AN EFFICIENT SAMPLING METHOD FOR CHARACTERIZING POINTS OF INTERESTS ON MAPS

Mr. Sai Krishna Bonagala1, Ms. Sk.Ayisha Begum[2*]

1 Final Year MCA Student, QIS College of Engineering and Technology, Ongole

2* Assistant Professor, MCA Dept., QIS College of Engineering and Technology, Ongole

Abstract: *As of late guide administrations (e.g., Google maps) and area based online informal organizations (e.g., Foursquare) pull in a ton of consideration and organizations. With the expanding ubiquity of these area based administrations, investigating and portraying point of interests (PoIs), for example, eateries and inns on maps gives significant data to applications, for example, start up advertising research. Because of the absence of a direct completely access to PoI databases, it is infeasible to comprehensively pursuit and gather all PoIs inside a huge region utilizing open APIs, which more often than not force a point of confinement on the most extreme inquiry rate. In this paper, we propose sampling techniques to precisely assess PoI measurements, for example, sum and average sum from as few questions as could be expected under the circumstances. Experimental results dependent on genuine datasets demonstrate that our strategies are productive, and require multiple times less questions than best in class techniques to accomplish a similar precision.*

Keywords: Point of interest, Sampling, Measurements.

I. INTRODUCTION

Total measurements (e.g., entirety, normal, and appropriation) of purposes of interests (PoIs), e.g., eateries and lodgings on map administrations, for example, Google maps [2] and Foursquare [3], give important data to applications, for example, showcasing basic leadership. For instance, the learning of the PoI rating conveyance empowers us to assess a specific PoI's [1], relative administration quality positioning. In addition, an eatery start-up can gather sustenance inclinations of individuals in a geographic zone by looking at the fame of eatery PoIs serving distinctive cooking styles inside the zone of intrigue [4].

In the mean time, it can likewise appraise its market measure dependent on PoI total measurements, for example, the quantity of foursquare clients checked in PoIs inside the region. Correspondingly, a lodging start-up can use lodging PoIs' properties, for example, evaluations also, audits to comprehend its market and rivals. To precisely compute the above total measurements, it requires recovering all PoIs inside the zone of intrigue.

Anyway most guide specialist organizations don't give the open with a direct completely access to their PoI databases, so we can just depend on open guide APIs to investigate and gather PoIs. Besides, open APIs more often than not force confines on the greatest inquiry rate and the most extreme number of PoIs returned in a reaction to a question; in this manner it is exorbitant to gather PoIs inside an expansive region. For instance, foursquare map API [5] returns up to 50 PoIs for every inquiry and it permits 500 questions for each hour per account. To gather PoIs inside 14 urban communities in Foursquare, Li et al. [6] went through right around two months utilizing 40 machines in parallel.

To address the above test, inspecting is required. That is, a little division of PoIs are tested and used to ascertain PoI measurements. Because of the absence of a direct completely access to PoI databases, one can't test over PoIs in a direct way, so it is difficult to test PoIs consistently. The existing examining techniques [7], [8] have been demonstrated to test PoIs with predispositions. Subsequent to testing a small amount of PoIs utilizing these two strategies, one has no ensures whether the PoI insights acquired straightforwardly are to be trusted. To tackle this issue, Dalvi et al. [7] propose a technique to address the inspecting inclination. Anyway the technique is expensive in light of the fact that it requires countless for each inspected PoI (e.g., all things considered 55 inquiries are utilized in their paper). The technique in [8] tests PoIs with obscure predisposition, so it is hard to evacuate its testing predisposition. In this work we propose another strategy random reason zoom-in (RRZI) to dispose of the estimation inclination. The essential thought behind RRZI is to test [9], a lot of sub-locales from a territory of enthusiasm aimlessly and after that gather PoIs inside inspected locales. Be that as it may, when we question an inspected sub region counting an expansive number of PoIs, an obscure inspecting inclination [10], is presented on the off chance that we just gather PoIs returned.

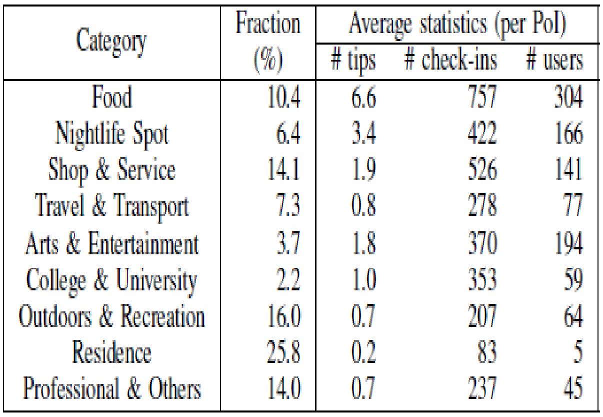
Else, we have to additionally isolate the examined sub region to thoroughly gather all PoIs inside it. It requires a substantial number of inquiries. To take care of this issue [11], we isolate the zone of enthusiasm into completely open sub-locales without covering, where an area is characterized as a completely open area in the event that it incorporates PoIs not exactly the greatest [12], number of PoIs returned for an inquiry. At that point it is proficient to gather PoIs inside a tested sub-locale, which requires only one question [13]. To test a completely available area, RRZI fills in as pursues:

From a predetermined zone, RRZI isolates the current questioned district into two sub-areas without covering, and after that haphazardly chooses a non-void sub-area as the following district to question. It rehashes this procedure until it watches a completely open area [14]. We demonstrate that RRZI is effective, and it requires just a couple of inquiries to test a completely available area. Other than its productivity, the inspecting inclination of RRZI is anything but difficult to be amended, which requires no additional inquiries in correlation with the current strategies [7], [8]. To further decrease the quantity of inquiries, we propose a blend technique RRZI URS, which first picks a little sub-district from the territory of enthusiasm indiscriminately and afterward tests PoIs inside the sub-locale utilizing RRZI. Additionally, for guide administrations such as Google maps giving the absolute number of PoIs inside an information look locale, we propose a strategy to improve the exactness [15], of RRZI by using this Meta data. We perform tests utilizing an assortment of genuine datasets, and demonstrate that our techniques drastically decrease the quantity of questions required to accomplish a similar estimation precision of best in class strategies.

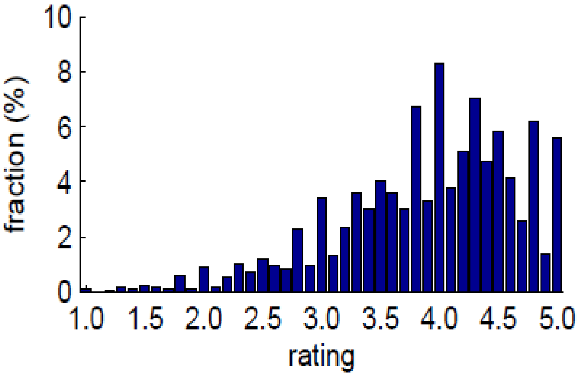
II. RELATED WORK

Recently heaps of attention has been paid to check hidden databases victimisation public search interfaces. Previous work focuses on creeping, retrieving, and mining data from web search engines [16]–[19], text-based databases [20] and form-based [21] databases. Many sampling methods ar given in [18], [22], to estimate a formbased hidden database's size (i.e., the amount of tuples, refer to Khelghati et al. [17], for a decent survey). These methods ar designed for search engines with inputs fixed as categorical information, and their performances depend upon (Our real application on Foursquare) Statistics of PoIs within US. Category Fractio

(Our real application on Foursquare) Statistics of PoIs within US.

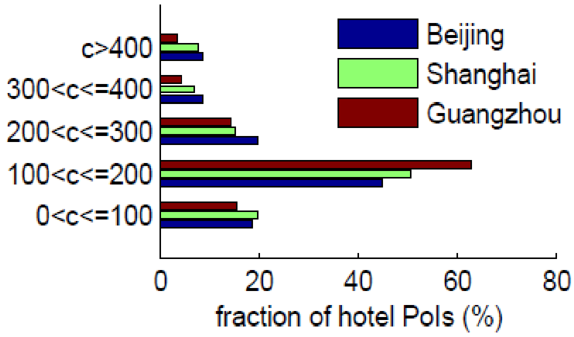


Category	Fraction (%)	Average statistics (per PoI)		
		# tips	# check-ins	# users
Food	10.4	6.6	757	304
Nightlife Spot	6.4	3.4	422	166
Shop & Service	14.1	1.9	526	141
Travel & Transport	7.3	0.8	278	77
Arts & Entertainment	3.7	1.8	370	194
College & University	2.2	1.0	353	59
Outdoors & Recreation	16.0	0.7	207	64
Residence	25.8	0.2	83	5
Professional & Others	14.0	0.7	237	45



(Our real application on Google maps)

Rating distribution of food-type PoIs inside North American nation. the vary of input values, so that they can not be directly applied to sample PoIs exploitation map-based search engines, which have a quite sizable amount of input values (latitude and line of longitude pairs inside the world of interest). to deal with this challenge, two strategies in [7] and [8] ar given to sample PoIs exploitation public map arthropod genus. Next we tend to discuss them well.



(Our real application on Baidu maps) Distribution of *c*, hotel-type PoIs' prices per room per night.

III EXISTING SYSTEM

The existing sampling methods have been proved to sample PoIs with biases. After sampling a fraction of PoIs using these two methods, one has no guarantees whether the PoI statistics obtained directly are to be trusted. Besides its efficiency, the sampling bias of RRZI is easy to be corrected, which requires no extra queries in comparison with the existing methods. We can see that there exist three fully accessible regions a, b, and c, which could be observed and sampled by RRZI. The probabilities of sampling a, b, and c are 1/2, 1/4, and 1/4 respectively. We expect that the most efficient method is RRZIC MHWRS when there exists a publicly available API with meta information (i.e., the total number of PoIs within an input search region) returned for a query, and RRZI URS otherwise, which is validated by our experiments later.

Disadvantages:

While such a sampling methods search interface is often sufficient for an individual user looking for the nearest shops or restaurants, data analysts and researchers interested in an LBS service often desire a more comprehensive view of its underlying data. For example, an analyst of the fast-food industry may be interested in obtaining a list of all McDonald's restaurants in the world, so as to analyze their geographic coverage, correlation with income levels reported in Census, etc. Our objective in this paper is to enable the crawling of an LBS database by issuing a small number of queries through its publicly available kNN web search interface, so that afterwards a data analyst can simply treat the crawled data as an offline database and perform whatever analytics operations desired.

IV PROPOSED SYSTEM

Propose sampling methods to accurately estimate PoI statistics such as sum and average aggregates from as few queries as possible. Experimental results based on real datasets show that our methods are efficient, and require six times less queries than state-of-the-art methods to achieve the same accuracy. Propose a method to correct the sampling bias. However the method is costly because it requires a large number of queries for each sampled PoI (e.g., on average 55 queries are used in their paper). The method in samples PoIs with unknown bias, so it is difficult to remove its sampling bias. we propose a new method random region zoom-in (RRZI) to eliminate the estimation bias. The basic idea behind RRZI is to sample a set of sub-regions from an area of interest at random and then collect PoIs within sampled regions.

Our aim is to estimate aggregate statistics (e.g., sum, average, and distribution) of PoIs' attributes. Formally, let A be the area of interest. Denote by P the set of PoIs within

A. For example, P can be the set of hotels within A. We want to estimate the following statistics over P.

1) *Sum aggregate*. For any function

f : P → R, whereR is the set of real numbers, the sum aggregate is defined as *fs*(P) =Σ*p*∈P *f*(*p*). If *f*(*p*) is the number of rooms a hotel *p* has, then *fs*(P) corresponds to the total number of hotel rooms within A. If *f*(*p*) is the constant function *f*(*p*) = 1, then *fs*(P) corresponds to |P|, the number of hotels within A.

2) *Average aggregate*. For any function

f : P →R, the average aggregate is defined as *fa*(P) =1|P|Σ*p*∈P *f*(*p*). If *f*(*p*) is the price per room per night for a hotel *p*, then *fa*(P) corresponds to the average price for hotels within A.

3) *PoI distribution*. Let *L*(*p*) be the label of a PoI *p* specifying a certain property of *p*. For example, *L*(*p*) can be the star rating of a hotel *p*. Denote the range of PoI labels as *{l*1*; : : : ; lJ}*. Let _ = (_1*; : : : ;* _*J*) be the distribution of a set of PoIs, where _*j* (1 ≤ *j* ≤ *J*) is the fraction of PoIs with label *lj* . Formally,_*j* = 1|P|Σ *p*∈P **1**(*L*(*p*) = *lj*)*;* 1 ≤ *j* ≤ *J*, where **1**(*L*(*p*) = *lj*) is the indicator function that equals one when predicate *L*(*p*) = *lj* is true, and zero otherwise. If *L*(*p*) is the star rating of *p*, then is the star rating distribution of hotels within A.

As eluded before, we focus on designing *sampling* methods to accurately estimate the above statistics from as few queries as possible.

Advantages:

A small fraction of PoIs are sampled and used to calculate PoI statistics. Due to the lack of a direct fully access to PoI databases, one cannot sample over PoIs in a direct manner, so it is hard to sample PoIs uniformly.

V SYSTEM COMPONENTS

There are three different components proposed in this system as

1. Points Of Interests:-

Popularity of these location-based services, exploring and characterizing points of interests (PoIs) such as restaurants and hotels on maps provides valuable information for applications such as startup marketing research. Points of interests (PoIs), e.g., restaurants and hotels on map services such as Google maps and Foursquare provide valuable information for applications such as marketing decision making. For example, the knowledge of the PoI rating distribution enables us to evaluate a particular PoI's relative service quality ranking. Moreover, a restaurant start-up can infer food preferences of people in a geographic area by comparing the popularity of restaurant PoIs serving different cuisines within the area of interest.

2. Sampling:-

Sampling Methods to accurately estimate PoI statistics such as sum and average aggregates from as few queries as possible. Experimental results based on real datasets show that our methods are efficient, and require six times less queries than state-of-the-art methods to achieve the same accuracy. To address the above challenge, sampling is required. That is, a small fraction of PoIs are sampled and used to calculate PoI statistics. Results for Foursquare datasets are similar, which are omitted here. In summary, the above straightforward sampling method is not easy to be implemented, so designing accurate and efficient sampling methods for estimating PoI statistics is a much challenging task.

3. Measurement

The sampling bias might introduce large errors into the measurement of PoI statistics. To solve this problem, we use a counter to record the probability of sampling a region from A, which is used to correct the sampling bias later. is initialized with 1, and updated as follows: At each step, we set = =2 if both 0(Q) and 1(Q) are non-empty, otherwise keeps unchanged. Finally records the probability of sampling a fully accessible sub-region from A.

ALGORITHM

Random Region Zoom-in:-

We propose a new method random region zoom-in (RRZI) to eliminate the estimation bias. The basic idea behind RRZI is to sample a set of sub-regions from an area of interest at random and then collect PoIs within sampled regions. Besides its efficiency, the sampling bias of RRZI is easy to be corrected, which requires no extra queries in comparison with the existing methods. To further reduce the number of queries, we propose a mix method RRZI URS, which first picks a small sub-region from the area of interest at random and then samples PoIs within the sub- region using RRZI.

$$\begin{cases} \chi_0(Q) = [(x_{SW}, y_{SW}), (\lceil \frac{x_{SW}+x_{NE}}{2\delta} \rceil \delta - \delta, y_{NE})] \\ \chi_1(Q) = [(\lceil \frac{x_{SW}+x_{NE}}{2\delta} \rceil \delta, y_{SW}), (x_{NE}, y_{NE})]. \end{cases} \qquad (1)$$

Random Region Zoom-in with Count dataIn this section we have a tendency to propose a technique, named random region zoom-in with count data (RRZIC), to further improve the accuracy of RRZI for map services like Google maps, wherever results from a question embody the number of PoIs inside the input search region. Compared to RRZI, RRZIC tends to sample PoIs uniformly, giving us smaller estimation errors for PoIs statistics. The pseudocode RRZIC (A) is shown as algorithmic program a pair of within the Appendix. Initially we have a tendency to set Q = A.

and estimate $\theta = (\theta_1, \ldots, \theta_J)$ as

$$\bar{\theta}_j = \frac{1}{m} \sum_{i=1}^{m} \sum_{p \in P(r_i)} \frac{1(L(p) = l_j)}{n(r_i)}, \qquad 1 \leq j \leq J. \quad (6)$$

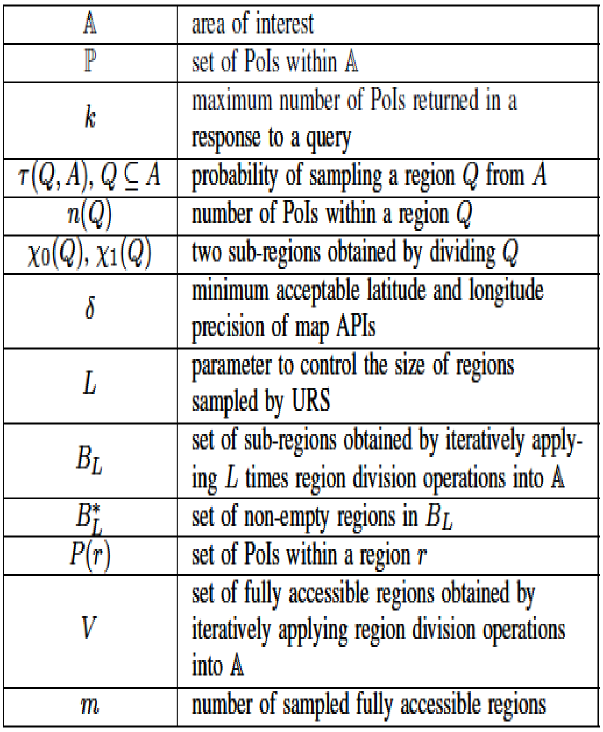
Their variances are

$$Var(\bar{f}_s(\mathbb{P})) = \frac{1}{m} \left(\sum_{r \in V} \frac{\left(\sum_{p \in P(r)} f(p) \right)^2 n(\mathbb{A})}{n(r)} - f_s^2(\mathbb{P}) \right)$$

$$Var(\bar{\theta}_j) = \frac{1}{m} \left(\sum_{r \in V} \frac{\left(\sum_{p \in P(r)} 1(L(p) = l_j) \right)^2}{n(r) n(\mathbb{A})} - \theta_j^2 \right).$$

Random Region Zoom-in Count:-

In this section, we tend to gift our sampling strategies to estimate oI mixture statistics outlined in Section two. We first propose a random region zoom-in (RRZI) methodology to sample PoIs inside a section A of interest, and provide our estimators of dish statistics. to enhance the accuracy of RRZI, we tend to then propose a technique RRZIC by utilizing the meta data (i.e., the entire variety of PoIs inside associate degree input search region) came for a question , that is provided by map services like Google maps. To more cut back the amount of queries of RRZI and RRZIC needed, we tend to propose combine strategies RRZI URS and RRZIC URS, that initial choose alittle sub-region from A indiscriminately and so sample PoIs inside the sub-region exploitation RRZI and RRZIC severally. Finally, we tend to show that RRZIC URS would possibly exhibit larger errors than RRZIC for estimating dish statistics. to unravel this downside, we tend to propose a technique RRZIC MHWRS to extend the accuracy of RRZIC URS. For simple reading, we tend to list notations used throughout the paper.



$\mathbb{A}$	area of interest
$\mathbb{P}$	set of PoIs within $\mathbb{A}$
k	maximum number of PoIs returned in a response to a query
$\tau(Q, A), Q \subseteq A$	probability of sampling a region Q from A
$n(Q)$	number of PoIs within a region Q
$\chi_0(Q), \chi_1(Q)$	two sub-regions obtained by dividing Q
δ	minimum acceptable latitude and longitude precision of map APIs
L	parameter to control the size of regions sampled by URS
B_L	set of sub-regions obtained by iteratively applying L times region division operations into $\mathbb{A}$
B_L^*	set of non-empty regions in B_L
$P(r)$	set of PoIs within a region r
V	set of fully accessible regions obtained by iteratively applying region division operations into $\mathbb{A}$
m	number of sampled fully accessible regions

Algorithm 1: RRZI($\mathbb{A}$) pseudo-code.

```
input : $\mathbb{A}$
/* $Q$ is a sub-region sampled from $\mathbb{A}$ at
   random, and $\tau$ records the probability of
   sampling $Q$ from $\mathbb{A}$.                    */
output: $Q$ and $\tau$
/* searchPoI($Q$) is the set of PoIs returned for
   querying the region $Q$                             */
$Q \leftarrow \mathbb{A}$, $\tau \leftarrow 1$, $l \leftarrow 0$, and $O \leftarrow$ searchPoI($Q$);
while $|O| = k$ do
    /* $\chi_0(Q)$ and $\chi_1(Q)$ are the two sub-regions of $Q$
       defined as (1) and (2)                          */
    $Q_0 \leftarrow \chi_0(Q)$ and $Q_1 \leftarrow \chi_1(Q)$;
    /* If $O$ includes no PoI in the region $Q_0/Q_1$,
       then emptyRegion($Q_0, Q_1, O$) returns 0/1.
       Otherwise, both $Q_0$ and $Q_1$ are non-empty, and
       emptyRegion($Q_0, Q_1, O$) returns -1           */
    $i \leftarrow$ emptyRegion($Q_0, Q_1, O$);
    if $i \neq -1$ then
        $O' \leftarrow$ searchPoI($Q_i$);
        if $|O'| = 0$ then
            $Q \leftarrow Q_{1-i}$;
        else
            /* random($Q_0, Q_1$) returns $Q_0$ and $Q_1$ at
               random                                  */
            $Q \leftarrow$ random($Q_0, Q_1$);
            $O \leftarrow O'$ and $\tau \leftarrow \tau/2$;
        end
    else
        $Q \leftarrow$ random($Q_0, Q_1$);
        $O \leftarrow$ searchPoI($Q$) and $\tau \leftarrow \tau/2$;
    end
end
```

Algorithm 2: RRZIC($\mathbb{A}$) pseudo-code.

```
input : $\mathbb{A}$
/* $Q$ is a sub-region sampled from $\mathbb{A}$ at
   random, and $\tau$ records the probability of
   sampling $Q$ from $\mathbb{A}$.                    */
output: $Q$ and $\tau$
/* countPoI($Q$) returns the number of PoIs in $Q$.   */
$Q \leftarrow \mathbb{A}$, $\tau \leftarrow 1$, and $z \leftarrow$ countPoI($Q$);
/* $k$ is the maximum number of PoIs returned in
   a response to a query.                              */
while $z > k$ do
    /* $\chi_0(Q)$ and $\chi_1(Q)$ are the two sub-regions of $Q$
       defined as (1) and (2).                         */
    $Q_0 \leftarrow \chi_0(Q)$ and $Q_1 \leftarrow \chi_1(Q)$;
    /* $z_0$ and $z_1$ are the numbers of PoIs within the
       regions $Q_0$ and $Q_1$ respectively.           */
    $z_0 \leftarrow$ countPoI($Q_0$) and $z_1 = z - z_0$;
    /* $U(0,1)$ is a random sample from $(0,1)$ .      */
    $u \leftarrow U(0,1)$;
    if $u < z_0/z$ then
        $Q \leftarrow Q_0$, $\tau \leftarrow \tau \times z_0/z$, and $z \leftarrow z_0$;
    else
        $Q \leftarrow Q_1$, $\tau \leftarrow \tau \times z_1/z$, and $z \leftarrow z_1$;
    end
end
```

Area Of Interest on Estimating

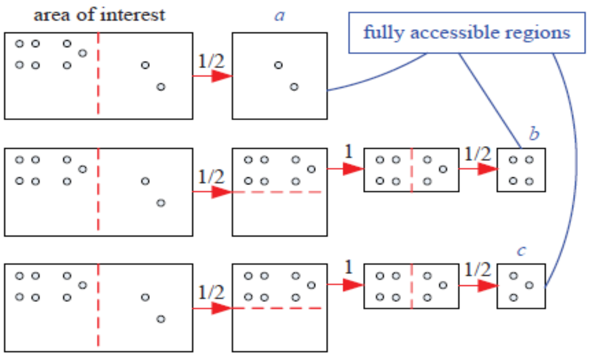
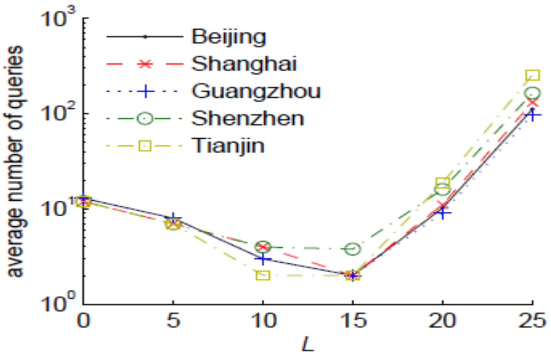


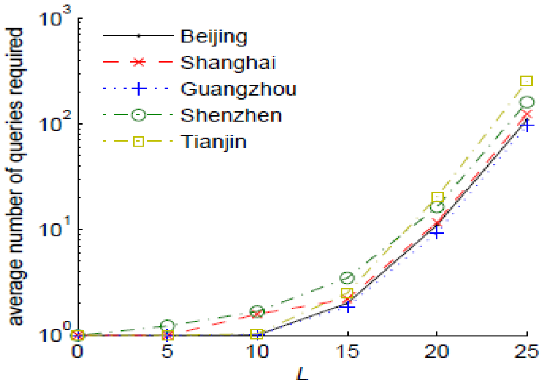
Fig: An example of applying RRZI into the area of interest, where k = 5. The number above a red arrow refers to the probability of selecting a sub-region of the current queried region to zoom in.

VI. RESULT

In this section, we tend to conduct experiments to guage the performance of our strategies for estimating PoIs' average and distribution statistics. For the Baidu dish datasets we used, PoIs area unit classified into differing kinds like restaurant, hotel, and looking. The numbers of restauranttpye PoIs area unit seventy five,255, 36,417, 24,353, 16,025, and 10,032 for datasets Peiping, Shanghai, Guangzhou, Shenzhen, and Tianjin severally. We tend to use these restaurant-type PoIs to generate benchmark datasets for our following experiments. We manually generate a value for every restauranttype PoI mistreatment 2 totally different value distribution schemes CDS UNI and CDS NOR. For CDS UNI, the price of a dish is uniformly designated from the vary (0, 300) indiscriminately. For CDS NOR, the price of a dish may be a positive variety every which way selected from (0, +∞) per a traditional distribution with mean one hundred fifty and stand deviation a hundred. We tend to additionally conduct experiments on foursquare datasets. Table half-dozen and show the $64000 values of the associated average statistics (i.e., the average numbers of check-ins, users, and tips) and dish distributions (i.e., the distributions of PoIs by the numbers of check-ins, users, and tips), that area unit of interest. 0



(Baidu maps) Average number of queries required to sample a fully accessible region for RRZI URS.



(Baidu maps) Average number of queries required to sample a non-empty sub-region for URS with different *L*.

VII. CONCLUSION

In this paper, we propose strategies to test PoIs on maps, and give reliable estimators of PoI total insights. We demonstrate that the blend strategy RRsZI URS is more precise than RRZI under a similar number of questions utilized. At the point when PoI check data is given by open APIs, RRZIC MHWRS using this Meta data is more precise than RRZI URS. The test results dependent on an assortment of genuine datasets demonstrate that our strategies are effective, and they pointedly lessen the quantity of questions required to accomplish a similar estimation precision of state-of-the-arts techniques.

VIII. REFERENCES

[1] P. Wang, W. He, and X. Liu, “An efficient sampling method for characterizing points of interests on maps,” in *Proceedings of IEEE ICDE 2014*, 2014.

[2]“Google maps,” https://maps.google.com/, 012.

[3]“Foursquare,” http://www.foursquare.com.

[4] Y. Zhu, J. Huang, Z. Zhang, Q. Zhang, T. Zhou, and Y. Ahn, “Geography and similarity of regional cuisines in china,” *arXiv preprint arXiv:1307.3185*, 2013.

[5]“search venues on foursqure,” https://developer.foursquare.com/docs/enues/search, 2013.

[6] Y. Li, M. Steiner, L. Wang, Z.-L. Zhang, and J. Bao, “Exploring venue popularity in foursquare,” in *The Fifth IEEE International Workshop on Network Science for Communication Networks*, 2013,pp. 1–6.

[7] N. Dalvi, R. Kumar, A. Machanavajjhala, and V. Rastogi, “Sampling hidden objects using nearest-neighbor oracles,” in *Proceedings of ACM SIGKDD 2011*, December 2011, pp. 1325–1333.

[8] Y. Li, M. Steiner, L. Wang, Z.-L. Zhang, and J. Bao, “Dissecting foursquare venue popularity via random region sampling,” in *Proceedings of the 2012 ACM conference on CoNEXT student workshop*, 2012, pp. 21–22.

[9] “Baidu maps,” http://map.baidu.com/, 2012.

[10] “Baidu poi datasets,” http://ishare.iask.sina.com.cn/f/34170612.html,2013.

[11] S. Chib and E. Greenberg, “Understanding the metropolis-hastings algorithm,” *The American Statistician*, vol. 49, no. 4, pp. 327–335, November 1995.

[12] W. K. Hastings, “Monte carlo sampling methods using markov chains and their applications,” *Biometrika*, vol. 57, no. 1, pp. 97–109, April 1970.

[13] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, “Equations of state calculations by fast computing machines,” *IEEE Journal on Selected Areas in Communications*,vol. 21, no. 6, pp. 1087–1092, June 2011.

[14] “search places on google maps,” https://developer.foursquare.com/docs/venues/search, 2013.

[15] “search places on baidu maps,” http://developer.baidu.com/map/lbs-geosearch.htm, 2013.

[16] P. Rusmevichientong, D. M. Pennock, S. Lawrence, and L. C. Giles, “Methods for sampling pages uniformly from

the world wide web," in *AAAI Fall Symposium on Using Uncertainty Within Computation*, November 2001, pp. 121–128.

[17] Z. Bar-Yossef and M. Gurevich, "Efficient search engine measurements," in *Proceedings of WWW 2007*, 2007, pp. 401–410.

[18] "Mining search engine query logs via suggestion sampling," *Proceedings of VLDB Endowment*, vol. 1, no. 1, pp. 54–65, Aug. 2008.

[19] M. Zhang, N. Zhang, and G. Das, "Mining a search engine's corpus: efficient yet unbiased sampling and aggregate estimation," in *Proceedings of SIGMOD 2011*, New York, NY, USA, 2011, pp.793–804.

[20] J. Callan and M. Connell, "Query-based sampling of text databases," *ACM Trans. Inf. Syst.*, vol. 19, no. 2, pp. 97–130, Apr. 2001.

[21] P. G. Ipeirotis and L. Gravano, "Distributed search over the hidden web: hierarchical database sampling and selection," in *Proceedings of VLDB 2002*, 2002, pp. 394–405.

[22] E. Agichtein, P. G. Ipeirotis, and L. Gravano, "Modeling query-based access to text databases," in *WebDB*, 2003, pp. 87–92.

Authors Profile

Mr. **Sai Krishna Bonagala** pursuing MCA 3rd year in Qis College and Engineering and Technology in Department of Master of Computer Applications, Ongol

Ms. **Sk. Ayisha Begum** is currently working as an Assistant Professor in Department of Master of Computer Applications in QIS College of Engineering & Technology with the Qualification MCA.