Location based Hierarchical Classifier for Multi-dimensional Spatial Data Mining

Miss. N. Meenatshi
Research Scholar
Department of Computer Science
SNR College of Arts and Science

Dr. K. Vijay Kumar PhD
Associate Professor
Department of Computer Science
SNR College of Arts and Science

Abstract

Geospatial data is becoming massive which leads to effective data management by compressing, updating and querying the data fields of the multidimensional spatial and temporal data. The explosion of both the data volumes and dimensionality of these geospatial field data makes the storage, management, query and processing a daunting challenge to existing solutions. In this work, hierarchical tensor decomposition based on the split-and-merge paradigm is developed for continuously compression and appending of multidimensional geospatial field data. Our goal is to propose a hierarchical data structure to reformulate and store the large volume of geospatial field data and to develop methods for data storage, query and computation support using this data structure. We illustrate this through a prototype implementation. The prototype has five components: 1) the design of a buffered hierarchical data structure and data decomposition strategies; 2) a proposal for a blocked data separation mechanism for splitting the huge tensors into small blocks according to the spatial-temporal reference; 3) a proposed algorithm that allows for data appending which is free of arithmetical operations and also computationally adaptive with continuous compression; 4) the development of a hierarchical structure-preserving and dimensional-independent data query which needs only to reform the row of the matrix in the leaf node; 5) the provision of computational operators such as tensor addition and linear operations, as well as a hierarchical structure-preserving computational framework.

Keywords : Location based Service, Publish and Subscribe System, Data Classification, Hierarchical data structure.

1.Introduction

Location based Service (LBS) exploring and exploiting in the internet market. Currently, LBS have been widely accepted because they can provide users with Location-aware experiences [1][2]. Up to date LBS systems employ a pull model or user-initiated model where a user issues a query to a server which responds with location aware answers. To provide users with instant replies, a push model or server-initiated model is becoming an inevitable computing model in next-generation location-based services. In the push model, subscribers register spatio-textual subscriptions [3][4] to capture their interests, and publishers post spatio textual messages. These calls for a high-performance location-aware publish/subscribe system to deliver messages to relevant subscribers. This computing model brings new user experiences to mobile users, and can help users retrieve information without explicitly issuing a query. One big challenge in a publish/subscribe system is to achieve high performance. A publish/subscribe system should support tens 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 rather costly to deliver messages to relevant subscribers. These calls for an efficient filtering technique to support location-aware publish/subscribe services. The explosion of both the data volumes and dimensionality of these geospatial field data makes the storage, management, query and processing a daunting challenge to existing solutions. In this work, hierarchical tensor decomposition based on the split-and-merge paradigm is developed for continuously compression and appending of multidimensional geospatial field data.

Our goal is to propose a hierarchical data structure[5] to reformulate and store the large volume of geospatial field data and to develop methods for data storage, query and computation support using this data structure. We illustrate this through a prototype implementation. The prototype has five components: 1) the design of a buffered hierarchical data structure and data decomposition strategies; 2) a proposal for a blocked data separation mechanism for splitting the huge tensors into small blocks according to the spatial-temporal reference; 3) a proposed algorithm that allows for data appending which is free of arithmetical operations and also computationally adaptive with continuous compression; 4) the development of a hierarchical structure-preserving and dimensional-independent data query [6]which needs only to reform the row of the matrix in the leaf node; 5) the provision of computational operators such as tensor addition[7] and linear operations[8], as well as a hierarchical structure-preserving computational framework. The rest of paper is organized as follows; section 2 explains the background knowledge regarding the related work. Section 3 explains and formulates the proposed System. The experimental results are discussed in section 4; we conclude the work with future work of the paper at section 5.

2. Related Work

2.1.Spatial keyword querying

With the proliferation of geo-positioning and geo-tagging, spatial web objects that possess both a geographical location and a textual description are gaining in prevalence, and spatial keyword queries that exploit both location and textual description are gaining in prominence. However, the queries studied so far generally focus on finding individual objects that each satisfies a query rather than finding groups of objects where the objects in a group collectively satisfy a query.

2.2.Efficient retrieval of the top-k most relevant spatial web objects

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 our knowledge, only naive techniques exist that is capable of computing a general web information retrieval query while also taking location into account. The framework leverages the inverted file for text retrieval and the R-tree for spatial proximity querying[9]. 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 accounts both text relevancy and location proximity to prune the search space.

2.3. Location-aware instant search

Location-Based Services (LBS) have been widely accepted by mobile users recently. Existing LBS-based systems require users to type in complete keywords. However for mobile users it is rather difficult to type in complete keywords on mobile devices. Index structure named as prefix region tree (called PR-Tree) which is used to efficiently support location aware instant search. PR-Tree[10] is a tree-based index structure which seamlessly integrates the textual description and spatial information to index the spatial data. Using the PR Tree is efficient algorithms can be modelled to support single prefix queries and multi-keyword queries for classifying the geospatial data.

3. Overview

Geo Spatial data Classification

We developed the Classification technique using the feature extraction methods based on the termination Criteria and data size. The data classifies into block tensor in terms of Computation balance aspects. To reduce the complexity, we initialized a list for storing the balanced tree data structure of each block during the decomposition.

Random selection

A naive method is to randomly select a representative token for each subscription. This method does not utilize the token distribution and can be optimized.

The df-based method

Based on time complexity is to improve the filtering performance, we should reduce the number of candidate nodes. Intuitively the smaller sizes of representative-token sets, the larger pruning power, and the smaller number of candidates[11]. To achieve high performance, it is important to reduce the sizes of the representative-token sets[12]. However we find the problem of minimizing the representative-token set is an NP-hard problem which can be proved by a reduction from the minimum hitting set problem or the set cover problem.

Hierarchical Tree Construction for Data Compression and Data Representation

The subspace combination of original tensor data can be represented as a tree structure, called dimension tree or subspace partition tree. With the subspace split of STR and the attribution dimension, the multidimensional geospatial field data can be split into blocks. Each block has its own spatial-temporal references.

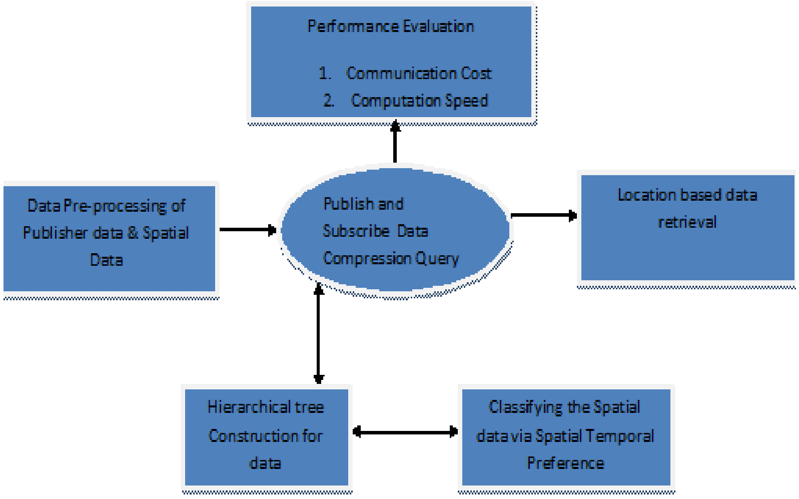

Performance Evaluation
1. Communication Cost
2. Computation Speed
Data Pre-processing of Publisher data & Spatial Data
Publish and Subscribe Data Compression Query
Location based data retrieval
Hierarchical tree Construction for data
Classifying the Spatial data via Spatial Temporal Preference

Figure 1 : Architecture Diagram of the Hierarchical publish and scribe System using push Method

In typical situations, the STR of multidimensional attribution fields is the same and is hidden in the data organization [13]. That is, the multiple attribution data are usually a separate data cube, which can be directly represented by a tensor. Two critical termination criteria, accuracy and the data size, are considered in the blocked hierarchical tensor representation. In the blocked hierarchical tensor representation, both the accuracy and the data size are first controlled by the block size and rank.

Algorithm 1-Filter formation

```
Input: Geospatial dataset
        Process: Tensor Formation Using Classification
                *Applying data pre-processing for geospatial data
                * Apply data Classification for Tensor formation
                *Apply Hierarchy based on the Data redundancy in Tensor
                CompRank ()
                * SVD modelling to Split and Compress the data based on the feature formation
Output: Block formation
```

Algorithm 2: Merging

```
Input: Tensors
        Process:
                *Calculate the Data Similarity between tensor
                If (Value > Threshold)
                Merge block
Output: Block merging to avoid the data redundancy and Memory Management
```

Algorithm 3: Parallel data analysis and querying Strategies based on Pruning

```
Input: Tensor Block
        Process: Query Strategy
                *Query Analysis is calculated for feature evolution determination between the data
                Feature evolution = absence of no. of the outlier data in the block
                Apply CompRank()
                        If (data === existing)
                                Compress ()
                        Else
                                *Establish a Unique Cluster
                                *Rank the data based on the uniqueness
Output: Data querying in Short span
```

Tree based index for Textual descriptions

As the standard R-tree has no textual pruning power, we propose a token-based R-tree, called Rt-tree, by integrating tokens of subscriptions into R-tree nodes. Rt-tree is a balanced search tree. Each leaf node contains between b and B data entries, where each entry is a subscription. Each internal node has between b and B node entries. Each entry is a triple hChild, MBR, TokenSeti, where Child is a pointer to its child node, [14][15] MBR is the minimum bounding rectangle of all entries within this child, and Token Set is a set of tokens selected from subscriptions (or a pointer to the token set). A leaf node's token set is the union of tokens of all subscriptions within this node and an internal node's token set is the union of token sets of all entries within this node. As an entry corresponds to a node [16], for simplicity a node is mentioned interchangeably with its corresponding entry if the context is clear.

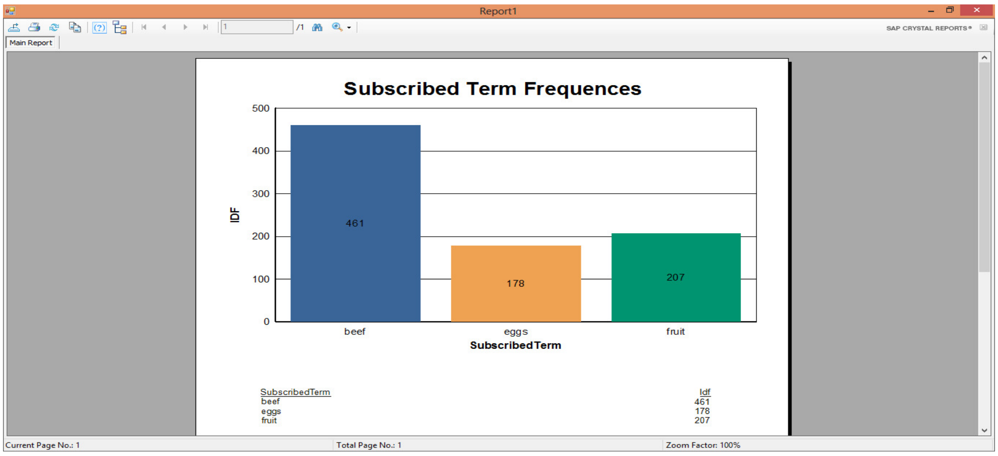
4. Experiential Results

In this section, we detailed analysis about the proposed using yelp dataset. Yelp dataset contains the user reviews of the restaurants, shopping, entertainment and nightlife services. It write reviews on their own locations, accusations of Yelp manipulating reviews to extort ad spending, concerns about the authenticity of reviews as well as the privacy and freedom of speech of reviewers.

Table 1 describes the performance of the technique in data classification

Technique	Throughput	Elapsed time	Long point Message	Short Point Message
R++ tree	100	9ms	600	250
Event based matching Algorithm	90	10ms	700	300

Instead we select different representative tokens for different ancestors and each ancestor contains a token and the performance of the filtering technique is depicted in the Table 1.



SubscribedTerm	Idf
beef	461
eggs	178
fruit	207

Figure 2. Evaluation of the Frequency of availability against the Subscriber interest terms

We evaluated Rt-tree with different token sets, i.e., Rt-tree with token sets, Rtþ-tree with representative tokens, Rtþþtree with multiple representative tokens.

Data representation can be used not only for data compression, updating and query, but also to support the computation with linear tensor operators will result in less memory utilization and less computation time. Three key indices, relative error, peak memory occupation and computation time, are used to benchmark the performances. The running time is also reduced from dozens of minutes to less than one minute with the blocking mechanism. The peak memory occupations and running times are affected by both the block size and the rank.

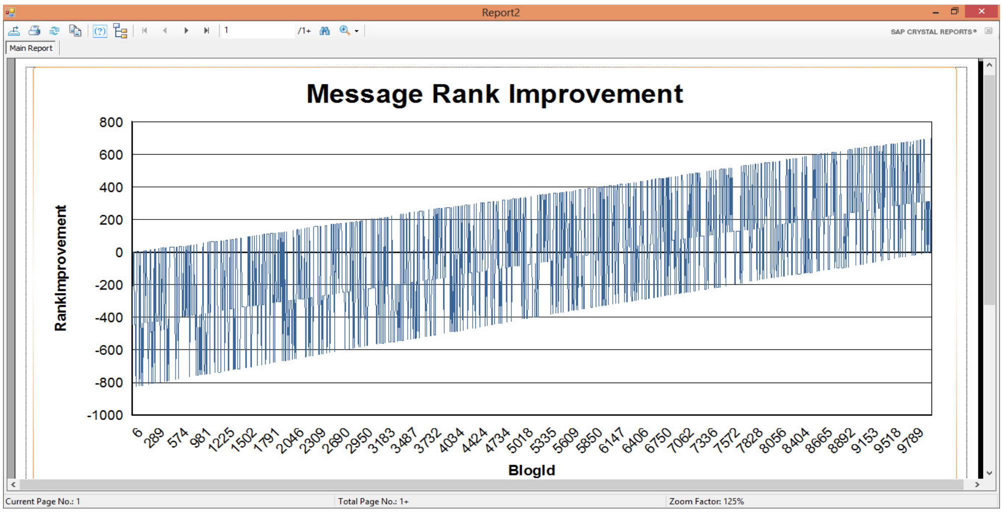


Figure 3 Message rank Improvement against the Different blog id

We can see that our method scaled very well, and with the increase of the numbers of subscriptions, the elapsed time increased sub linearly. This is because even if the number of subscriptions increased, our indexes still pruned large numbers of unnecessary subscriptions. We also evaluated the effect on updates of token frequencies. With the increase of frequencies, the performance slightly decreased. With the decrease of frequencies, the performance slightly increased. The new tokens had no effect on performance as they only enlarge hash table and efficiency of getting token frequency is not affected.

4. Conclusion

Proposed System design and implement an effective index structure R-tree by integrating textual description into R-tree nodes. We develop a filter-and-verification framework and devise efficient filtering algorithms. We propose reducing the number of tokens in each node which not only reduces index sizes but improves performance. We devise an efficient algorithm to directly find answers without the verification step. Algorithms are extending to support both conjunctive queries and ranking queries. We discuss how to support ranking semantics. Experimental results on real datasets show our method achieves high performance and good scalability.

5.Reference

[1] M. K. Aguilera, R. E. Strom, D. C. Sturman, M. Astley, and T. D. Chandra, “Matching events in a content-based subscription system,” in Proc. 18th Annu. ACM Symp. Principles Distrib. Comput., 1999, pp. 53–61.

[2] M. Altinel and M. J. Franklin, “Efficient filtering of XML documents for selective dissemination of information,” in Proc. 26th Int. Conf. Very Large Data Bases, 2000, pp. 53–64.

[3] B. Babcock, S. Babu, M. Datar, R. Motwani, and J. Widom, “Models and issues in data stream systems,” in Proc. 21st ACM SIGMOD-SIGACT-SIGART Symp. Principles Database Syst., 2002, pp. 1–16.

[4] X. Cao, G. Cong, and C. S. Jensen, “Retrieving top-k prestigebased relevant spatial web objects,” Proc. VLDB Endowment, vol. 3, no. 1, pp. 373–384, 2010.

[5] X. Cao, G. Cong, C. S. Jensen, and B. C. Ooi, “Collective spatial keyword querying,” in Proc. ACM SIGMOD Int. Conf. Manage. Data, 2011, pp. 373–384.

[6] X. Chen, Y. Chen, and F. Rao, “An efficient spatial publish/ subscribe system for intelligent location-based services,” in Proc. 2nd Int. Workshop Distrib. Event-Based Syst., 2003, pp. 1–6.

[7] Y.-Y. Chen, T. Suel, and A. Markowetz, “Efficient query processing in geographic web search engines,” in Proc. ACM SIGMOD Int. Conf. Manage. Data, 2006, pp. 277–288.

[8] G. Cong, C. S. Jensen, and D. Wu, “Efficient retrieval of the top-k most relevant spatial web objects,” Proc. VLDB, vol. 2, no. 1, pp. 337–348, 2009.

[9] P. Costa and G. P. Picco, “Semi-probabilistic content-based publish-subscribe,” in Proc. 25th IEEE Int. Conf. Distrib. Comput. Syst., 2005, pp. 575–585.

[10] G. Cugola and J. E. M. de Cote, “On introducing location awareness in publish-subscribe middleware,” in Proc. IEEE Int. Conf. Distrib. Comput. Syst. Workshops, 2005, pp. 377–382.

[11] Y. Diao and M. J. Franklin, “Query processing for high-volume XML message brokering,” in Proc. 29th Int. Conf. Very Large Data Base, 2003, pp. 261–272.

[12] P. T. Eugster, B. Garbinato, and A. Holzer, “Location-based publish/subscribe,” in Proc. 4th IEEE Int. Symp. Netw. Comput. Appl., 2005, pp. 279–282.

[13] P. T. Eugster, B. Garbinato, and A. Holzer, “Pervaho: A specialized middleware for mobile context-aware applications,” Electron. Commerce Res., vol. 9, no. 4, pp. 245–268, 2009.

[14] F. Fabret, H.-A. Jacobsen, F. Llirbat, J. Pereira, K. A. Ross, and D. Shasha, “Filtering algorithms and implementation for very fast publish/subscribe,” in Proc. ACM SIGMOD Int. Conf. Manage. Data, 2001, pp. 115–126.

[15] J. Fan, G. Li, L. Zhou, S. Chen, and J. Hu, “Seal: Spatio-textual similarity search,” Proc. VLDB Endowment, vol. 5, no. 9, pp. 824–835, 2012.

[16] I. D. Felipe, V. Hristidis, and N. Rishe, “Keyword search on spatial databases,” in Proc. Int. Conf. Data Eng., 2008, pp. 656–665.

[17] L. Fiege, F. C. G€artner, O. Kasten, and A. Zeidler, “Supporting mobility in content-based publish/subscribe middleware,” in Proc. USENIX Int. Conf. Middleware, 2003, pp. 103–122.