Survey on Framework for Location-Aware Indexing and Query Processing

K. Kausika[1] M. Sangeetha[2]
[1]Research Scholar [2]Assistant Professor
[1,2]Department of Computer Science
[1,2]Kaamadhenu Arts and Science College, Sathyamangalam, Tamilnadu, India

Abstract— The generic location-aware rank query (GLRQ) over a set of location-aware objects. A GLRQ is composed of a spatial location, a set of keywords, a query predicate, and a ranking function formulated on location, text and other attributes. The result consists of k objects satisfying the predicate ranked according to the ranking function. Some queries cannot be processed efficiently using existing techniques. To handle the predicate and the attribute-based scoring, we devise a new index structure called synopses tree, which contains the synopses of different subsets of the dataset. The synopses tree enables pruning of search space according to the satisfiability of the predicate. To process the query constraints over the location and keywords, the framework integrates the synopses tree with the Spatio-textual index. Conduct extensive experiments to demonstrate the solution provides excellent query performance.

Key words: Location-Aware, Rank Query, Synopses Tree, LINQ

I. INTRODUCTION

The widespread adoption of mobile telephony is more and more convenient for users to capture and publish geo-locations. As a consequence, more and more location-aware datasets have been created and made available on the Web. For example, Flicker, one of the biggest photo-sharing website, has millions of geo-tagged items every month. The popularity and large scale of the location-aware datasets make location-aware queries important. location-aware rank query include the (k-) nearest neighbor (NN) query and location aware keyword query (LKQ, spatial keyword query) [1], [2], [3], [4], [5]. NN queries and LKQs have wide applications in many domains. On the Flicker dataset, users may want to fetch the relevant and nearest photos that are highly-rated. People may wish to search by not only location and keywords, but also conditions on other attributes. The generic location-aware rank query (GLRQ) over a set of location-aware objects is composed of a query location, a set of keywords, a query predicate, and a ranking function combining spatial proximity, textual relevance and other measures (e.g. certain attribute values). The result of the query consists of k objects that satisfy the predicate, ranked according to the specified ranking function. The GLRQ has the NN query and LKQ as special cases. Conduct extensive experiments on both real and synthetic datasets, and the results demonstrate the effectiveness and efficiency of our method.

II. LITERATURE REVIEW

A. I. De Felipe, V. Hristidis, and N. Rishe, "Keyword search on spatial databases," In Proceedings of International Conference on Data Engineering (ICDE), 2008 [1].

This work addresses a novel spatial keyword query called the m-closest keywords (mCK) query. Given a database of spatial objects, each tuple is associated with some descriptive information represented in the form of keywords. The mCK query aims to find the spatially closest tuples which match m user-specified keywords. Given a set of keywords from a document, mCK query can be very useful in geotagging the document by comparing the keywords to other geotagged documents in a database. To answer mCK queries efficiently, They introduce a new index called the bR*-tree, which is an extension of the R*-tree. Based on bR*-tree, They exploit a priori-based search strategies to effectively reduce the search space. They also propose two monotone constraints, namely the distance mutex and keyword mutex, as their a priori properties to facilitate effective pruning. Their performance study demonstrates their search strategy is indeed efficient in reducing query response time and demonstrates remarkable scalability in terms of the number of query keywords which is essential for our main application of searching by document

B. G. Cong, C. S. Jensen, and D. Wu, "Efficient retrieval of the top-k most relevant spatial web objects," In Proceedings of VLDB Endowment., volume 2 [2].

The conventional Internet is acquiring a geo-spatial dimension. Web documents are being geo-tagged, and geo-referenced objects such as points of interest are being associated with descriptive text documents. The resulting fusion of geo-location and documents enables a new kind of top-k query that takes into account both location proximity and text relevancy. To their knowledge, only naive techniques exist that capable of computing a general web information retrieval query while also taking location into account. They proposes a new indexing framework for location aware top-k text retrieval. The framework leverages the inverted file for text retrieval and the R-tree for spatial proximity querying. Several indexing approaches are explored within the framework. The framework encompasses algorithms that utilize the proposed indexes for computing the top-k query, thus taking into account both text relevancy and location proximity to prune the search space. Results of empirical studies with an implementation of the framework demonstrate that the paper's proposal offers scalability and is capable of excellent performance.

C. Z. Li, K. Lee, B. Zheng, W.-C. Lee, D. L. Lee, and X. Wang, "IR-Tree: an efficient index for geographic document search," IEEE Trans. Knowledge and Data Engineering [3].

Given a geographic query that is composed of query keywords and a location, a geographic search engine

retrieves documents that are the most textually and spatially relevant to the query keywords and the location, respectively, and ranks the retrieved documents according to their joint textual and spatial relevances to the query. The lack of an efficient index that can simultaneously handle both the textual and spatial aspects of the documents makes existing geographic search engines inefficient in answering geographic queries. They propose an efficient index, called IR-tree that together with a top-k document search algorithm facilitates four major tasks in document searches, namely, 1) spatial filtering, 2) textual filtering, 3) relevance computation, and 4) document ranking in a fully integrated manner. In addition, IR-tree allows searches to adopt different weights on textual and spatial relevance of documents at the runtime and thus caters for a wide variety of applications. A set of comprehensive experiments over a wide range of scenarios has been conducted and the experiment results demonstrate that IR-tree outperforms the state-of-the-art approaches for geographic document searches.

D. Chengyuan Zhang, Ying Zhang, Wenjie Zhang, and Xuemin Lin, "Inverted linear quadtree: Efficient top k spatial keyword search," In Proceedings of International Conference on Data Engineering 2013 [4].

With advances in geo-positioning technologies and geo-location services, there are a rapidly growing amount of spatio-textual objects collected in many applications such as location based services and social networks, in which an object is described by its spatial location and a set of keywords (terms). Consequently, the study of spatial keyword search which explores both location and textual description of the objects has attracted great attention from the commercial organizations and research communities. In the paper, we study the problem of top k spatial keyword search (TOPK-SK), which is fundamental in the spatial keyword queries. Given a set of spatio-textual objects, a query location and a set of query keywords, the top k spatial keyword search retrieves the closest k objects each of which contains all keywords in the query. Based on the inverted index and the linear quadtree, we propose a novel index structure, called inverted linear quadtree (IL-Quadtree), which is carefully designed to exploit both spatial and keyword based pruning techniques to effectively reduce the search space. An efficient algorithm is then developed to tackle top k spatial keyword search. In addition, we show that the IL-Quadtree technique can also be applied to improve the performance of other spatial keyword queries such as the direction-aware top k spatial keyword search and the spatio-textual ranking query. Comprehensive experiments on real and synthetic data clearly demonstrate the efficiency of our methods.

E. G. Cormode, M. Garofalakis, P. J. Haas, and C. Jermaine, "Synopses for massive data: Samples, histograms, wavelets, sketches," Found. Trends databases, volume 4 [5].

Methods for Approximate Query Processing (AQP) are essential for dealing with massive data. They are often the only means of providing interactive response times when exploring massive datasets, and are also needed to handle high speed data streams. These methods proceed by computing a lossy, compact synopsis of the data, and then executing the query of interest against the synopsis rather than the entire dataset. They describe basic principles and recent developments in AQP. They focus on four key synopses: random samples, histograms, wavelets, and sketches. They consider issues such as accuracy, space and time efficiency, optimality, practicality, range of applicability, error bounds on query answers, and incremental maintenance. They also discuss the trade-offs between the different synopsis types.

F. K. Tzoumas, A. Deshpande, and C. S. Jensen, "Lightweight graphical models for selectivity estimation without independence assumptions," In Proceedings of VLDB Endowment., volume 4 [6].

Query optimizers rely on statistical models that succinctly describe the underlying data. Models are used to derive cardinality estimates for intermediate relations, which in turn guide the optimizer to choose the best query execution plan. The quality of the resulting plan is highly dependent on the accuracy of the statistical model that represents the data. It is well known that small errors in the model estimates propagate exponentially through joins, and may result in the choice of a highly sub-optimal query execution plan. Most commercial query optimizers make the attribute value independence assumption: all attributes are assumed to be statistically independent. This reduces the statistical model of the data to a collection of one-dimensional synopses (typically in the form of histograms), and it permits the optimizer to estimate the selectivity of a predicate conjunction as the product of the selective constituent predicates. However, this independence assumption is more often than not wrong, and is considered to be the most common cause of sub-optimal query execution plans chosen by modern query optimizers. They take a step towards a principled and practical approach to performing cardinality estimation without making the independence assumption. By carefully using concepts from the field of graphical models, they are able to factor the joint probability distribution over all the attributes in the database into small, usually two-dimensional distributions, without a significant loss in estimation accuracy. They show how to efficiently construct such a graphical model from the database using only two-way join queries, and show how to perform selectivity estimation in a highly efficient manner. They integrate their algorithms into the PostgreSQL DBMS. Experimental results indicate that estimation errors can be greatly reduced, leading to orders of magnitude more efficient query execution plans in many cases. Optimization time is kept in the range of tens of milliseconds, making this a practical approach for industrial-strength query optimizers.

G. D. Papadias, P. Kalnis, J. Zhang, and Y. Tao, "Efficient OLAP operations in spatial data warehouses," In Proceedings of International Symposium, Advances in Spatial and Temporal Databases (SSTD) [7].

Spatial databases store information about the position of individual objects in space. In many applications however, such as traffic supervision or mobile communications, only summarized data, like the number of cars in an area or phones serviced by a cell, is required. Although this information can be obtained from transactional spatial databases, its computation is expensive, rendering online processing inapplicable. Driven by the non-spatial

paradigm, spatial data warehouses can be constructed to accelerate spatial OLAP operations. They consider the star-schema and focus on the spatial dimensions. Unlike the non-spatial case, the groupings and the hierarchies can be numerous and unknown at design time, therefore the well-known materialization techniques are not directly applicable. In order to address this problem, they construct an ad-hoc grouping hierarchy based on the spatial index at the finest spatial granularity. They incorporate this hierarchy in the lattice model and present efficient methods to process arbitrary aggregations. They finally extend their technique to moving objects by employing incremental update methods.

H. G. R. Hjaltason and H. Samet, "Distance browsing in spatial databases," In Proceedings of ACM Trans. Database System, volume 24 [8].

They compare two different techniques for browsing through a collection of spatial objects stored in an R-tree spatial data structure on the basis of their distances from an arbitrary spatial query object. The conventional approach is one that makes use of a k-nearest neighbor algorithm where k is known prior to the invocation of the algorithm. Thus if m < k neighbors are needed, the k-nearest neighbor algorithm has to be reinvoked for m neighbors, thereby possibly performing some redundant computations. The second approach is incremental in the sense that having obtained the k nearest neighbors, the k + 1st neighbor can be obtained without having to calculate the k + 1 nearest neighbors from scratch. The incremental approach is useful when processing complex queries where one of the conditions involves spatial proximity (e.g., the nearest city to Chicago with population greater than a million), in which case a query engine can make use of a pipelined strategy. They present a general incremental nearest neighbor algorithm that is applicable to a large class of hierarchical spatial data structures. This algorithm is adapted to the R-tree and its performance is compared to an existing k-nearest neighbor algorithm for R-trees [Rousseopoulos et al. 1995]. Experiments show that the incremental nearest neighbor algorithm significantly outperforms the k-nearest neighbor algorithm for distance browsing queries in a spatial database that uses the R-tree as a spatial index. Moreover, the incremental nearest neighbor algorithm usually outperforms the k-nearest neighbor algorithm when applied to the k-nearest neighbor problem for the R-tree, although the improvement is not nearly as large as for distance browsing queries. In fact, they prove informally that at any step in its execution the incremental nearest neighbor algorithm is optimal with respect to the spatial data structure that is employed. Furthermore, based on some simplifying assumptions, they prove that in two dimensions the number of distance computations and leaf nodes accesses made by the algorithm for finding k neighbors is O(k + k).

I. A. Guttman, "R-trees: a dynamic index structure for spatial searching," In Proceedings of SIGMOD Rec., volume 14 [9].

In order to handle spatial data efficiently, as required in computer aided design and geo-data applications, a database system needs an index mechanism that will help it retrieve data items quickly according to their spatial locations However, traditional indexing methods are not well suited to data objects of non-zero size located m multi-dimensional spaces they describe a dynamic index structure called an R-tree which meets this need, and give algorithms for searching and updating it. They present the results of a series of tests which indicate that the structure performs well, and conclude that it is useful for current database systems in spatial applications.

J. J. A. B. Rocha-Junior, O. Gkorgkas, S. Jonassen, "Efficient processing of top-k spatial keyword queries," In Proceedings of International Conference, Advances in spatial and temporal databases (SSTD), [10].

Given a spatial location and a set of keywords, a top-*k* spatial keyword query returns the *k* best spatio-textual objects ranked according to their proximity to the query location and relevance to the query keywords. There are many applications handling huge amounts of geotagged data, such as Twitter and Flickr, that can benefit from this query. Unfortunately, the state-of-the-art approaches require non-negligible processing cost that incurs in long response time. They propose a novel index to improve the performance of top-*k* spatial keyword queries named *Spatial Inverted Index* (S2I). Their index maps each distinct term to a set of objects containing the term. The objects are stored differently according to the document frequency of the term and can be retrieved efficiently in decreasing order of keyword relevance and spatial proximity. Moreover, they present algorithms that exploit S2I to process top-*k* spatial keyword queries efficiently. Finally, they show through extensive experiments that our approach outperforms the state-of-the-art approaches in terms of update and query cost.

S.no	Author	Algorithm	Features	Problems
1	I.De.Felipe, V.Hristids, N.Rishe	m-closest keywords (mCK) query	Effectively reduce the search space	Not good accuracy
2	G.Cong, C.S.Jensen, D.Wu	Top-k query	Text relevancy and Proximity to prune the search space	Establish robustness
3	Z.Li, K.Lee, B.Zheng	Top k document	Allows searches to adopt different weights	Wide range of scenarios has been conducted
4	Chengyuan Zhang, Ying Zhang	Inverted linear quadtree (IL-Quadtree)	To improve the performance of other spatial keyword queries	Ranking process are difficult
5	G.Cormoda, M.Garofalakis, P.J.Haas	Approximate query processing (AQP)	Computing a lossy and compact synopsis of data	Accuracy, Optimality and Time efficiency
6	K.Tzoumas, A.Deshpande, C.S.Jensen	Sub-optimal query	Estimation errors can be reduced	Time in the range of tens of milliseconds

7	D.Papadias, P.Kalnis, J.Zhang, Y. Tao	Star schema	Efficient methods to process arbitrary aggregations	Object techniques are extending
8	G.R.Hjaltason, H. Samet	K- Nearest neighbor	Simplifying assumptions	No improvement for distance browsing
9	A.Guttman	Dynamic index structure	Performs searching and updating	Only for current database system
10	J.A.B.Rocha-Junior, O.Gkorgkas, S.Jonassen	Top-k spatial keyword query	Efficiently retrieved in decreasing order	Highly expensive

Table 1: Comparison of Framework Indexing and Query Processing

III. Conclusion

This paper as presented different algorithm for an important class of query, generic location-aware rank query (GLRQ) and propose a framework called LINQ to process the query. In this framework, a novel index structure called synopses tree is built, which indexes synopses of objects, and enables efficient pruning and estimation. The synopses tree is designed to reduce the cost of construction while preserving accuracy. This paper shows how to process GLRQs efficiently in the LINQ framework, leveraging synopses tree and other index structures. Experimental results show that the proposed framework is effective and efficient.

References

[1] I. De Felipe, V. Hristidis, and N. Rishe, "Keyword search on spatial databases," in Proc. Int'l Conf. Data Eng. (ICDE), 2008, pp. 656–665.
[2] G. Cong, C. S. Jensen, and D. Wu, "Efficient retrieval of the top-k most relevant spatial web objects," Proc. VLDB Endow., vol. 2, pp. 337–348, 2009.
[3] Z. Li, K. Lee, B. Zheng, W.-C. Lee, D. L. Lee, and X. Wang, "IR-Tree: an efficient index for geographic document search," IEEE Trans. Knowledge and Data Eng., vol. 23, no. 4, pp. 585–599, 2011.
[4] Chengyuan Zhang, Ying Zhang, Wenjie Zhang, and Xuemin Lin, "Inverted linear quadtree : Efficient top k spatial keyword search," in Proc. Int'l Conf. Data Eng. (ICDE), 2013.
[5] G. Cormode, M. Garofalakis, P. J. Haas, and C. Jermaine, "Synopses for massive data: Samples, histograms, wavelets, sketches," Found. Trends databases, vol. 4, no. 1–3, pp. 1–294, 2012.
[6] K. Tzoumas, A. Deshpande, and C. S. Jensen,"Lightweight graphical models for selectivity estimation without independence assumptions," Proc. VLDB Endow., vol. 4, no. 11, pp. 852–863, 2011.
[7] D. Papadias, P. Kalnis, J. Zhang, and Y. Tao, "Efficient OLAP operations in spatial data warehouses," in Proc. Int'l Symp. Advances in Spatial and Temporal Databases (SSTD), 2001, pp. 443–459.
[8] G. R. Hjaltason and H. Samet, "Distance browsing in spatial databases," ACM Trans. Database Syst., vol. 24, no. 2, pp. 265– 318, 1999.
[9] A. Guttman, "R-trees: a dynamic index structure for spatial searching," SIGMOD Rec., vol. 14, no. 2, pp. 47–57, 1984.
[10] J. A. B. Rocha-Junior, O. Gkorgkas, S. Jonassen, and K. Nørvag, ˚ "Efficient processing of top-k spatial keyword queries," in Proc. Int'l Conf Advances in spatial and temporal databases (SSTD), 2011, pp. 205–222.
[11] F. Buccafurri, G. Lax, S. Domenico, L. Pontieri, and D. Rosaci, "Enhancing histograms by tree-like bucket indices," The VLDB Journal, vol. 17, no. 5, pp. 1041–1061, 2008.
[12] D. Lemire, O. Kaser, and K. Aouiche, "Sorting improves wordaligned bitmap indexes," Data Knowl. Eng., vol. 69, no. 1, pp. 3–28, 2010.
[13] D. Wu, G. Cong, and C. Jensen, "A framework for efficient spatial web object retrieval," The VLDB Journal, pp. 1–26, 2012.
[14] A. Silberschatz, H. F. Korth, and S. Sudarshan, Database system concepts. McGraw- Hill Higher Education, 2006.
[15] K. S. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft, "When is nearest neighbor meaningful?" in Proc. Int'l Conf. Database Theory (ICDT), 1999, pp. 217–235.

