**RETRIEVING TOP-K DATA SET BASED ON NEIGHBORHOOD FEATURES AND THEIR DISTANCE**

Jacqueline Kennady1, B.Sathiya2

*Department of Computer Science*

*Dhanalakshmi Srinivasan Engineering College*

*Perambalur*

*India.*

[jacquelinekennady@gmail.com](mailto:eugeneberna31@gmail.com), [sathiyabalacse@gmail.com](mailto:sankar.winning@gmail.com)

1. PG Student 2. Project Guide

**Abstract:**

Top-k spatial preference queries return a ranked set of the k best data objects based on the scores of feature objects and shortest path distance in their spatial neighborhood. Despite the wide range of location-based applications that rely on spatial preference queries, existing algorithms incur non-negligible processing cost resulting in high response time. The reason is that computing the score of a data object requires examining its spatial neighborhood to find the feature object with highest score. Here a mapping of pairs of data and feature objects to a distance-score space, which in turn allows us to identify and materialize the minimal subset of pairs that is sufficient to answer any spatial preference query. For example, consider a relation with information on available restaurants, including their location, price range for one diner, and overall food rating. A user who queries such a relation might simply specify the user’s location and target price range, and expect in return the best 10 restaurants in terms of some combination of proximity to the user, closeness of match to the target price range, and overall food rating*.* Here I define spatial preference queries and propose appropriate indexing techniques and search algorithms for them.

**Index Terms**:-Query processing, Spatial Database

**1. INTRODUCTION**

With the popularization of geo tagging information, there has been an increasing number of Webs information systems specialized in providing interesting results through location-based queries. However, most of the existing systems are limited to plain spatial queries that return the objects present in a given region of the space. In this paper, we study a more sophisticated query that returns the best spatial objects based on the features (facilities) in their spatial neighborhood. Given a set of data objects of interest, a *top-*k *spatial preference query* returns a ranked set of the k best data objects. The score of a data object is defined based on the non-spatial score (quality) of feature objects in its spatial neighborhood.

On the other hand, the score of a feature object does not depend on its spatial location, but on the quality of the feature object. Such quality values can be obtained by a rating provider (e.g. www.zagat.com). For example, Figure 1 presents a spatial area containing data objects p (hotels) together with feature objects t (restaurants) and v (cafes) with their respective scores (e.g. rating). Consider a tourist interested in hotels with good restaurants and cafes in their spatial neighborhood. The tourist specifies a spatial constraint (in the figure depicted as a range around each hotel) to restrict the distance of the eligible feature objects for each hotel. Thus, if the tourist wants to rank the hotels based on the score of restaurants, the top-1 hotel is p3 (0.8) whose score 0.8 is determined by t4. However, if the tourist wants to rank the hotels based on cafes, the top-1 hotel is p1 (0.9) determined by v2. Finally, if the tourist is interested in both restaurants and cafes (e.g. summing the scores), the top-1 hotel is p2 (1.2).

Top-k spatial preference queries are intuitive and comprise a useful tool for novel location-based applications. Unfortunately, processing top-k spatial preference queries is complex, because it may require searching the spatial neighborhood of all data objects before reporting the top-k. Due to this complexity, existing solutions are costly in terms of both I/Os and execution time .

In this paper, I propose a novel approach for processing spatial preference queries efficiently. The main difference compared to traditional top-k queries are that the score of each data object is defined by the feature objects that satisfy a spatial constraint (for example range constraint). Therefore, pairs of data and feature objects need to be examined to determine the score of an object. Our approach relies on mapping of pairs of data and feature objects to a distance-score space, which in turn allows us to identify the minimal subset of pairs that is sufficient to answer all spatial preference queries. Capitalizing on the materialization of this subset of pairs, we present an efficient algorithm that improves query processing performance by avoiding examining the spatial neighborhood of data objects during query execution. In addition, we propose an efficient algorithm for materialization and describe useful properties that reduce the cost of maintaining the materialized information.



**Figure 1: Spatial area containing data and feature objects.**

**2. RELATED WORK**

Several approaches have been proposed for ranking spatial data objects. The reverse nearest neighbor (RNN) query was first proposed by Korn and Muthukrishnan . Then, Xia *et al.* studied the problem of retrieving the top-k most influential spatial objects , where the score of each spatial data object p is defined as the sum of the scores of all feature objects that have p as their nearest neighbor. Yang *et al.* studied the problem of finding an optimal location . The main difference compared to is that the optimal location can be any point in the data space and not necessarily an object of the dataset, while the score is computed in a similar way to.

The aforementioned approaches define the score of a spatial data object p based on the scores of feature objects that have p as their nearest neighbor and are limited to a single feature set. Differently, Yiu *et al.* first considered computing the score of a data object p based on feature objects in its spatial neighborhood from multiple feature sets. To this end, three different spatial scores were defined: range, nearest neighbor, and influence score; and differentalgorithms were developed to compute top-k spatial preference queries for these scores.

The algorithms developed by Yiu *et al.* assume that the data objects are stored in an R-tree based on spatial attributes, while the feature objects of each feature set are stored in a separate aggregate R-tree (aR-tree). The proposed algorithms can be divided into three categories. The first category is composed by probing algorithms, namely Simple (SP) and Group (GP) probing. These algorithms need to compute the score of all data objects before reporting the top-k result set. The second category is composed by Branch and Bound (BB) and Branch and Bound Star (BB\*) algorithms. These algorithms avoid computing the score of some data objects. The idea is computing an upper bound for each entry of the R-tree of the data objects, and prune the entries whose upper bound is smaller or equal to the score of the k-th data. 

**Figure 2: Examples of partial scores and spatial constraints**

**3. PROPOSED METHADOLOGY**

**a.Mapping to distance scorespace**

Top-k spatial preference queries return a ranked set of spatial data objects. The main difference to traditional top-k queries is that the score of each data object p is an element of O is obtained by the feature objects in its spatial neighborhood. Thus, determining the partial score of a data object p based on the feature set Fi requires that the *pairs* of objects (p, t) with t element of Fi need to be examined. Consequently, the *search space* that needs to be explored to determine the partial score is the Cartesian product between O and Fi. As the total number of pairs with respect to all feature datasets(Pc |O||Fi|) is significantly larger than the cardinality |O| of dataset O, processing top-k spatial preference queries is particularly challenging.

In this section, we formally define the search space of the top-kspatial preference queries by defining a mapping of the data objectsO and any feature dataset Fi to a *distance-score space*1. Then, weprove that only a subset of the pairs (p, t), where p O and t Fi,are sufficient to answer all top-k spatial preference queries. Thisdrastically reduces the search space for any given query, therebysaving computational costs significantly. In addition, we prove thatthis subset of pairs is the minimal subset of pairs necessary.

In a preprocessing step, the subset of pairs is computed andstored using a multi-dimensional index. As a result, we avoid computingpairs of the Cartesian product on-the-fly during query processing,leading to an efficient algorithm for processing top-k spatialpreference queries

**b. Query processing**

In this section, I present the *Skyline Feature Algorithm (SFA)* for processing top-k spatial preference queries. First, we present analgorithm that exploits the distance-score space and returns the dataobjects in descending order of their partial scores. Then, we presentthe algorithmic details of SFA, which produces the result of thetop-k spatial preference query by coordinating access to the partialscores of data objects. For ease of presentation, in the following,we refer to a pair (p, t), where p O and t Fi, as a *data point* indexed by

SFA computes the top-k spatial preference data objects progressively, by aggregating the partial scores of the data objects retrieved from each R-tree using NextObject algorithm. We use sum as the aggregate function in the following description and in the pseudocode.

Each time NextObject is invoked, the data object p with highest partial score is retrieved from P’inthus any unseen data object p’ in as a smaller partial score than p (. Therefore, we can compute an upper bound on the score of any data object p based on the highest partial scores of seen data objects in each .

SFA employs an upper bound Ui on the score of any unseen object in each heap Hi. Also, for each Hi, a list Li of seen objectsis maintained. Moreover, each time an object p is retrieved from Hi for the first time, p’s lower bound on score (p−) can be updated using the aggregate function (in this case sum). In addition, SFA maintains a list C of candidate data objects that may eventually become top-k results. C is sorted based on descending lower bound on score. In each iteration , SFA selects one heap Hi to retrieve the next data object p . The upper bound Ui on the score of Hi is set based on p’s partial score . Then, if p has not been seen before in Hi, its lower bound p− is updated based on the partial score and p is. added to Li . Notice that although p may be retrieved again from Hi, the maximum is. encountered at the first time, because Hi is accessed in descending order of score. In addition, p is added to the list C of candidate objects . Then, the upper bound (denoted as max) on the score of any object is computed in line 18. We can safely report as next top-k result, any object q in the top of the list C whose lower bound q− is greater than or equal to max . SFA continues in the same fashion, until k objects have been reported, or until all heaps are exhausted. In the latter case, if fewer than k objects have been reported, the objects in C are returned based on the sorting of C (because the lower bound now equals to the real score), until we have k objects .

**4.EXPIRIMENTAL RESULTS**

I evaluate my approach against GP and BB algorithms for processing nearest neighbor queries. There is no implementation of BB\* for nearest neighbor queries, and it is not trivial to adapt BB\* for these queries. I compare the effect of increasing the number of features c. Similar to other experiments where I evaluated the impact of varying c, SFA performs more efficiently. I evaluate the effect of increasing values of k. Again, SFA is two orders of magnitude better than BB and GP regardless of the exact value of k.

**5.CONCLUSIONS**

In this paper, I present a novel approach for boosting the performance of top-k spatial preference query processing. At the heart of our framework lies a mapping of pairs of data and feature objects to a distance-score space, which enables us to identify the minimal subset of pairs necessary to answer any ranked spatial preference query. By materializing this subset of pairs, we present efficient algorithms for query processing that result in improved performance. Furthermore, I describe an efficient algorithm for materialization and elaborate on useful properties that reduce the cost of maintenance. My experimental evaluation demonstrates that our approach reduces I/Os and response time by more than one order of magnitude compared to the state-of-the-art algorithms in most of the setups.

**9. REFERENCES**

1. C. B¨ohm, B. C. Ooi, C. Plant, and Y. Yan. Efficiently processing continuous k-nn queries on data streams. In *Proc. of Int. Conf. on*
2. *Data Engineering (ICDE)*, pages 156–165, 2007. S. B¨orzs¨onyi, D. Kossmann, and K. Stocker. The skyline operator. In *Proc. of Int. Conf. on Data Engineering (ICDE)*, page 421430, 2001.
3. S. Chaudhuri, N. N. Dalvi, and R. Kaushik. Robust cardinality and cost estimation for skyline operator. In *Proc. of Int. Conf. on Data* *Engineering (ICDE)*, page 64, 2006.
4. Y. Du, D. Zhang, and T. Xia. The Optimal-Location query. In *Proc. of the Int. Symposium on Spatial and Temporal Databases (SSTD)*,pages 163–180, 2005.
5. R. Fagin, A. Lotem, and M. Naor. Optimal aggregation algorithms for middleware. *Journal of Computer and System Sciences*, 66(4):614–656, 2003.
6. Guttman. R-trees: a dynamic index structure for spatial searching. In *Proc. of the Int. Conf. on Management of Data (SIGMOD)*, pages 47–57, Boston, Massachusetts, 1984.
7. J. Han, M. Kamber, and A. K. H. Tung. Spatial clustering methods in data mining: A survey. *Geographic Data Mining and Knowledge* *Discovery*, pages 1–29, 2001.
8. F. Korn and S. Muthukrishnan. Influence sets based on reverse nearest neighbor queries. In *Proc. of the Int. Conf. on Management of* *Data (SIGMOD)*, pages 201–212, 2000.
9. N. Mamoulis, M. L. Yiu, K. H. Cheng, and D. W. Cheung. Efficient top-k aggregation of ranked inputs. *ACM Transactions on Database* *Systems (TODS)*, 32(3):19, 2007.
10. D. Papadias, P. Kalnis, J. Zhang, and Y. Tao. Efficient OLAP operations in spatial data warehouses. In *Proc. of the Int. Symposium on Advances in Spatial and Temporal Databases (SSTD)*, pages 443–459. Springer-Verlag, 2001.