R-tree Based Filtering Algorithms for Location-Aware Based System

Priya H. Jagtap, Prof. N. K. Zalte
PG Student, Professor, Computer Engineering Department,
Affiliated to kalyani charitable trust Late G.N.Sapkal College of Engineering,
Savitribai Phule, Pune University, Nasik, Maharashtra, India
Priyajgtp23@gmail.com nzalte@gmail.com

Abstract

Location Based services (LBS) have attracted and valuable attention from both industrial and academic communities. Existing LBS systems used a pull model also called as user-initiated model, from where a user can issues a query to server which responds with location-aware answers. A push model also called as server-initiated model is becoming an unavoidable computing model in next-generation LBS to provide users with instant replies. In the push model, subscribers register spatio-textual subscriptions to capture their interest of user. For Location-aware publish/subscribe, a system return result to achieve high-performance. This is achieve by using push model or server-initiated model. In push model or server-initiated model, subscribers register spatio-textual subscription that contain region based information[3] and return relevant result.

The challenge we have to resolve in a publish/subscribe system is to achieve high performance. A publish/subscribe system should support hundreds of millions of subscribers and deliver messages to relevant subscribers in milliseconds. Since messages and subscriptions contain both location information and textual description, it is significantly extent costly to deliver messages to relevant subscribers. This calls for an efficient filtering technique to support location-aware publish/subscribe services.

To resolve this problems, we use R-tree which is a token-based R-tree index structure that integrate each R-tree node with a set of tokens selected from subscriptions. Rt-tree is a balanced search tree. Every leaf node contains between a and A data entries, where
each entry tends to subscription. Each internal node has between a and A node entries. Each entry is a triple <Child, MBR, TokenSet>, where Child is a pointer to its child node, MBR is the minimum bounding rectangle of all entries within this child, and TokenSet is a set of tokens selected from subscriptions. A leaf node’s token set is the sum of tokens of all subscriptions within this node and an internal node’s token set is the sum of token sets of all entries within this node. As an entry corresponds to a node, for directness a node is mentioned exchangeable with its corresponding entry if the context is clear. Using the Rt-tree, we develop a filter-and-certification framework able to deliver a message. To reduce the number of tokens associated with Rt-tree nodes, we select some superiority representative tokens from subscriptions and collaborate them with Rt-tree nodes. This technique not only reduces index sizes but also improves the performance. This method achieves high. We go through following steps:

1. To formalize the location-aware publish/subscribe problem we introduce a new computing model.
2. We use novel index structure, the Rt-tree, by integrating superiority representative tokens selected from subscriptions into the R-tree nodes. Our method can support both conjunctive queries and ranking queries.
3. Using our proposed indexes, we develop efficient filtering algorithms and effective pruning techniques to improve the performance.
4. We develop FlexRP [5] Set Algorithm for to filtration to improve relevancy of result.

2. Existing System

Spatial Keyword Search method

Existing methods focus on finding relevant points-of-interest based on users’ locations and query keywords. [6] There are many existing studies on spatial keyword search [7, 8, 9]. One problem is knn based keyword search, which, given a query consisting of a location and a set of keywords, finds top-k relevant POIs by considering distance and textual relevance. Felipe et al. [10] integrated signature files into R-tree, and Cong et al. [11] combined inverted files and R-tree. Another problem is region-based keyword search, which, given a query consisting of a region and a set of keywords, finds the relevant POIs relevant to the keywords in the region. The methods addressing the problem also employed the R-tree index, and integrated inverted lists of keywords into R-tree nodes [7, 8, 9]. Our spatio-textual similarity search problem is appreciably different from the above-mentioned problems. The underlying data is a set of spatio-textual objects that exist regions and tokens. We focus on spatio-textual similarity between objects and queries and design efficient filtering algorithms for similarity search.

Keyword - Search method

Firstly it uses existing content based publish/subscribe techniques to generate the candidates that satisfy the textual constraint [6]. This method then verifies candidates used to check whether they satisfy spatial constraint. Evidently this method generates huge numbers of candidates and causes poor performance.

Spatial –Search method

This method first generates the location based candidates that satisfy the spatial constraint, using existing methods, e.g., segment tree or R-tree [11]. Then it filters candidates which do not satisfy textual constraint. This method also generates large numbers of candidates which causes poor performance.

Spatial keyword search based method

We can extend existing spatial keyword search methods, e.g., IRTree [1], to support our model. We construct an IRTree where IRTree visit from the root node to leaves. For each node, we use the minimum bounding rectangle to check the spatial constraint and utilize the inverted index to examine the textual constraint. However the IRTree has to visit many nodes unnecessarily and leads to low efficiency since if a node contains a token and satisfies the spatial constraint, it must access such node. To address this problem, we propose new indexes and filtering algorithms. In this paper, we focus on the in-memory setting to achieve instant performance.

Frequent pattern search method

An MinRPset algorithm find a minimum representative pattern set with error guarantee. MinRPset produces the smallest solution that we can possibly have in practice under the given problem setting, when the number of frequent closed patterns is below one million it takes a reasonable amount of time to finish. But MinRPset have some issues. They are very space-consuming and time-consuming on some compact datasets when the number of frequent closed patterns is large. To solve this problem, we propose another algorithm called FlexRPset, which provides one extra parameter K to allow users to make a trade-off between result size and efficiency[5].
3. Proposed System

We start implementation from giving input user query from where a user can send a detailed queries. Spatial dataset is input to system. In the standard R-tree, it doesn’t has textual pruning power, we use a token-based R-tree, called Rt-tree, by integrating tokens of subscriptions into R-tree nodes. Rt-tree is a balanced search tree. However an Rt-tree node may have large numbers of leaf descendants and it is expensive to check whether a node is a pivotal node. Based on this observation, we implement a filter-and-verification framework.

In the filter step, we find a set of candidate nodes which is a superset of pivotal nodes. In the verification step we verify the subscriptions in the leaf candidate nodes generated in the filter step. Efficient pruning techniques is used to achieve high performance.

RPLocal is developed based on FPclose [4]. It integrates frequent pattern mining with representative pattern finding. RPLocal is very efficient, but it produces more representative patterns than FlexRPset. Algorithm MinRPset is similar to RPglobal, but it utilizes several techniques to reduce running time and memory usage. MinRPset is only several times slower than RPLocal in most of the cases. Algorithm FlexRPSet is developed based on MinRPset. Algorithm FlexRPSet is developed based on MinRPset.

FlexRPSet, which provide one extra parameter K to allow users to make a trade-off between result size and efficiency. We adopt an incremental approach to let the users make the trade-off conveniently. MinRPset and FlexRPset produce fewer representative patterns than RPLocal when FlexRP set parameter equal to 1. This algorithm helps to improve scalability.

4. Advantages

1. Rt-tree based algorithm always achieved the highest performance for any types of messages.
2. Rt-tree seamlessly integrated the spatial and textual information and had large pruning power.
3. Rt-tree minimize filtering time.
4. Using FlexRP algorithm, result is more relevant and improve efficiency.

5. Conclusion

We study the location-aware publish/subscribe problem. We use an effective index structure Rt-tree by integrating textual description into R-tree nodes. Then implement reducing the number of tokens in each node. We implement efficient algorithm to directly find answer without verification steps. We use a filter-and-verification framework. Used algorithms support both conjunctive queries and ranking queries. We propose another algorithm called FlexRPSet, which provide one extra parameter K to allow users to make a trade-off between result size and efficiency. We adopt an incremental approach to let the users make the trade-off conveniently. MinRPset and FlexRPset produce fewer representative patterns than RPLocal – an efficient algorithm that is developed for solving the same problem and our method achieves high performance and good scalability.

References


