International Journal of Computer Science & Mechatronics
A peer reviewed international journal | Article Available at www.ijcsm.in | sjif-4.454
©smsamspublications.com | Vol.3.Issue.2.2017

Crawling Hidden Objects with K-NN Queries

ISSN: 2455-1910

Mr.A.ChandraMouli[1], Ch.Vasantha[2], G.Laskhmi Prasanna[3], A.Sriharsha[4], N.Triveka[5]
[1]Assistant Professor of CSE, [2,3,4,5]Project Team of CSE, Department of CSE
PSCMR College of Engineering & Technology, Vijayawada-1.AP.INDIA.
Email:achandramouli@pscmr.ac.in,chvasantha066@gmail.com,lakshmiprasannagoddati@gmail.com ,ajrsrharsha@gmail.com,trivekaniduprolu@gmail.com

ABSTRACT: Many website giving Location based primarily services(LBS) offer a K-NN search interface that returns the top-k nearest-neighbour objects (e.g., nearest restaurants) for a given question location. This paper address the matter of travel all objects with efficiency from associate in nursing LBS website through the general public K-NN net search interface it provides. Specifically, we have a tendency to develop travel rule for MD and higher-dimensional areas severally and demonstrate through theoretical analysis that the overhead of our algorithms will be finite by a perform of variety| the quantity |the amount of dimensions and the number of crawled objects no matter the underlying distributions of objects we have a tendency. To conjointly extend the algorithms to leverage eventualities where ever sure auxiliary data regarding the underlying knowledge distribution, e.g., the population density of vicinity that is commonly absolutely related with the density of LBS objects, is accessible intensive experiments on real-world datasets demonstrate the prevalence of our algorithms over the progressive competitors within the literature.

Key Words: *Library, Android, Online, Database, Transactions, Distributed etc.*

I.Introduction: With chop-chop growing quality, Location primarily based Services (LBS), e.g., Google Maps, Yahoo native, We Chat, Four sq., etc., started providing web-based search options that correspond k-NN question interface. Specifically for a user specified question location alphabetic character, these web-sites extract from the objects in its backend information the top-k nearest neighbours to alphabetic character and come these k objects to the user through the net interface. Here k is usually a tiny low price like fifty or one hundred. For instance, Mc-Donald's returns the highest twenty five nearest restaurants for a user-specified location through its locations search webpage. While such a MD 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 MD 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. Here "crawling" is broadly defined, i.e., it can refer to the extraction of all objects from the database, or only those objects that satisfy certain selection conditions, so long as such conditions can be "passed through" to the k-NN interface. For example, if the target here is to crawl Google Maps, then the objective may be to crawl all Vietnamese restaurants in Washington, DC. One can see that this condition can be easily passed through to Google Maps by restricting query locations to be from Washington, DC, and specifying "Vietnamese restaurants" as the search keyword. It is important to note that the key technical challenge for crawling through a MD interface is to minimize the number of queries issued to the LBS service. The requirement is caused by limitations imposed by most LBS services on the number of queries allowed from an IP address or a user account (in case of an API service such as Google Maps) for a given time period (e.g., one day). For example, Twitter limits the search rate at 180 queries per 15 minute. Of course, no algorithm can possibly accomplish the task without issuing at least n=k queries, where n is output size (i.e., the number of crawled objects), because each query returns at most k of the n objects. As such, we are bound to have an output-sensitive algorithm, which nevertheless should have a query cost as close to n=k as possible.

II.MULTIPLE DIMENSIONAL ANALYSES:

Though 2-D spatial databases are the most popular ones in the real world, there still exist some applications of k-NN spatial databases in higher dimensional spaces (three or more dimensions). For example, the coastal.com website [4] allows users to perform k-NN queries for looking for glasses in 4-D space, with dimensions including temple arm length, lens height, lens width and DBL (distance between lenses). In order to give a solution for higher dimensional spaces and make our approach more complete, in this section we discuss how to extend our 2-D crawling algorithm to an m-D space.

The crawling algorithm for m-D spaces is designed in a recursive manner that an m-D space crawler is conducted by calling (m-1)-D crawling algorithm.

Let D = fp$_1$; p$_2$; ; p$_n$g be a spatial database with k-NN interface in an m-D space, bounded in $V^m = [a_{1;l}; a_{1;r}]\ [a_{2;l}; a_{2;r}]\ [a_{m;l}; a_{m;r}]$. Now each query q covers a sphere $V^m(q)$ in the m-D space instead of a circle $V^2(q)$ in the 2-D space. In order to crawl all points of D, we first get the mid hyper plane $V^{m\ 1}$ of V^m and conduct (m-1)-D crawling algorithm on $V^{m\ 1}$ (suppose m > 3).

Algorithm m-D Crawling Algorithm

Algo (Vm,m):

Input: D: a database in an m-D space; $V^m = [a_{1;l}; a_{1;r}][a_{m;l}; a_{m;r}]$

Output: all points of D

U = fV ^{m}g /*the set of uncovered subspaces */

P = fg /*D points returned currently*/

while (U is not empty) **do**

get V_i^m from U (V_i^m is an element in U); $V_i^{m\ 1}$ is the mid hyperplane of $V_i^{m\ 1}$

call Algo($V_i^{m\ 1}$; m 1); get slice S_i

add returned D points to P

if V_i^m is not fully covered by S_i **then**

two sub-spaces $V_{i;}^{m}{}_1$, and $V_{i;}^{m}{}_2$ are produced from

V_i^m S_i

U = U ^{S}f$V_{i;}^{m}{}_1$; $V_{i;}^{m}{}_2$g /*add the two new sub-spaces*/

end if

U = U $fV_i^m g$ /*remove the subspace*/

end while

return P

To study the upper bound of the m-D crawling algorithm, we first look at the upper bound for covering a 1-D line segment using m-D queries. In fact, we can easily extend the claim of Proposition 1 to line covering in multidimensional spaces.

III. RESULTS:

Figure 3.1 Home Page

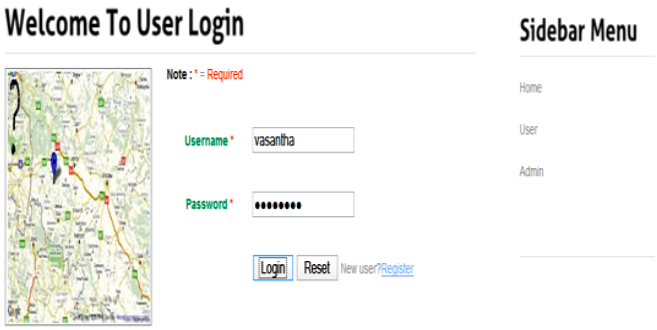

Figure 3.2 Login Page

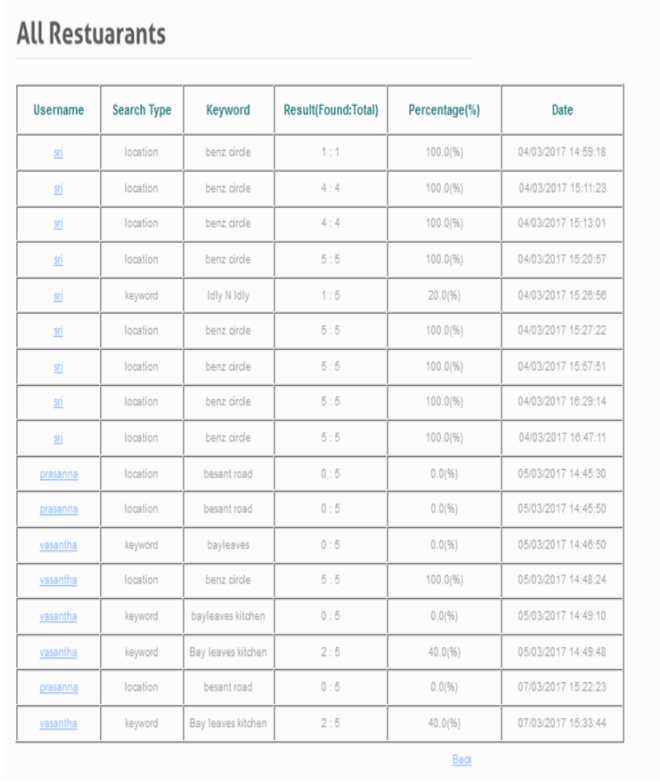

Figure 3.3: Results Searched

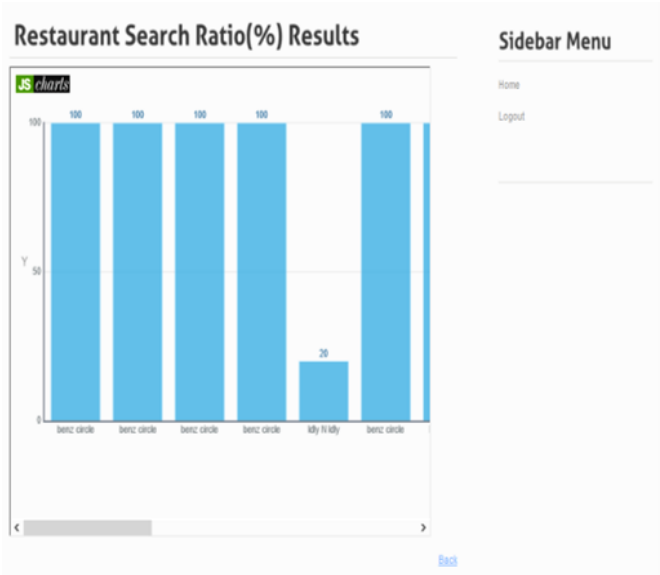

Figure 3.4: Restaurant Search Ratio Results

IV. CONCLUSION: In this paper, we study the problem of crawling the LBS through the restricted k-NN search interface. Al-though hidden points usually exist in 2-D space, there are some applications with points in higher dimensional spaces. We extend the 2-D crawling algorithm to the general m-D space, and give the m-D crawling algorithm with theoretical upper bound analysis. For 2-D space, we take external

knowledge into consideration to improve the crawling performance. The experimental results show the effectiveness of our proposed algorithms. In this study, the proposed algorithms crawl data objects by given a rectangle (cube) in the spatial space. In the general situation when the bounded region of the objects is irregular, it can be pre-partitioned into a set of rectangles (cubes) before using the techniques proposed in this paper.

REFERENCES:

[1] Mcdonalds, "Mcdonalds page, http://www.mcdonalds.com/," [Accessed: Aug. 6, 2014]. [Online]. Available: nurlfhttp://www.mcdonalds.com/us/ en/restaurant locator.htmlg

[2] S. Byers, J. Freire, and C. T. Silva, "Efficient acquisition of web data through restricted query interfaces," in Poster Proceedings of the Tenth International World Wide Web Conference, WWW 10, Hong Kong, China, May 1-5, 2001, 2001. [Online]. Available: http://www10.org/cdrom/posters/1051.pdf

[3] W. D. Bae, S. Alkobaisi, S. H. Kim, S. Narayanappa, and C. Shahabi, "Web data retrieval: solving spatial range queries using k-nearest neighbor searches," Geoinformatica, vol. 13, no. 4, pp. 483–514, 2009.

[4] G. E. Glasses, "Great eye glasses page, http://www.greateyeglasses.com/shop/search.php ," [Ac-cessed: Jan. 20, 2014]. [Online]. Available: nurlfhttp: //www.greateyeglasses.com/shop/search.phpg

[5] Yahoo, "Yahoo local page, https://local.yahoo.com/," [Accessed: Dec. 2012]. [Online]. Available: nurlfhttps: //local.yahoo.com/g

[6] U. Census, "Us census, http://www.census.gov/cgi-bin/geo/shapefiles2013/layers.cgi," [Accessed: Dec. 2013]. [Online]. Available: nurlfhttp://www.census.gov/cgi-bin/ geo/shapefiles2013/layers.cgig

[7] L. Devroye, "Sample-based non-uniform random variate gen-eration," in Proceedings of the 18th conference on Winter simula-tion. ACM, 1986, pp. 260–265.

[8] L. Barbosa and J. Freire, "Siphoning hidden-web data through keyword-based interfaces," in SBBD, 2004, pp. 309–321.

[9] A. Ntoulas, P. Pzerfos, and J. Cho, "Downloading textual hidden web content through keyword queries," in Digital Libraries, 2005. JCDL'05. Proceedings of the 5th ACM/IEEE-CS Joint Conference on. IEEE, 2005, pp. 100–109.

[10] K. Vieira, L. Barbosa, J. Freire, and A. Silva, "Siphon++: a hidden-webcrawler for keyword-based interfaces," in Proceed-ings of the 17th ACM conference on Information and knowledge management. ACM, 2008, pp. 1361–1362.

[11] L. Jiang, Z. Wu, Q. Feng, J. Liu, and Q. Zheng, "Efficient deep web crawling using reinforcement learning," in Advances in Knowledge Discovery and Data Mining. Springer, 2010, pp.

ABOUT AUTHORS:

Mr.**A.Chandra Mouli,** Assistant Professor of CSE, His research interests are Computer Networks, Data Ware Housing and Data Mining.

Ms.**Challa Vasantha**, IV, B.Tech CSE, her technical interests include Android, Java, MySql, Rational Rose and Web Applications.

Ms **G. Lakshmi Prasanna** IV B.Tech CSE, her Technical Interests include Android, Java, MySql, Web Applications and SEO Tools.

Mr.**A.J.R.Sri Harsha**, IV B.Tech CSE, his technical interests include Android, Java, MySql.

Ms.N.Triveka, IV B.Tech CSE, her technical interests are Android, Java, MySQL.