Efficiently Estimating Statistics of Points of Interests on Maps

D.TEJASWI
P.G.Scholor,Department of CSE,
St.Mary's Women's Engineering College,
Budampadu,Gutur, Andhra Pradesh.

SD.NAGULMEERA
Assistant Professor,Department of CSE,
St.Mary's Women's Engineering College,
Budampadu,Gutur, Andhra Pradesh.

Abstract

Recently map services (e.g., Google maps) and location-based online social networks (e.g., Foursquare) attract a lot of attention and businesses. With the increasing 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. Due to the lack of a direct fully access to PoI databases, it is infeasible to exhaustively search and collect all PoIs within a large area using public APIs, which usually impose a limit on the maximum query rate. In this paper, we 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.

Introduction

Aggregate statistics (e.g., sum, average, and distribution) of points of interests (PoIs), e.g., restaurants and hotels on map services such as Google maps [2] and Foursquare [3], 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 geo-graphic area by comparing the popularity of restaurant PoIs serving different cuisines within the area of interest [4]. Meanwhile, it can also estimate its market size based on PoI aggregate statistics, such as the number of Foursquare users checked in PoIs within the area.

Similarly, a hotel start-up can utilize hotel PoIs' properties such as ratings and reviews to understand its market and competitors.

To exactly calculate the above aggregate statistics, it requires to retrieve all PoIs within the area of interest. However most map service providers do not provide the public with a direct fully access to their PoI databases, so we can only rely on public map APIs to explore and collect PoIs. Moreover, public APIs usually impose limits on the maximum query rate and the maximum number of PoIs returned in a response to a query, therefore it is costly to collect PoIs within a large area. For example, Foursquare map API [5] returns up to 50 PoIs per query and it allows 500 queries per hour per account. To collect PoIs within 14 cities in Foursquare, Li et al. [6] spent almost two months using 40 machines in parallel.

To address the above challenge, sampling is required. That is, 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. The existing sampling methods [7], [8] 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. To solve this problem, Dalvi et al. [7] 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 [8] samples PoIs with unknown bias, so it is difficult to remove its sampling bias.

In this work 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. However, when we query a sampled sub- region including a large number of PoIs, an unknown sampling bias is introduced if we only collect PoIs returned. Otherwise, we need to further divide the sampled sub- region to exhaustively collect all PoIs within it. It requires a large number of queries. To solve this problem, we divide the area of interest into fully accessible sub-regions without overlapping, where a region is defined as a fully accessible region if it includes PoIs less than the maximum number of PoIs returned for a query.

Then it is efficient to collect PoIs within a sampled sub-region, which requires just one query. To sample a fully accessible region, RRZI works as follows: From a specified area, RRZI divides the current queried region into two sub-regions without overlapping, and then randomly selects a non-empty sub-region as the next region to query. It repeats this process until it observes a fully accessible region. We show that RRZI is efficient, and it requires only a few queries to sample a fully accessible region. Besides its efficiency, the sampling bias of RRZI is easy to be corrected, which requires no extra queries in comparison with the existing methods [7], [8]. 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.

Moreover, for map services such as Google maps providing the total number of PoIs within an input search region, we propose a method to improve the accuracy of RRZI by utilizing this meta information. We perform experiments using a variety of real datasets, and show that our methods dramatically reduce the number of queries required to achieve the same estimation accuracy of state-of-the-art methods.

EXISTING SYSTEMS:-

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.

PROPOSED SYSTEMS:-

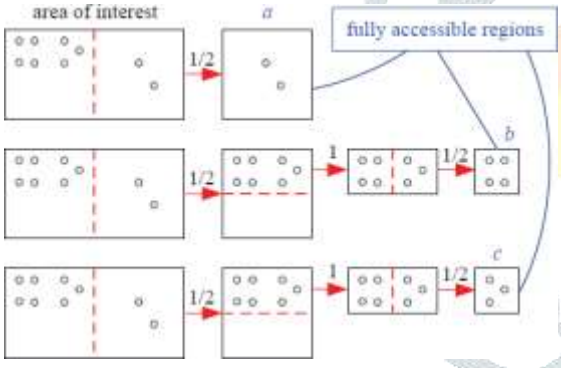
Propose sampling methods to accurately estimate PoI statistics such as sum and average aggregates from as few queries as possible. Experimental results based on

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.

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.

SYSTEM ARCHITECTURE



IMPLEMENTATION

Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective.

The implementation stage involves careful planning, investigation of the existing system and it's constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods.

Modules Description:

In this Project, We described three modules i).Points Of Interests,
ii). Sampling,
iii).Measurement

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.

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.

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.

Algorithms:-

i).Random Region Zoom-in

ii).Random Region Zoom-in Count **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.

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

Random Region Zoom-in Count:

Algorithm 2: RRZIC(A) pseudo-code.

```
input : A
/* Q is a sub-region sampled from A at
   random, and τ records the probability of
   sampling Q from A.                          */
output: Q and τ
/* countPoI(Q) returns the number of PoIs in Q.  */
Q ← A, τ ← 1, and z ← countPoI(Q);
/* k is the maximum number of PoIs returned in
   a response to a query.                      */
while z > k do
    /* χ0(Q) and χ1(Q) are the two sub-regions of Q
       defined as (1) and (2).                 */
    Q0 ← χ0(Q) and Q1 ← χ1(Q);
    /* z0 and z1 are the numbers of PoIs within the
       regions Q0 and Q1 respectively.         */
    z0 ← countPoI(Q0) and z1 = z − z0;
    /* U(0,1) is a random sample from (0,1) .  */
    u ← U(0,1);
    if u < z0/z then
        Q ← Q0, τ ← τ × z0/z, and z ← z0;
    else
        Q ← Q1, τ ← τ × z1/z, and z ← z1;
    end
end
```

Conclusion:-

We propose methods to sample PoIs on maps, and give consistent estimators of PoI aggregate statistics. We show that the mix method RRZI URS is more accurate than RRZI under the same number of queries used. When PoI count information is provided by public APIs, RRZIC MHWRS utilizing this meta information is more accurate than RRZI URS. The experimental results based on a variety of real datasets show that our methods are efficient, and they sharply reduce the number of queries required to achieve the same estimation accuracy of stateof- the-art methods.

REFERENCES

[1].P. Wang, W. He, X. Liu, "An efficient sampling method for characterizing points of interests on maps", Proc. IEEE 30th Int. Conf. Data Eng., pp. 1012–1023.

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

[3].Y. Li, M. Steiner, L. Wang, Z.-L. Zhang, J. Bao, "Exploring venue popularity in foursquare", Proc. 5th IEEE Int. Workshop Netw. Sci. Commun. Netw., pp. 1- 6, 2013.

[4].N. Dalvi, R. Kumar, A. Machanavajjhala, V. Rastogi, "Sampling hidden objects using Nearest-neighbor oracles", Proc. ACM SIGKDD, pp. 1325-1333, Dec. 2011.

[5].Y. Li, M. Steiner, L. Wang, Z.-L. Zhang, J. Bao, "Dissecting foursquare venue popularity via random region sampling", Proc. ACM Conf. CoNEXT Student Workshop, pp. 21-22, 2012.

[6].S. Chib, E. Greenberg, "Understanding the Metropolis-hastings algorithm", The Am. Statist., vol. 49, no. 4, pp. 327-335, Nov. 1995.

[7].W. K. Hastings, "Monte carlo sampling methods using Markov chains and their applications",

Biometrika, vol. 57, no. 1, pp. 97-109, Apr. 1970.

[8].N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, E. Teller, "Equations of state calculations by fast computing machines", IEEE J. Sel. Areas Commun., vol. 21, no. 6, pp. 1087-1092, Jun. 2011.

[9].P. Rusmevichientong, D. M. Pennock, S. Lawrence, L. C. Giles, "Methods for sampling pages uniformly from the world wide web", Proc. AAAI Fall Symp. Using Uncertainty Within Comput., pp. 121-128, Nov. 2001.

[10].Z. Bar-Yossef, M. Gurevich, "Efficient search engine measurements", Proc. WWW, pp. 401-410, 2007.

[11].Z. Bar-Yossef, M. Gurevich, "Mining search engine query logs via suggestion sampling", Proc. VLDB Endowment, vol. 1, no. 1, pp. 54-65, Aug. 2008.

[12].M. Zhang, N. Zhang, G. Das, "Mining a search engine's corpus: Efficient yet unbiased sampling and aggregate estimation", Proc. ACM SIGMOD Int. Conf. Manage. Data, pp. 793-804, 2011.