isrjournals

International Journal of Advanced Research in Computer Science Engineering and Information Technology

Volume: 6, Issue: 3,Special Issue: 1 ,Apr,2021 ,ISSN_NO: 2321-3337

CUSTOMER RELATIONSHIP MANAGEMENT

Ms. Varalakshmi M.E CSE

Mr.B.Arunmozhi M.E., Professor of Computer science Department St. Joseph college of Engineering, Sriperumbudur, Chennai

Abstract

Call center management is an increasingly important skill as the use of call centers becomes a popular method of centralizing information services, streamlining order taking and providing valuable customer support.

The skills required to successfully set-up and manage a call center encompass everything from staff recruitment and personnel management, to technical understanding of the options available, and the all important customer relationship management.

From small customer service departments to large call centers, the importance of developing successful call center management is vital for building a valued relationship with customers to support long-term business growth.

This system (Call Center Management) is useful to the organization, it maintains the information about the employees and it also contains the necessary information of the customer and their phone Numbers, their services also. It also maintains the employee roaster details.

With the proliferation of spatial- textual data such as location-based services and geo-tagged websites, spatial keyword queries are ubiquitous in real life. One example of spatial- keyword query is the so-called *collective spatial keyword query* (CoSKQ) which is to find for a given query consisting a query location and several query keywords a set of objects which *covers* the query keywords collectively and has the smallest *cost* with the query location.

In the literature, many different functions were proposed for defining the *cost* and correspondingly, many different approaches were developed for the CoSKQ problem.

In this paper, we study the CoSKQ problem systematically by proposing *a unified cost function* and *a unified approach* for the CoSKQ problem (with the unified cost function).

The unified cost function includes all existing cost functions as special cases and the unified approach solves the CoSKQ problem with the unified cost function in a unified way.

Experiments were conducted on both real and synthetic datasets which verified our proposed approach. We also proposed with rating based and location based for searching the nearest place in the unified manner. Experiments were conducted.

Index Terms—Spatial keyword queries, unified framework

1. Introduction

NOWA DAYS, geo-textual data which refers to data with both spatial and textual information is ubiquitous. Some examples of geo- textual data include the spatial points of interest (POI) with textual description (e.g., restaurants, cinema, tourist attractions, and hotels), geo-tagged web objects (e.g., web pages and photos at Flickr), and also geo-social networking data (e.g., users of Four Square have their check-in histories which are spatial and also profiles which are textual).

In this project, we discuss the CoSKQ downside consistently by proposing a unified value operate and a unified approach for the CoSKQ downside (with the unified cost function). One scenario of this application is that a tourist wants to find several POIs such that she/he could do site- seeing, shopping and dining and the POIs are close to the hotel. In this case, the user can set the query location to the hotel location and the query keywords to be “attractions” ”shopping” and “restaurants” to search for a set of POIs. Another scenario is that a manager wants to set up a project consortium of partners close to each other such that they together offer the capabilities required for successful execution of the whole project. In this case, the user can issue the query with his/her location as the query location and the required skills for the partners as the query keywords to find a group of people.

The unified value operate includes all existing value functions as special cases and also the unified approach solves the CoSKQ downside with the unified value operate in a much-unified manner as reviewing our user preferences as location-based searching or rating based searching. If it is location- based search it views as nearest places with the accurate latitude and longitude directions. If it is rating based search, showing the result as best and highest rating at the first places.

It uses the unified cost function algorithm to form sequences of the nearest places.

1.1 Different approaches

In this paper, we study the CoSKQ problem systematically by proposing a unified cost function and a unified approach for the CoSKQ problem (with the unified cost function).

Without the unified approach, we need to handle different cost functions by different algorithms, which increase the difficulty for CoSKQ to be used in practice. Also, when researchers work on improving the performance of an algorithm, only the corresponding cost function is benefited.

Although sometimes it is possible that one algorithm originally designed for one cost function can be adapted for another cost function, the performance of the adapted algorithm is not satisfactory. A better idea is to have a unified cost function and a unified approach, where the unified cost function captures all known cost functions and some other cost functions which are not known before but useful.

1.2 A unified cost function

We propose a unified cost function cost unified which expresses all existing cost functions and a few new cost functions that have not been studied before. The core idea of cost unified is that first two distance components, namely the query-object distance component and the object- object distance component, are defined, where the former is based on the distances between the query location and those of the objects and the latter is based on the pair wise distances among the set of objects.

Then cost unified is defined based on the two distance components carefully such that all existing cost functions are captured (Note that this is possible since all ingredients of defining a cost function are distances between the query location and those distances among objects which are captured by the two components).

1.3 A unified approach.

We design a unified approach, which consists of one exact algorithm and one approximate algorithm, for the CoSKQ problem with the unified cost function. For the CoSKQ problem with the cost function instantiated to those existing cost functions, which have been proved to be NP-hard, our exact algo- rithm is superior over the state-of-the-arts in that it not only has a unified procedure, but also runs faster under all settings for some cost functions (e.g., costMinMax and costMinMax2) and under the majority of settings for the other cost functions, and our ap- proximate algorithm is always among those algorithms which give the best approximation ratios and runs faster than those algorithms which give similar approximation ratios. For the CoSKQ problem with the cost function instantiated to those new cost functions that have not been studied before, our exact algorithm runs reasonably fast and our approximate algorithm provides certain approximation ratios.

isrjournals

International Journal of Advanced Research in Computer Science Engineering and Information Technology

Volume: 6, Issue: 3,Special Issue: 1 ,Apr,2021 ,ISSN_NO: 2321-3337

Besides, we conducted extensive experiments based on both real and synthetic datasets which verified our unified approach.

The rest of this paper is organized as follows. Section 2 gives the related work. Section 3 introduces the unified cost function and Section 4 presents the unified approach for CoSKQ. Section 5 gives the empirical study and Section 6 concludes the paper.

2 A UNIFIED APPROACH

In this section, we introduce our unified approach which consists of one exact algorithm called Unified-E (Section 4.1) and one approximate algorithm called Unified-A (Section 4.2). While the unified cost function combines existing ones, our unified approach is not one which simply combine existing approaches. In fact, both the exact algorithm and approximate algorithm proposed in this paper are clean and elegant while existing approaches have quite different structures.

Before presenting the algorithms, we first give some definitions as follows. Given a query q and an object o in, we say o is a relevant object if o.ψ q.ψ . We denote q to be the set of all relevant objects. Given a set S of objects, S is said to be a feasible set if S covers q.ψ (i.e. q.ψ o∈So.ψ). Note that the CoSKQ problem is to find a feasible set with the smallest cost.

Given a non-negative real number r, we denote the circle centered at q.λ with radius r by C(q, r). Similarly, the circle centered at o.λ with radius r is denoted by C(o, r).

Let q be a query and S be a feasible set. We say that an object o S is a query-object distance contributor wrt S if d(o, q) contributes in Dq,o(S φ1). Specifically, we have the following three cases according to the value of φ1.

In the case of **φ1=1** where **Dq,o(S φ1) =o S d(o, q)**, each object in S is a query- object distance contributor wrt S;

In the case of **φ1 = ∞** where **Dq,o(S|φ1) = max o∈S d(o, q)**, only those objects in S which have the maximum distance from q are the query-object distance contributors wrt S;

In the case of **φ1 = −∞** where **Dq, o(S|φ1) = mino∈S d(o, q)**, only those objects in S which have the minimum distance from q are the query-object distance contributors wrt S.

Then, we define the key query-object distance contributor wrt S to the object with the greatest distance from q among all query- object distance contributors wrt S. The concept of "key query- object distance contributor" is inspired by the concept of "query distance owner" proposed in [17], and the concept of "key query- object distance contributor" is more general in the sense that a query distance owner corresponds to a key query distance contributor in the case of φ1 = but not in other cases.

Let S be a set of objects and oi and oj are two objects in S.

We say that oi and oj are object- object distance contributors wrt S if d(oi, oj) contribute in maxo,ot ∈S d(o, oJ), i.e, (oi, oj) = arg maxo,ot ∈S d(o, oJ).

Given a query q and a keyword t, the t-keyword nearest neighbour of q, denoted by NN (q, t), it defined to be the nearest neighbor (NN) of q containing keyword t .

Similarly, NN (o, t) is defined to be the NN of o containing keyword t. Besides, we define the nearest neighbor set of q, denoted by N (q) to be the set containing q's t-keyword nearest neighbor for each t ∈ q.ψ, i.e., N (q) = ∪t∈q.ψNN (q, t). Note that N

(q) is a feasible setGovernments and International Organizations.

2.1 Algorithm1 A Unified Approach

Input: A query q, a set O of objects and a unified cost function cost unified (S|α,φ1, φ2)

1: curSet ← N (q)

2: curCost ← cost(curSet)

3: RS ← C(q, r1)

4: P ← a set of all relevant object pairs (oi, oj) where oi, oj ∈ RS and dLB ≤ d(oi, oj) < dUB

5: for each (oi, oj) ∈ P in ascending order of cost({oi, oj})

do

Volume: 6, Issue: 3,Special Issue: 1 ,Apr,2021 ,ISSN_NO: 2321-3337

6: if cost(oi, oj)LB > curCost then

7: break;

8: Rij C(oi, d(oi, oj))C(oj, d(oi, oj))

9: a set of all relevant objects om Rij where rLB d(om, q) rUB

10: for each om in ascending order of d(om, q)

do

11: SJ findBestFeasibleSet(oi, oj, om)

12: if SJ =andcost(SJ)< curCost then

13: curSet SJ

14: curCost cost(SJ)

15: return curSet

Proof. The proof of rLB is shown as follows. When φ1 ∈ {1, ∞},

RS and dLB ≤ d(oi, oj) < dUB

5: for each (oi, oj) ∈ P in ascending order of cost({oi, oj})LB

d(om, q) > df because otherwise S is not a feasible set. For costSumMax, costMaxMax and costSum, we do not need to con- sider an object o if d(o, q) < max d(oi, q), d(oj, q) because it cannot be the key query- object distance contributor of S by definition.

Similarly,for costMaxMax2, we do not need to con- sider object o if d(o, q) < d(oi, oj) because it cannot be the key query-object distance contributor of S. When φ1 = , we set rLB = df d(oi, oj) because otherwise S is not a feasible set.

The proof of rUB is shown as follows. For costSumMax and costMaxMax, if S contains an object o with d(o, q) curCost d(oi, oj), it is obvious that cost(S) curCost.

Similarly, for costMaxMax2 (costSum), if S contains an ob- ject o with d(o, q)curCost (d(o, q) > curCost d(oi, q)d(oj,q)),cost(S) curCost.

For costMinMax and costMinMax2, we do not need to consider an object o if d(o, q) min d(oi, q), d(oj, q) because it cannot be the key q̄uery-object distance contributor of S by definition.

Also, in costMinMax, if d(o, q) ≥ curCost - d(oi, oj), cost(S) ≥ curCost.

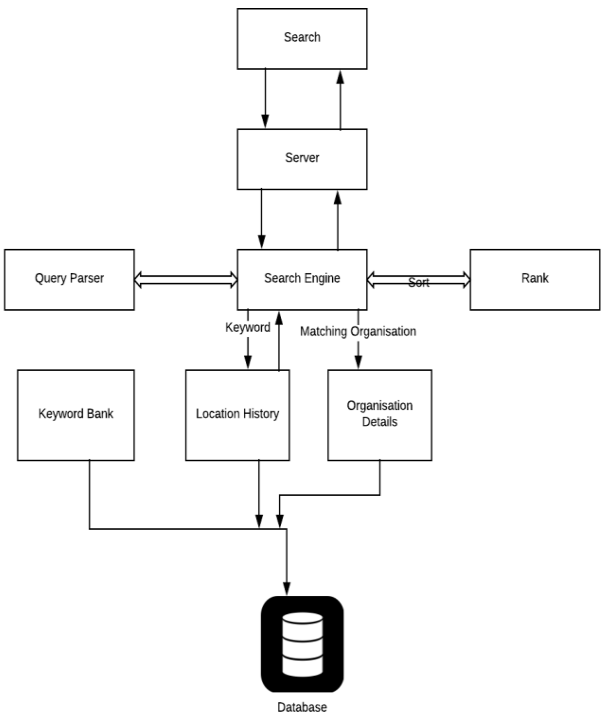
isrjournals

International Journal of Advanced Research in Computer Science Engineering and Information Technology

Volume: 6, Issue: 3,Special Issue: 1 ,Apr,2021 ,ISSN_NO: 2321-3337

With the above search strategy introduced, we present the Unified-E algorithm in Algorithm 1. Specifically, we maintain an object set curSet for storing the best-known solution found so far, which is initialized to N (q) (line 1), and curCost to be the cost of curSet. Recall that N (q) is a feasible set. Then, we initialize RS to be C(q, r1) and find a set P of all object pairs (oi, oj) where oi and oj are in RS to take the roles of object-object distance contributors.

Figure.1 Architecture diagram



Advantages

- Searching easily.
- Easy to search nearest places.
- Time saving.

Disadvantages

- Show only one nearest place.
- No unique cost.
- Retrieving process to be delay.

3. CONCLUSION

In this project, we tend to planned a unified value perform for CoSKQ. This cost perform expresses all existing value functions within the literature and a number of value functions that haven't been studied before. We designed a unified approach that consists of 1 actual rule and one approximate rule. The precise rule runs comparably quick because the existing actual algorithms, whereas the approximate algorithm provides a comparable approximation quantitative relation as the existing approximate algorithms. In depth experiments were conducted that verified our theoretical findings.

4. REFERENCES

[1] Y. Zeng, X. Chen, X. Cao, S. Qin, M. Cavazza, and Y. Xiang. Optimal route search with the coverage of users' preferences. In IJCAI, pages 2118–2124, 2015.

[2] C. Zhang, Y. Zhang, W. Zhang, and X. Lin. Inverted linear quadtree: Efficient top k spatial keyword search. In ICDE, pages 901–912. IEEE, 2013.

[3] D. Zhang, Y. M. Chee, A. Mondal, A. Tung, and M. Kitsuregawa. Keyword search in spatial databases: Towards searching by document. In ICDE, pages 688–699. IEEE, 2009.

[4] D. Zhang, B. C. Ooi, and A. K. H. Tung. Locating mapped resources in web 2.0. In ICDE, pages 521–532. IEEE, 2010.

[5] D. Zhang, K.-L. Tan, and A. K. H. Tung. Scalable top-k spatial keyword search. EDBT/ICDT, pages 359–370. ACM, 2013.

[6] D. Wu, M. L. Yiu, C. S. Jensen, and G. Cong. Efficient continuously moving top-k spatial keyword query processing. In ICDE, pages 541–552. IEEE, 2011.