I}[i-1][l]$ is a naive lower bound by setting $q = 0$ (line 4.4). Then we compute an upper bound $\xi^{\uparrow}[i][q]$ from $q = 0$ to $q = l$ by calling function `EstimateBound`, which will be discussed later. Next if $\mathbb{I}^{\downarrow}[i][l] \ge \mathbb{I}[i-1][l-q] + \xi^{\uparrow}[i][q]$, we do not need to compute $\xi[i][q]$, because we cannot increase the influence using cluster C_i, and thus we can save the cost of calling EnumSel (lines 4.11-4.12). If $\mathbb{I}^{\downarrow}[i][l] < \mathbb{I}[i-1][l-q] + \xi^{\uparrow}[i][q]$, we need to compute $\xi[i][q]$, by calling *EnumSel*(C_i, q), and update $\mathbb{I}^{\downarrow}[i][l] = \mathbb{I}[i-1][l-q] + \xi[i][q]$ (lines 4.7-4.10). Finally, we set $\mathbb{I}[i][l]$ as $\mathbb{I}^{\downarrow}[i][l]$ since we already know $\mathbb{I}^{\downarrow}[i][l]$ is good enough to obtain the solution with a guaranteed approximation ratio (line 4.14), and return the corresponding selected billboard set as S (line 4.15).

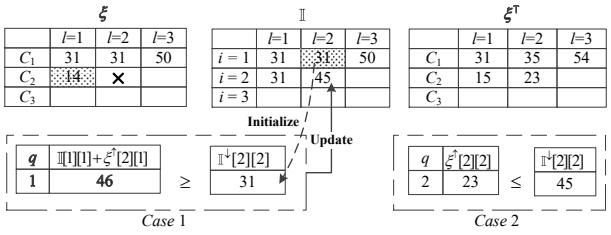


Figure 3: A running example for LazyProbe

Estimation of Upper Bound $\xi^{\uparrow}[i][q]$. A key challenge to ensure the approximation ratio of LazyProbe is to get a tight upper bound $\xi^{\uparrow}[i][q]$. Unfortunately, we observe that it is hard to obtain a tight upper bound efficiently due to the overlap influence among billboards. Fortunately, we can get an approximate bound, with which our algorithm can still guarantee the $(1 - 1/e)$ approximation ratio (see Section 5.2). To achieve this goal, we first utilize the basic greedy algorithm GreedySel to select the billboards $S' = \{b_1, b_2, \ldots, b_k\}$. Let b_{k+1} be the next billboard with the maximal marginal influence. If we include b_{k+1} in the selected billboards, then the cost will exceed L. If we do not include it, we will lost the cost of $L - w(S')$ where $w(S') = \sum_{1 \le i \le k} w(b_i)$. Then we can utilize the unit influence of b_{k+1} to remedy the lost cost, and thus we can get an upper bound $\xi^{\uparrow}[i][q] = I(S') + \frac{I(S' \cup \{b_{k+1}\}) - I(S')}{L - w(S')}$. We later show that $\xi^{\uparrow}[i][q] \ge (1 - 1/e)\xi[i][q]$. Moreover, the solution quality of Algorithm 4 remains the same as Algorithm 3. The details of the theoretical analysis will be presented in Section 5.2.

Example 5.1. Figure 3 shows an example on how Algorithm 4 works. There are three clusters C_1, C_2 and C_3 in a partition P, and the estimator matrix $\xi^{\uparrow}$ is shown in upper right corner. When C_i ($i = 1, 2, 3$) is considered, it computes $\mathbb{I}[i][l]$, for each $l = 1, .., L$, by the bound comparisons. Taking $\mathbb{I}[2][2]$ as an example, Algorithm 4 first initializes $\mathbb{I}^{\downarrow}[2][2] = \mathbb{I}[1][2]$ and then computes $\mathbb{I}[1][2 - q] + \xi^{\uparrow}[2][q]$ ($q = 1, 2$) for bound comparisons. For case 1 ($q = 1$), as $\mathbb{I}^{\downarrow}[2][2] \le \mathbb{I}^{\uparrow}[2][2]$, Algorithm 4 needs to compute $\xi[2][1]$ by invoking EnumSel and update $\mathbb{I}^{\downarrow}[2][2]$ by $\mathbb{I}[1][1] + \xi[2][1]$. For case 2 ($q = 2$), since $\mathbb{I}^{\downarrow}[2][2] \ge \mathbb{I}^{\uparrow}[2][2]$, $\mathbb{I}^{\downarrow}[2][2]$ does not need to be updated and finally $\mathbb{I}[2][2] = \mathbb{I}^{\downarrow}[2][2] = 45$.

5.2 Theoretical Analysis

In this section, we conduct theoretical analysis to establish the equivalence between LazyProbe and PartSel in terms of the approximation ratio. We first show that if the bound $\xi^{\uparrow}[i][l]$ in LazyProbe is $(1 - 1/e)$ approximate to the *TIP* instance of billboards in cluster i using budget l, then the approximation ratio of *LazyProbe* and *PartSel* is the same (Theorem 5.1). We then move on to show that $\xi^{\uparrow}[i][l]$ is indeed $(1 - 1/e)$-approximate in Lemmas 5.2-5.4.

Theorem 5.1. *If $\xi^{\uparrow}[i][l]$ obtained by Function 5 achieves a $(1-1/e)$ approximation ratio to the TIP instance for cluster i with budget l, LazyProbe ensures the same approximation ratio with PartSel presented in Section 4.1.*

Theorem 5.1 requires that $\xi^{\uparrow}[i][l]$ is $(1-1/e)$-approximate to the corresponding TIP instance in a small cluster. To show that $\xi^{\uparrow}[i][l]$ returned by Function 5 satisfies such requirement, we describe the following hypothetical scenario on the TIP instance for cluster

C_i and budget l. Let b_{k^*+1} be a billboard in the optimal solution, and it is the first billboard that violates the budget constraint in Algorithm 1. The following inequality holds [8].

Lemma 5.2. *[8] After the ith iteration ($i = 1, ..., k^* + 1$) of the hypothetical scenario running Algorithm 1, the following holds:*

$$I(S_i) \geq [1 - \prod_{j=1}^{i} (1 - \frac{w(b_j)}{L})] \cdot I(OPT) \quad (6)$$

Where S_i be the billboard set that is selected by the first i iterations of the hypothetical scenario.

With lemma 5.2, we analyze the solution quality of running the hypothetical scenario by using the $k^* + 1$ billboards to deduce $\xi^{\uparrow}[i][l]$ by Lemma 5.3.

Lemma 5.3. *Let $\mathcal{M}_{k^*+1}$ denote the unit marginal influence of adding b_{k^*+1} in the hypothetical scenario, i.e., $\mathcal{M}_{k^*+1} = [I(S_{k^*} \cup \{b_{k^*+1}\}) - I(S_{k^*})]/w(b_{k^*+1})$. Then $I(S_{k^*}) + [L - w(S_{k^*})] \cdot \mathcal{M}_{k^*+1} \geq (1 - 1/e)I(OPT)$.*

Proof. First, we observe that for $a_1, ..., a_n \in \mathbb{R}^+$ such that $\sum_{i=1}^{n} a_i = \alpha A$, the function

$$1 - \prod_{i=1}^{n} (1 - \frac{a_i}{A})$$

achieves its minimum of $1 - (1 - \alpha/n)^n$ when $a_1 = a_2 = ... = a_n = \beta A/n$, for $A, \beta \geq 0$.

Suppose b' is a virtual billboard with cost $L - w(S_{k^*})$ and the unit marginal influence of b' to S_i is $\mathcal{M}_{k^*+1}$, for $i = 1, ..., k^*$. We modify the instance by adding b' into U and let $U \cup \{b'\}$ be denoted by U'. Then after the first k^*th iterations of Algorithm 1 on this new instance, b' must be selected at the $(k^* + 1)$th iteration. As $L(S_{k^*}) + w(b') = L$, by applying Lemma 5.2 and the observation to $I(S')$ ($S' = S_k \cup \{b'\}$), we get:

$$\begin{aligned} I((S') &\geq \left[1 - \prod_{j=1}^{k^*+1} \left(1 - \frac{w(b_j)}{L}\right)\right] \cdot I(OPT') \\ &\geq \left(1 - (1 - \frac{1}{k^* + 1})^{k^*+1}\right) \cdot I(OPT') \\ &\geq (1 - \frac{1}{e}) \cdot I(OPT') \end{aligned}$$

Note that $I(OPT')$ is surely larger than $I(OPT)$, thus $I(S_{k^*}) + [L - w(S_{k^*})]\mathcal{M}_{k^*+1} = I(S') \geq (1 - \frac{1}{e}) \cdot I(OPT') \geq (1 - \frac{1}{e}) \cdot I(OPT)$. □

Finally, we show that the estimator obtained by Function 5 is larger than the bound value obtained by the hypothetical scenario described in Lemma 5.3, which indicates that Function 5 is $(1-1/e)$-approximate and it further implies that Theorem 5.1 holds.

Lemma 5.4. *Given an instance of TIP. Let $\xi[i][l]$ be an estimator returned by Function 5, we have $\xi[i][l] \geq (1 - 1/e)I(OPT)$.*

6 EXPERIMENT

6.1 Experimental Setup

Datasets. We collect the billboard and trajectory data for the two largest cities in US, i.e., NYC and LA.

1) **Billboard** data is crawled from LAMAR[6], one of the largest outdoor advertising companies worldwide.

[6]http://www.lamar.com/InventoryBrowser

2) **Trajectory** data is obtained from: (1) the TLC trip record dataset[7] for NYC, (2) the Foursquare check-in dataset[8] for LA.

Table 2: Statistics of Datasets

	$\|\mathcal{T}\|$	$\|U\|$	AvgDistance	AvgTravelTime	AvgPoint#
NYC	4m	1500	2.9km	569s	159
LA	200k	2500	2.7km	511s	138

Table 3: Parameter Settings

Parameter	Values
L	100k, **150k**, ... 300k
$\|\mathcal{T}\|$ (NYC)	40k, ...,**120k**..., 4m
$\|\mathcal{T}\|$ (LA)	40k, 80k,**120k**, 160k, 200k
$\|U\|$ (NYC)	0.5k, 1k, 1.46k, (**2k**...10k by replication)
$\|U\|$ (LA)	1k, **2k**, 3k, (4k... 10k by replication)
θ	0, 0.1, **0.2**, 0.3, 0.4
λ	25m, **50m**, 75m, 100m

For NYC, we collect taxi trips from Jan 2013 to Sep 2016. Each trip record includes the pick-up and drop-off locations, time and trip distances. We use Google Maps API[9] to generate the trajectories, and if the distance of the recommended route by Google is close to the trip distance and travel time in the record (within 5% error rate), we use this route as an approximation of this trip's real trajectory. As a result, we obtain 4 million trajectories. For LA, as there is no public taxi record, we collect the Foursquare checkin data in LA and generate the trajectories using Google Maps API by randomly selecting the pick-up and drop-out locations from the checkins.

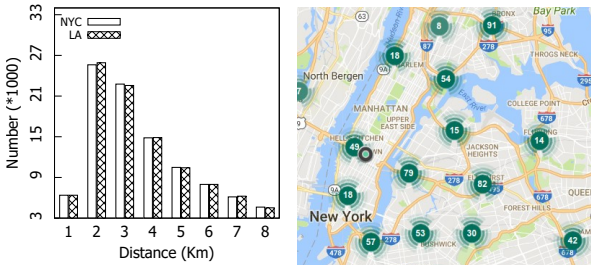
The statistics of those datasets are shown in Table 2, the distribution of trajectories' distance is shown in Figure 4a, and a snapshot of the billboards' locations in NYC is shown in Figure 4b.

Algorithms. Recall Sections 1 and 2.2, this is the first work studying the TIP problem, there is no previous work for direct comparison. In particular, we compare five methods: TrafficVol which picks billboards by a descending order of the volume of trajectories meeting those billboards within budget L; a basic greedy GreedySel (Algorithm 1); a Greedy Enumeration method EnumSel (Algorithm 2); our partition based method PartSel (Algorithm 3); our lazy probe method LazyProbe (Algorithm 4). Note that EnumSel is too slow to converge in 170 minutes even for a small dataset (because the complexity of EnumSel is proportional to $|U|^5$). Thus in our default setting we do not include EnumSel, but evaluation of EnumSel on a smaller subset of NYC is at our technical report [26].

Performance Measurement. For each method we evaluate the runtime and the influence value of the selected billboards. Each experiment is repeated 10 times, and the average result is reported.

Billboard costs. All advertising companies do not provide the exact leasing cost; instead, they provide a range of costs for a suburb. E.g., the costs of billboards in Long Island by LAMAR range from 2,500 to 14,000 for 4 weeks [1]. So we generate the cost of a billboard b by designing a function proportional to the number of trajectories influenced by b: $w(b) = \lfloor \beta \cdot I(b)/100 \rfloor \times 1000$, where β is a factor chosen from 0.8 to 1.2 randomly to simulate various cost/benefit ratios. Here we compute the cost w.r.t. $|\mathcal{T}| = 200k$ trajectories.

[7]http://www.nyc.gov/html/tlc/html/about/trip_record_data.shtml
[8]https://sites.google.com/site/yangdingqi/home
[9]https://developers.google.com/maps/



(a) Trajectory Distance (b) Billboard Location

Figure 4: Distribution of Datasets in NYC

Choice of influence probability $pr(b_i, t_j)$. In Section 2.1, we define two choices to compute $pr(b_i, t_j)$. By default, we use the first one. The result of the second one is in our technical report [26].

Parameters. Table 3 shows the settings of all parameters, and the default one is highlighted in bold. In all experiments, we vary one parameter while the rest are kept default, unless specified otherwise. Since the total number of real-world billboards in LAMAR is limited (see Table 2), the $|U|$ larger than the limit are replicated by randomly selecting locations in the two cities.

Setup. All codes are implemented in Java. Experiments are conducted on a server with 2.3 GHz Intel Xeon 24 Core CPU and 256GB memory running Debian/4.0 OS.

6.2 Choice of θ-partition

Since θ-partition is an input of PartSel method and θ indicates the degree of overlap among clusters generated in the partition phase of PartSel (and LazyProbe), we would like to find a generally good choice of θ that strikes a balance between the efficiency and effectiveness of PartSel and LazyProbe.

We vary θ from 0 to 0.4, and record the number of clusters as input of PartSel and LazyProbe methods, the percentage of the largest cluster size over U (i.e., $\frac{|C_m|}{|U|}$), the runtime and the influence value of PartSel and LazyProbe. The results on both datasets are shown in Figure 5 and Table 4. Note that EnumSel is too slow, so we do not include it.

Table 4: The $|C_m|/|U|$ Ratio w.r.t. Varying θ

	0.1	0.2	0.3	0.4
NYC	12.6%	7.1%	6.4%	5.8%
LA	13.5%	7.8%	5.9%	5.1%

By linking these results, we have four observations. (1) With the increase of θ, the influence quality decreases and the efficiency is improved, because for a larger θ, the tolerated influence overlap is larger and there are many more clusters with larger overlaps. (2) When θ is 0.1 and 0.2, PartSel and LazyProbe achieve the best influence (Figures 5a and 5c), while their efficiencies are not much worse than that of θ=0.3 (Figures 5b and 5d). The reason is that, it results in an appropriate number of clusters (e.g., 23 clusters for NYC dataset at θ=0.2 in Figure 5e), and the largest cluster covers 9.1% of all billboards, as evidenced by the value of $\frac{|C_m|}{|U|}$ in Table 4. (3) In an extreme case that θ=0.4, although the generated clusters are dispersed and small, it results in high overlaps among clusters, so the influence value drops and becomes worse than GreedySel, and meanwhile the efficiency of PartSel (LazyProbe) only improves by around 12 (6) times as compared to that of θ=0.2 on the NYC (LA) dataset. This is because PartSel and LazyProbe find influential billboards within a cluster and do not consider the influence overlap to billboards in other cluster. (4) All other methods beat the TrafficVol baseline by 45% in term of the influence value of selected billboards.

The result on LA is very similar to that of NYC. Therefore, we choose 0.2 as the default value of θ in the rest of the experiments.

6.3 Effectiveness Study

We study how the influence is affected by varying the budget L, trajectory number $|\mathcal{T}|$, distance threshold λ and overlap ratio respectively. Last we study the approximation ratio of all algorithms.

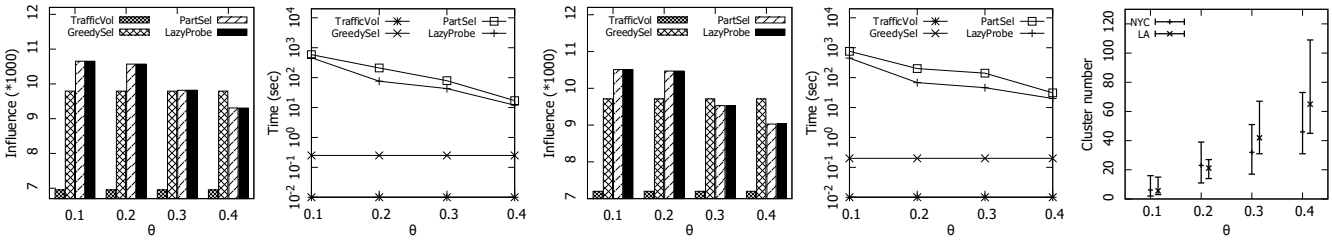
6.3.1 Varying the budget L. The influence of all algorithms on NYC and LA by varying L is shown in Figure 6, and we have the following observations. (1) TrafficVol has the worst performance. PartSel and LazyProbe achieve the same influence. The improvement of PartSel and LazyProbe over TrafficVol exceeds 99%. (2) With the growth of L, the advantage of PartSel and LazyProbe over GreedySel is increasing, from 1.8% to 6.5% when L varies from 100k to 300k on LA. This is because when the influence overlaps between clusters is unavoidable, how to maximize the benefit/cost ratio in clusters is critical to the performance. PartSel and LazyProbe can exactly achieve it by exploiting the locality feature within clusters.

6.3.2 Varying the trajectory number $|\mathcal{T}|$. Figure 7 shows the result by varying $|\mathcal{T}|$. We find: (1) the influence of all methods increases because more trajectories can be influenced; (2) the influence by PartSel and LazyProbe is consistently better than that of GreedySel and TrafficVol, because the trajectory locality is an important factor that should be considered to increase the influence.

6.3.3 Varying the threshold λ. Figure 8 shows the influence by varying λ, which determines the influence relationship between billboards and trajectories (Definition 2.1). With the increase of λ, the performance of all algorithms becomes better, as a single billboard can influence more trajectories. Moreover, PartSel and LazyProbe have the best performance and outperform the GreedySel baseline by at least 8%, because the enumerations can easily find influential billboards when the influence overlap becomes larger.

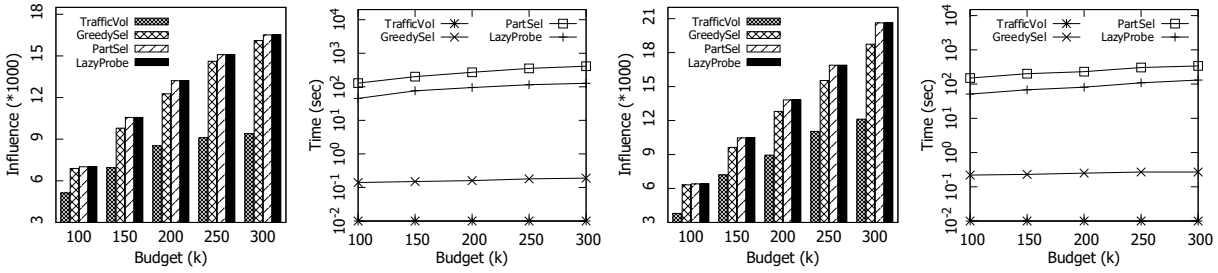
6.3.4 Additional Discussion. We also compared our solution with a meta heuristic algorithm, Simulated Annealing (Annealing), to verify the practical effectiveness. Although Annealing is costly and provides no theoretical bound, it has been proved to be able to find a near optimal solution for most optimization problems [18]. Since Annealing is a random search algorithm and its performance is not stable, we run it 50 times for each instance and select the best solution as our baseline. Table 5 reports both the influence value and its relative improvement w.r.t. Annealing for three different choices of budget L.

We find: (1) PartSel and LazyProbe have a very close performance to EnumSel in average. This is because when the overlap between clusters is small, each billboard selected by PartSel and LazyProbe is



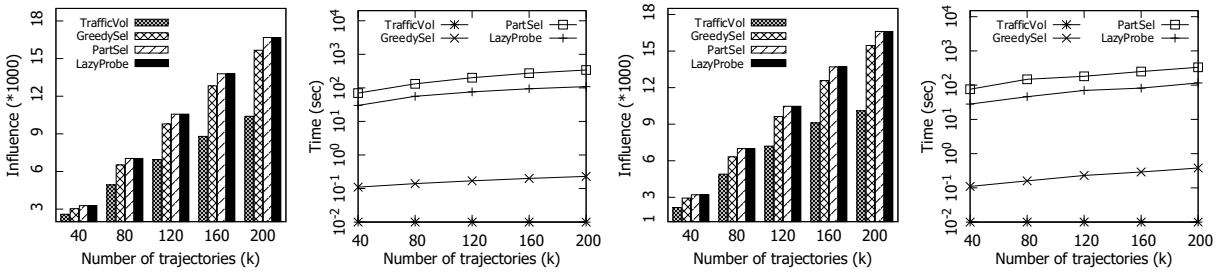
(a) Influence (NYC) (b) Efficiency (NYC) (c) Influence (LA) (d) Efficiency (LA) (e) Number of clusters

Figure 5: Effect of varying θ



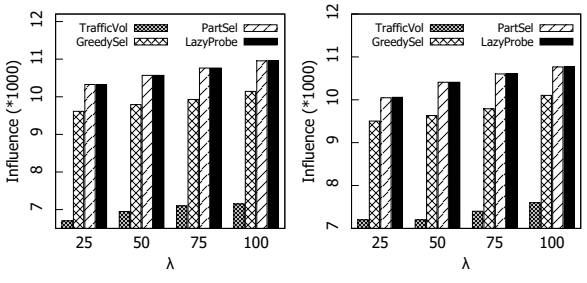
(a) Influence (NYC) (b) Efficiency (NYC) (c) Influence (LA) (d) Efficiency (LA)

Figure 6: Effect of varying budget L



(a) Influence (NYC) (b) Efficiency (NYC) (c) Influence (LA) (d) Efficiency (LA)

Figure 7: Effect of varying trajectory number $|\mathcal{T}|$



(a) NYC (b) LA

Figure 8: Effect of varying λ

less likely to overlap with the billboards in other clusters, and thus the performance of PartSel and LazyProbe would not lose much accuracy. As discussed in Section 6.4, EnumSel is very slow to work in practice. (2) PartSel and LazyProbe improve the influence by 6.6% in average as compared to Annealing. (3) TrafficVol which simply uses the traffic volume to select billboards has the worst performance.

Table 5: Additional test on NYC

	L=100k		L=200k		L=300k	
Annealing	6805	0.00%	11777	0.00%	15773	0.00%
TrafficVol	5111	-24.89%	8520	-27.66%	9400	-40.40%
GreedySel	6890	1.25%	12267	5.56%	16108	2.12%
EnumSel	7080	4.04%	13161	11.75%	16570	5.05%
PartSel	7013	3.06%	13215	12.21%	16512	4.69%
LazyProbe	7013	3.06%	13215	12.21%	16512	4.69%

6.4 Efficiency Study

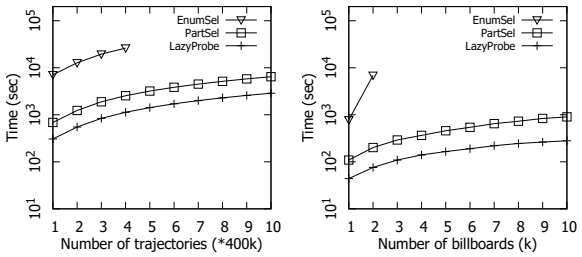
6.4.1 Varying the budget L. Figures 6b and 6d present the efficiency results when L varies from 100k to 300k. As EnumSel is too slow to complete in 10^4 seconds (because the complexity of EnumSel is proportional to $|U|^5$), we omit it. We have three observations regarding the running time of each method. (1) LazyProbe consistently beats PartSel by almost 3 times. (2) The time gap between GreedySel and PartSel becomes more significant with the increase of L. The reason is, PartSel has to invoke EnumSel L times for each cluster to construct the local solution, so the runtime of PartSel

grows quickly when L increases. (3) TrafficVol is the fastest one as it simply adopts a benefit-based selection.

6.4.2 Varying the trajectory number $\mathcal{T}$. Figure 7b and Figure 7d show the runtime of all algorithms on NYC and LA datasets. We observe that PartSel and LazyProbe scale linearly w.r.t. $\mathcal{T}$ which is consistent with our time complexity analysis; moreover, LazyProbe is around 4 times faster than PartSel.

6.5 Scalability Study

In this experiment we evaluate the scalability of EnumSel, PartSel and LazyProbe, by varying $|\mathcal{T}|$ (from 400k to 4M) and $|U|$ (from 1k to 10k). Since the effectiveness of GreedySel is not satisfying (Section 6.3), we omit it. The results are shown in Figure 9a and Figure 9b. We can see that LazyProbe scales very well and outperforms PartSel by 4-6 times. This is because even if the number of billboards is large, LazyProbe does not need to compute all local solutions for each cluster with different budgets, while it still can prune a large number of insignificant computation. Since EnumSel takes more than 10,000 seconds when the billboard number $|U|$ is larger than 2k, its result is omitted in the Figure for readability reason. It also shows that EnumSel has serious issues in efficiency making it impractical in real-world scenarios, while PartSel and LazyProbe scale well and can meet the efficiency requirement.



(a) Varying $|\mathcal{T}|$ **(b) Varying $|U|$**

Figure 9: Scalability test of our methods on NYC dataset

Summary. (1) Our methods EnumSel, PartSel and LazyProbe achieve much higher influence value than existing techniques (GreedySel, TrafficVol, and Annealing). (2) PartSel and LazyProbe achieve similar influence with EnumSel, but EnumSel is too slow to work in practice while LazyProbe and PartSel are much faster and can meet the efficiency requirement on large datasets.

7 CONCLUSION

We studied the problem of trajectory-driven influential billboard placement under a budget constraint and non-uniform billboard cost. We showed that it is NP-hard, and first proposed a greedy method with enumeration technique. We then exploited the locality property of the billboard influence and proposed a partition-based framework PartSel which significantly reduced the computation cost. We further devised a lazy probe method to prune billboards with low benefit/cost ratio. Lastly we conducted experiments on real datasets to verify the efficiency and effectiveness of our method. In future, we plan to investigate the scenario where the ads are dynamically updated [12, 25].

Acknowledgment. Zhiyong Peng was supported by the Ministry of Science and Technology of China (2016YFB1000700), and National Key Research & Development Program of China (2018YFB1003400). Zhifeng Bao was supported by ARC (DP170102726, DP180102050), NSFC (61728204, 91646204), and was a recipient of Google Faculty Award. Guoliang Li was supported by the 973 Program of China (2015CB358700), NSFC (61632016, 61472198, 61521002, 61661166012) and TAL education.

REFERENCES

[1] http://apps.lamar.com/demographicrates/content/salesdocuments/nationalratecard.xlsx.
[2] C. Borgs, M. Brautbar, J. Chayes, and B. Lucier. Maximizing social influence in nearly optimal time. In *SODA*, pages 946–957, 2014.
[3] S. Chen, J. Fan, G. Li, J. Feng, K. Tan, and J. Tang. Online topic-aware influence maximization. *PVLDB*, 8(6):666–677, 2015. URL http://www.vldb.org/pvldb/vol8/p666-chen.pdf.
[4] W. Chen, C. Wang, and Y. Wang. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *SIGKDD*, pages 1029–1038. ACM, 2010.
[5] F. M. Choudhury, J. S. Culpepper, Z. Bao, and T. Sellis. A general framework to resolve the mismatch problem in XML keyword search. *VLDB J.*, 2018. doi: 10.1007/s00778-018-0504-y.
[6] L. Guo, D. Zhang, G. Cong, W. Wu, and K.-L. Tan. Influence maximization in trajectory databases. *IEEE Transactions on Knowledge and Data Engineering*, 29 (3):627–641, 2017.
[7] D. Kempe, J. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. In *SIGKDD*, pages 137–146. ACM, 2003.
[8] S. Khuller, A. Moss, and J. S. Naor. The budgeted maximum coverage problem. *Information Processing Letters*, 70(1):39–45, 1999.
[9] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance. Cost-effective outbreak detection in networks. In *SIGKDD*, pages 420–429. ACM, 2007.
[10] G. Li, S. Chen, J. Feng, K.-l. Tan, and W.-s. Li. Efficient location-aware influence maximization. In *SIGMOD*, pages 87–98. ACM, 2014.
[11] Y. Li, D. Zhang, and K.-L. Tan. Real-time targeted influence maximization for online advertisements. *PVLDB*, 8(10):1070–1081, 2015.
[12] Y. Li, D. Zhang, Z. Lan, and K.-L. Tan. Context-aware advertisement recommendation for high-speed social news feeding. In *ICDE*, pages 505–516. IEEE, 2016.
[13] Y. Li, J. Fan, D. Zhang, and K.-L. Tan. Discovering your selling points: Personalized social influential tags exploration. In *SIGMOD*, pages 619–634. ACM, 2017.
[14] Y. Li, J. Fan, Y. Wang, and K.-L. Tan. Influence maximization on social graphs: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 2018.
[15] D. Liu, D. Weng, Y. Li, J. Bao, Y. Zheng, H. Qu, and Y. Wu. Smartadp: Visual analytics of large-scale taxi trajectories for selecting billboard locations. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):1–10, 2017.
[16] Y. Liu, R. C.-W. Wong, K. Wang, Z. Li, C. Chen, and Z. Chen. A new approach for maximizing bichromatic reverse nearest neighbor search. *Knowledge and Information Systems*, 36(1):23–58, Jul 2013.
[17] F. Murtagh. A survey of recent advances in hierarchical clustering algorithms. *The Computer Journal*, 26(4):354–359, 1983.
[18] E.-G. Talbi. *Metaheuristics: from design to implementation*, volume 74. John Wiley & Sons, 2009.
[19] Y. Tang, X. Xiao, and Y. Shi. Influence maximization: Near-optimal time complexity meets practical efficiency. In *SIGMOD*, pages 75–86. ACM, 2014.
[20] D. Wagner and F. Wagner. *Between Min Cut and Graph Bisection*, pages 744–750. Springer, Berlin, Heidelberg, 1993.
[21] S. Wang, Z. Bao, J. S. Culpepper, T. Sellis, M. Sanderson, and X. Qin. Answering top-k exemplar trajectory queries. In *ICDE*, pages 597–608. IEEE, 2017.
[22] S. Wang, Z. Bao, J. S. Culpepper, T. Sellis, and G. Cong. Reverse k nearest neighbor search over trajectories. *IEEE Trans. Knowl. Data Eng.*, 30(4):757–771, 2018.
[23] S. Wang, Z. Bao, J. S. Culpepper, Z. Xie, Q. Liu, and X. Qin. Torch: A search engine for trajectory data. In *SIGIR*. ACM, 2018.
[24] R. C.-W. Wong, M. T. Özsu, P. S. Yu, A. W.-C. Fu, and L. Liu. Efficient method for maximizing bichromatic reverse nearest neighbor. *PVLDB*, 2(1):1126–1137, 2009.
[25] D. Zhang, Y. Li, J. Fan, L. Gao, F. Shen, and H. T. Shen. Processing long queries against short text: Top-k advertisement matching in news stream applications. *ACM Transactions on Information Systems (TOIS)*, 35(3):28, 2017.
[26] P. Zhang, Z. Bao, Y. Li, G. Li, Y. Zhang, and Z. Peng. Trajectory-driven influential billboard placement. *CoRR*, abs/1802.02254, 2018.
[27] Z. Zhou, W. Wu, X. Li, M. L. Lee, and W. Hsu. Maxfirst for maxbrknn. In *ICDE*, pages 828–839. IEEE, 2011.