Research Proposal

Efficient Indexing and Query Processing of Large-scale Spatially Rich User-generated Content

Usvyatsov Mikhail

Supervised by Prof. Qiang Qu
Dainfos Lab, Innopolis University

June 1, 2015

1 Abstract

During the past decade, we have witnessed an exponential growth of user-generated data, such as social networked data. The increasing volume and complexity of user-generated content pose a great challenge to efficiently manage the data. To be noted, with the increasing use of geo-enabled devices (e.g., smartphones and tablets), a large volume of such content associated with geo-locations are becoming available. Particularly, enriched by other social media, e.g., textual description of business entities on Google Business, images of places of interest on Instagram, and user interactions on Twitter, geo-tagged content enable a wide array of location-based services. These services include but not limited to location-aware web search, location-based recommendation, and location prediction. In this proposal, such geo-tagged content is termed spatially rich data. Based on these trends on spatially rich data, I am interested in exploring research touching upon data indexing, query processing, and applications of such content. In this proposal, some initial ideas are briefly presented on data indexing and query processing where spatial and textual properties of objects available on the web are taken into account.

2 Introduction

We are witnessing a development where increasing volumes of data content is being associated with locations, yielding what we term points of interest (POIs). Each POI has a location and a set of non-spatial properties especially textual descriptions. It has been reported that 25% of tweets from mobile devices are associated with locations. This trend is reshaping one of the most important services on the web — keyword search. Billions of search queries are processed each day, a substantial fraction of which have local intent, meaning that they target web content that relates to places near the user. Studies report that 20% of Google desktop searches and 53% of mobile Bing searches have local intent. However, we have found that queries with spatial and non-spatial properties often result in computationally expensive.

Top-k queries have drawn substantial attention because of wide applications. To consider spatial constraints in web search, we may consider one type of top-k queries as a top-k spatial search. It searches most relevant POIs in a Region of Interest (ROI) considering both spatial and textual properties of objects. An object is represented by its spatial location (e.g., GPS coordinates) and textual description (e.g., business description of a shop) [1]. To address this problem, there exist several important questions that have to be tackled:

1. Given a dataset, how to efficiently index the data to enable efficient query processing?
2. How to combine new indexing methods with existing DBMS systems and tools?

3. How to incrementally maintain the data indexing and provide streaming services?

4. How to rank such objects by means of two different types of properties?

The rest of the paper is organized as follows. Section 3 briefly reviews the literature. Section 4 reports some of my current work. Finally, future work is discussed in Section 5.

3 Related Work

Several recent studies propose to use the following representation of spatial textual objects: [1, 5, 8].

D = <D.id, D.lat, D.lng, D.terms>,

Where D is a document consisting of textual terms D.terms. D has id D.id, and spatial coordinates D.lat (latitude) and D.lng (longitude).

Given a query with a location and a set of textual terms, according to the study [9], the scoring relevance can be calculated by Equation 1:

$$\tau(D, q) = \alpha \cdot \delta(D.l, q.l) + (1 - \alpha) \cdot \theta(D.d, q.d), \quad (1)$$

Where $\delta(D.l, q.l)$ denotes the spatial proximity between document and query locations, $\theta(D.d, q.d)$ is textual relevance of a document against the query terms, and $alpha \in (0, 1)$ is a parameter that specifies balance of importance between textual and spatial relevance scores. Note that textual proximity computation depends on document representation, e.g. Bag-of-words model or vector space model.

Many of existing studies [1, 6, 8] on spatial textual queries often use the combination of R-tree [3] for the spatial indexing and inverted index [4] for text indexing [2]. There are several methods that have been proposed for top-k query processing:

1. text-first search methods (I3 [5], S2I [6])

2. spatial-first search methods(SKI [7]) [8]

3. combined methods (IR-tree [8])

Text-first search methods consider textual index first, using textual terms to filter search results and then considering the constraints of spatial locations. Spatial-first search methods are the opposite case. Combined methods integrate the two types of indexing structures. Among them, I3, S2I, and IR-tree have been paid widely attention.

I3 is organized as inverted index with terms as keys and postings of Quadtrees. This structure is very efficient. However, it still has many problems, e.g. when a lot of objects are located on the same street, the size of postings lists increases very fast and search over it becomes slower. It also inherits all the limitations of Quadtree. Furthermore, in our experiments, we find that when objects share the same location, I3 simply could not perform indexing.

S2I consists of three components: vocabulary, blocks, and trees. The main idea of S2I is to store documents of different frequencies differently. The most frequent terms are stored in R-tree while rare documents are stored in blocks. The term is considered to be frequent if its frequency reaches a predefined threshold T. It works well because terms in a corpus follows Zipf's low, which means that there are not so many very frequent terms [9]. However, it is hard to predefine the threshold before looking into the datasets carefully. When T is small, we will frequently read objects from blocks which could not leverage the advantages of fast spatial search through R-trees. On the other hand, when T is big, many keywords become infrequent that affects spatial pruning during query processing.

IR-trees combines R-tree and textual indexing. Each node of R-tree is augmented by an inverted index on the textual terms of all the objects within the node. It allows to make textual pruning while pruning locations. However, this approach requires very high cost of updates: even one node split will require to rebuild all inverted indexes. Moreover, a large number of pseudo documents have to be accessed when a query contains frequent keywords, which means high IO. Hence, IR-tree cannot be scaled to a large-scale dataset.

4 Current Work

I am now working on a project at Dainfos lab advised by prof. Qiang Qu. We have done an experimental comparison of the three algorithms: I3, S2I and IR-trees on a collection of Twitter data. Furthermore, the work under a new algorithm was started.

Initial experiments are tested on a small dataset with 36000 documents, where I have done a unified platform to preprocess the data that can be fed into the three algorithms. A test algorithm is implemented, which generates random queries, launch algorithms, and report algorithms automatically. As follows, we show some initial results obtained by average performance on random queries with 15 keywords and 50 queries.

Table 1: Test results

name	average IO	average time per query, ms
I3	14.9	8.7
S2I	12.3	13.3
IR-tree	105.1	510

Further experiments will be done on a very large Twitter dataset collected in 2013 and 2014 containing geo-tagged tweets across the world.

The study [2] proposes an indexing method that doesn't require changing current system architectures (e.g., DBMS). At the moment, I am collaborated with Prof. Qiang Qu and Dr. Anders Skovosgaard to extend the study to support top-k queries, comparing with the above three methods. Moreover, I have already written the index construction algorithm based on key-value storage. For the moment comparison between Dr. Skovosgaard's algorithm results and my algorithm results is in progress.

5 Future work

Many studies can be conducted in this area. A possible topic that can be considered at this moment is incremental maintenance of the indexed data. Additionally, considering the challenge that the available data is growing, which requires parallel and efficient computation, many new software/hardware architectures can be considered, such as Spark, Hadoop, and GPUs.

As shown in the above sections, I am also interested in data query processing and data-intensive applications. Studies on various queries combining existing DBMSs and real applications (e.g., location prediction and incomplete data mining) will be considered for future work as well.

References

[1] Xin Cao, Lisi Chen, Gao Cong, Christian S Jensen, Qiang Qu, Anders Skovsgaard, Dingming Wu, and Man Lung Yiu. Spatial keyword querying. In *Conceptual Modeling*, pages 16–29. Springer, 2012.

[2] Anders Skovsgaard and Christian S Jensen. Top-k point of interest retrieval using standard indexes. In *Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 173–182. ACM, 2014.

[3] Antonin Guttman. *R-trees: a dynamic index structure for spatial searching*, volume 14. ACM, 1984.

[4] Justin Zobel and Alistair Moffat. Inverted files for text search engines. *ACM computing surveys (CSUR)*, 38(2):6, 2006.

[5] Dongxiang Zhang, Kian-Lee Tan, and Anthony KH Tung. Scalable top-k spatial keyword search. In *Proceedings of the 16th International Conference on Extending Database Technology*, pages 359–370. ACM, 2013.

[6] Lisi Chen, Gao Cong, Christian S Jensen, and Dingming Wu. Spatial keyword query processing: an experimental evaluation. *Proceedings of the VLDB Endowment*, 6(3):217–228, 2013.

[7] Ariel Cary, Ouri Wolfson, and Naphtali Rishe. Efficient and scalable method for processing top-k spatial boolean queries. In *Scientific and Statistical Database Management*, pages 87–95. Springer, 2010.

[8] Gao Cong, Christian S Jensen, and Dingming Wu. Efficient retrieval of the top-k most relevant spatial web objects. *Proceedings of the VLDB Endowment*, 2(1):337–348, 2009.

[9] João B Rocha-Junior, Orestis Gkorgkas, Simon Jonassen, and Kjetil Nørvåg. Efficient processing of top-k spatial keyword queries. In *Advances in Spatial and Temporal Databases*, pages 205–222. Springer, 2011.