On Generalizing Collective Spatial Keyword Queries

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? 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.

**EXISTING SYSTEM**

* In the existing system, the system studies on spatial keyword queries focus on finding an object set as a solution. Among them, some [3], [17], [2] studied the collective spatial keyword queries (CoSKQ). Caoet al. [3], [2] proposed four cost functions, namely costSum, costMaxMax, costMinMax and costSumMax, and developed algorithms for the CoSKQ problem with the first three cost functions, leaving that with the fourth cost function, i.e., costSumMax, as future work. Besides, they studied two variations of CoSKQ, namely top-k CoSKQ and weighted CoSKQ, in [2]. Long et al. [17] proposed exact and approximate algorithms for the CoSKQ problem with costMaxMax and also that with a new cost function costMaxMax2.
* Another query that is similar to the CoSKQ problem is the mCK query [28], [29], [14] which takes a set of m keywords as input and finds m objects with the minimum diameter that cover the m keywords specified in the query. In the existing studies of mCK queries, it is usually assumed that each object contains a single keyword. There are some variants of the mCK query, including the SK-COVER [7] and the BKC query [10]. These queries are similar to the CoSKQ problem in that they also return an object set that covers the query keywords, but they only take a set of keywords as input. In contrast, the CoSKQ problem studied in this paper takes both a set of keywords and a spatial location as inputs.

**Disadvantages**

* + There is no unified framework to process bulk spatial database data.
	+ There is no unified cost function to process spatial quires.

**PROPOSED SYSTEM**

* The system proposes 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 and 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.).
* **A unified approach**. The system designs 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 algorithm 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 approximate 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.

**Advantages**

* The proposed system designs a unified approach, which consists of one exact algorithm and one approximate algorithm, for the CoSKQ problem with the unified cost function.
* Fast techniques in Key Query-Object Distance Contributor Finding.

**SYSTEM REQUIREMENTS**

➢ **H/W System Configuration:-**

➢ Processor - Pentium –IV

➢ RAM - 4 GB (min)

➢ Hard Disk - 20 GB

➢ Key Board - Standard Windows Keyboard

➢ Mouse - Two or Three Button Mouse

➢ Monitor - SVGA

**Software Requirements:**

* Operating System - Windows XP
* Coding Language - Java/J2EE(JSP,Servlet)
* Front End - J2EE
* Back End - MySQL