2019 IEEE 35th International Conference on Data Engineering (ICDE)

On Generalizing Collective Spatial Keyword Queries (Extended Abstract)

Harry Kai-Ho Chan*, Cheng Long†, Raymond Chi-Wing Wong*
*The Hong Kong University of Science and Technology
†Nanyang Technological University
*{khchanak, raywong}@cse.ust.hk, †c.long@ntu.edu.sg

***Abstract*—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* wrt the query location. Quite a few cost functions have been proposed for CoSKQ and correspondingly, different approaches have been developed. However, given these cost functions in different forms and approaches in different structures, one could hardly compare existing cost functions systematically and needs to implement all approaches in order to tackle the CoSKQ problem with different cost functions, which is effort-consuming. In this paper, we design a unified cost function which generalizes the majority of existing cost functions for CoSKQ and develop a unified approach which works as well as (and sometimes better than) best-known approaches based on different cost functions. Experiments were conducted on both real and synthetic datasets which verified our proposed approach.**

I. Introduction

Nowadays, 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 and tourist attractions) and geo-tagged web objects (e.g., webpages and photos at Flickr). One application based on geo-textual data is to search a set of (geo-textual) objects wrt a query consisting of a query location (e.g., the location that the user is located at) and some textual information (e.g., some keywords expressing the targets the user wants to search) such that the objects have their textual information *matching* the query keywords and their locations close to the query location.

The above application was captured by the so-called *Collective Spatial Keyword Query* (CoSKQ) [2], [4], [1]. Given a query with a location and a set of keywords, the CoSKQ problem is to find a set S of objects such that S covers all query keywords, and the *cost* of S, denoted by $cost(S)$, is minimized. In the literature, many different cost functions have been proposed for $cost(S)$ in the CoSKQ problem, and these cost functions are applicable in different scenarios. For the CoSKQ problem with each particular cost function, at least one solution (including an exact algorithm and an approximate algorithm) was developed, and these solutions usually differ from one another. Usually, an existing algorithm for the CoSKQ problem with a particular cost function cannot be used to solve that with another cost function. As a result, we need to handle different cost functions by different algorithms, which increases the difficulty for CoSKQ to be used in practice. Besides, 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 as many known cost functions as possible.

The main contribution of this paper is summarized as follows. (1) We propose a unified cost function $cost_{unified}$ which expresses the majority of existing cost functions and a few new cost functions that have not been studied before. (2) We design a unified approach, which consists of one exact algorithm and one approximate algorithm, for the CoSKQ problem with $cost_{unified}$. For the CoSKQ problem with the cost function instantiated to those existing cost functions, which have been proved to be NP-hard, our exact algorithm outperforms the state-of-the-art. 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. (3) We conducted extensive experiments based on both real and synthetic datasets which verified our unified approach.

II. A Unified Cost Function

Let $\mathcal{O}$ be a set of objects, where each object $o \in \mathcal{O}$ is associated with a spatial location $o.\lambda$ and a set of keywords $o.\psi$. Given two objects o_1 and o_2, we denote by $d(o_1, o_2)$ the Euclidean distance between $o_1.\lambda$ and $o_2.\lambda$.

(1) Problem definition. A *collective spatial keyword query* (CoSKQ) is defined as follows.

Problem 1 (CoSKQ [3]): Given a query q with a location $q.\lambda$ and a set of keywords $q.\psi$, **the CoSKQ problem** is to find a set S of objects such that S covers $q.\psi$, i.e., $q.\psi \subseteq \cup_{o \in S} o.\psi$, and the *cost* of S, denoted by $cost(S)$, is minimized. □

(2) Existing cost functions. We focus on five existing cost functions proposed in the literature for defining $cost(\cdot)$ in the CoSKQ problem, namely $cost_{Sum}$ [2], $cost_{SumMax}$ [1], $cost_{MaxMax}$ [2], $cost_{MaxMax2}$ [4], and $cost_{MinMax}$ [1].

(3) A unified cost function $cost_{unified}$. In this paper, we propose a *unified* cost function $cost_{unified}$ which could be instantiated to many different cost functions including all those five existing ones. We first introduce the *query-object distance*

2375-026X/19/$31.00 ©2019 IEEE
DOI 10.1109/ICDE.2019.00252

	Parameter			$cost_{unified}(S\|\alpha,\phi_1,\phi_2)$	Existing/New	Unified-A Appro. ratio	Best-known Appro. ratio
	$\alpha \in (0,1]$	$\phi_1 \in \{1,\infty,-\infty\}$	$\phi_2 \in \{1,\infty\}$				
a	0.5*	1	1	$\sum_{o\in S} d(o,q) + \max_{o_1,o_2\in S} d(o_1,o_2)$	$cost_{SumMax}$ [1]	$2H_{\|q.\psi\|}$	N.A.
b	0.5*	1	∞	$\max\{\sum_{o\in S} d(o,q), \max_{o_1,o_2\in S} d(o_1,o_2)\}$	$cost_{SumMax2}$ (New)	$H_{\|\psi\|}$	$H_{\|q.\psi\|}$ [1]
c	0.5*	∞	1	$\max_{o\in S} d(o,q) + \max_{o_1,o_2\in S} d(o_1,o_2)$	$cost_{MaxMax}$ [2], [4], [1]	1.375	1.375 [4]
d	0.5*	∞	∞	$\max\{\max_{o\in S} d(o,q), \max_{o_1,o_2\in S} d(o_1,o_2)\}$	$cost_{MaxMax2}$ [4]	$\sqrt{3}$	$\sqrt{3}$ [4]
e	0.5*	$-\infty$	1	$\min_{o\in S} d(o,q) + \max_{o_1,o_2\in S} d(o_1,o_2)$	$cost_{MinMax}$ [1]	2	3 [1]
f	0.5*	$-\infty$	∞	$\max\{\min_{o\in S} d(o,q), \max_{o_1,o_2\in S} d(o_1,o_2)\}$	$cost_{MinMax2}$ (New)	2	N.A.
g	1	1	-	$\sum_{o\in S} d(o,q)$	$cost_{Sum}$ [2], [1]	$H_{\|\psi\|}$	$H_{\|q.\psi\|}$ [1]
h	1	∞	-	$\max_{o\in S} d(o,q)$	$cost_{Max}$ (New)	1	N.A.
i	1	$-\infty$	-	$\min_{o\in S} d(o,q)$	$cost_{Min}$ (New)	1	N.A.

* Following the existing studies, $\alpha = 0.5$ is used to illustrate the case of $\alpha \in (0,1)$ for simplicity

TABLE I: $cost_{unified}$ under different parameter settings

component, denoted by $D_{q,o}(S|\phi_1)$, which is defined based on the distances between the query location and the objects in S.

$$D_{q,o}(S|\phi_1) = [\sum_{o\in S}(d(o,q))^{\phi_1}\ ]^{\frac{1}{\phi_1}}$$

where $\phi_1 \in \{1,\infty,-\infty\}$ is a user parameter. Depending on the setting of ϕ_1, $D_{q,o}(S|\phi_1)$ corresponds to the summation, the maximum, or the minimum of the distances from the query location to the objects in S. We define $cost_{unified}$ as follows.

$$cost_{unified}(S|\alpha,\phi_1,\phi_2) = \{[\alpha \cdot D_{q,o}(S|\phi_1)]^{\phi_2} + [(1-\alpha)\max_{o_1,o_2\in S} d(o_1,o_2)]^{\phi_2}\}^{\frac{1}{\phi_2}}$$

where $\alpha \in (0,1]$[1], $\phi_1 \in \{1,\infty,-\infty\}$ and $\phi_2 \in \{1,\infty\}$ are user parameters. The instantiations of $cost_{unified}$ depending on different parameter settings are shown in Table I.

(4) Intractability results. We have the following result.

Theorem 1 (Intractability): The CoSKQ problem is NP-hard with all possible parameter settings of α, ϕ_1 and ϕ_2 except for the setting of $\alpha = 1$ and $\phi_1 \in \{\infty,-\infty\}$. □

III. A Unified Approach

In this section, we introduce our unified approach which consists of one exact algorithm called *Unified-E* and one approximate algorithm called *Unified-A*.

Given a query q and an object o in $\mathcal{O}$, we say that o is a **relevant object** if $o.\psi \cap q.\psi \neq \emptyset$. Given a set S of objects, S is said to be a **feasible set** if S covers $q.\psi$ (i.e. $q.\psi \subseteq \cup_{o\in S} o.\psi$). We say that an object $o \in S$ is a **query-object distance contributor** wrt S if $d(o,q)$ *contributes to* $D_{q,o}(S|\phi_1)$. 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. Let o_i and o_j be two objects in S. We say that o_i and o_j are **object-object distance contributors** wrt S if $d(o_i,o_j)$ *contribute to* $\max_{o,o'\in S} d(o,o')$, i.e, $(o_i,o_j) = \arg\max_{o,o'\in S} d(o,o')$.

(1) Unified-E. The idea of *Unified-E* is to iterate through the object-object distance contributors and search for the best feasible set S' in each iteration. This allows CoSKQ with different cost functions to be executed efficiently. Note that each existing algorithm [2], [4], [1] is designed for a specific cost function and they cannot be used to answer CoSKQ with different cost functions. We develop effective pruning techniques, which consider different parameter settings, to prune objects (objects pairs) that cannot be the key query-object distance contributors (object-object distance contributors).

[1] In the setting of $\alpha = 0$, the query location has no contribution to the cost. Thus, we do not consider this setting.

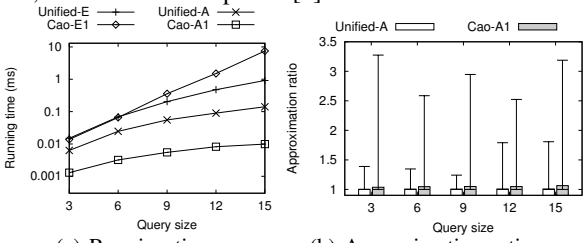
(2) Unified-A. Compared with *Unified-E*, *Unified-A* avoids the step of finding object-object distance contributors and replaces the expensive step of constructing the best feasible set with an efficient step of constructing the (arbitrary) feasible set, and thus it enjoys significantly better efficiency. Table I shows the approximation ratios of *Unified-A* for different parameter settings, where $|\psi| < |q.\psi|$. The *Unified-A* algorithm provides *better* (same) approximation ratios than (as) the state-of-the-arts for three (two) existing cost functions.

IV. Empirical Studies

Datasets. We used three real datasets in our experiments, namely Hotel, GN and Web. Dataset Hotel contains $20,790$ hotels in the U.S. (www.allstays.com), each of which has a spatial location and a set of words that describe the hotel. It contains $80,645$ words in total with 602 unique words. Some details of the other two datasets could be found in [2].

Algorithms. Both *Unified-E* and *Unified-A* are studied. For comparison, for the CoSKQ problem with an existing (a new) cost function, the state-of-the-arts (adaptions) are used.

We evaluated the running time and approximation ratio (for approximate algorithms only). The results for $cost_{MinMax2}$ are shown in Figure 1. According to the results, *Unified-E* runs faster than *Cao-E1*, and *Unified-A* gives better approximation ratios than *Cao-A1* with reasonable efficiency. Due to the page limit, more results are put in [3].



(a) Running time (b) Approximation ratio

Fig. 1: Effect of $|q.\psi|$ on $cost_{MinMax2}$ (Hotel)

Acknowledgments

The research of Harry Kai-Ho Chan and Raymond Chi-Wing Wong is supported by HKRGC GRF 14205117.

References

[1] X. Cao, G. Cong, T. Guo, C. S. Jensen, and B. C. Ooi. Efficient processing of spatial group keyword queries. *TODS*, 40(2):13, 2015.

[2] X. Cao, G. Cong, C. S. Jensen, and B. C. Ooi. Collective spatial keyword querying. In *SIGMOD*, pages 373–384. ACM, 2011.

[3] H. K.-H. Chan, C. Long, and R. C.-W. Wong. On generalizing collective spatial keyword queries. *TKDE*, 30(9):1712–1726, 2018.

[4] C. Long, R. C.-W. Wong, K. Wang, and A. W.-C. Fu. Collective spatial keyword queries:a distance owner-driven approach. In *SIGMOD*, pages 689–700. ACM, 2013.