



CusFinder: An Interactive Customer Ranking Query System

Yanghao Zhou, Xiaolin Qin^(✉), Xiaojun Xie, and Xingluo Li

Nanjing University of Aeronautics and Astronautics, Nanjing, China
{zyhreadboy, qinxcs, xiexj, lxldblab}@nuaa.edu.cn

Abstract. Finding a certain number of objects with optimum ranking based on spatial position constraints and given preference of attributes can be essential in numerous scenarios. In this paper, we demonstrate CusFinder, an interactive customer ranking query system to retrieve customers who favour a specific seller more than other people from the perspective of sellers, instead of retrieving sellers for a given customer similar to existing commercial systems. To make the query processing more efficient, a novel indexing is proposed to serve for query engine. Furthermore, we present the result of queries upon multiple visualization views with user-friendly interaction designs.

Keywords: Optimum ranking · Spatial constraints · Multiple views

1 Introduction

As an effective means of retrieving a specified number of objects according to ranking scores, rank-aware query is widely applied in various real-life scenarios. Nevertheless, we have noticed that none of existing work takes into account both sellers' perspective and customer-oriented ranking with spatial constraint, such as the following example.

Example 1.1 Given a collection of restaurants and a set of customers, the manager of one of the restaurants, named *Tom*, is trying to find a small number (denoted by k) of customers who prefer his restaurant compared with other customers. As different customers have different requirements (i.e. taste or price), and customers would not consider those restaurants far away (denoted by σ) from them, *Tom* is supposed to know the rankings of his restaurant for different customers, thereby obtaining those customers who can not only accept the distance between them and *Tom's* restaurant, but also give higher ranking according to their preference for multi-dimension properties.

Moreover, by reviewing the literature and commercial platforms serving similar purpose, we find that few efforts are made on visualizing the query results, which motivates us to develop a visualization-aided system to benefit sellers by identifying potential target customers.

Therefore, we propose an interactive customer ranking query system named CusFinder. It distinguishes from the literature in three-fold. First, it supports Reverse Spatial Object k -Ranks Query, a novel query which explores optimal k customers satisfying spatial constraints from the perspective of sellers. Second, CusFinder can answer this query efficiently owing to a hybrid index structure and query processing framework. Finally, it is designed to show the result on map dynamically via an interactive visualization procedure. All these functions are presented based on user-friendly interactions upon multiple visualization views, which heuristically guide users to interact with different visualization results [2].

2 Data Model and Queries

Data Model. We maintain two kinds of data in our system. One is object data, and each object $p = \langle \rho, \phi \rangle$ is a pair of position ρ and a set of associated terms $\phi = \langle t_1, \dots, t_d \rangle$ which describes the general attributes. The other one is user preference data represented by a triple $u = \langle \rho, \psi, \sigma \rangle$, where $u.\rho$ is the location where u is posted, $u.\psi = \langle w_1, \dots, w_d \rangle$ is the preference weight (normalized in $[0,1]$ and $\sum_{i=1}^d w_i = 1$), and $u.\sigma$ is the maximum distance the user can tolerate between him and any eligible object.

Query Model. We define an object's rank for a given user, namely $rank(u, q)$, by the number of objects with a smaller score than q for u , which is same with the definition in [3]. In this paper, the geographical position of objects and users has been considered by calculating Euclidean distance.

Definition 2.1 Reverse Spatial Object k -Ranks (RSO- k R) Query. Given an object set $D = \{p_1, \dots, p_n\}$, a user set $U = \{u_1, \dots, u_m\}$, an integer k , a query point q , the RSO- k R will return a set S of k users such that

- (1) $\forall u_i \in S, dist(q, u_i) \leq u_i.\sigma$;
- (2) $\forall u_i \in S, \forall u_j \in U - S$, if $dist(q, u_j) \leq u_j.\sigma$, then $rank(u_i, q) \leq rank(u_j, q)$.

3 System Architecture

As shown in Fig. 1, CusFinder includes a front-end of visual interface and a back-end of query processing. At the front-end, users could issue queries and view the selected results based on multiple coordinated views. Each new user interaction over the interface will be sent to the back-end, and a search engine is employed to answer the query on top of an indexing module. After the query is processed, results containing target customers will be returned and displayed at the front-end by map view and multidimensional view.

Indexing Module. Different index structures are applied to organize the data. A novel index named *Layer Grid Index* composed by a layer dominant graph I_D [5] and a grid index table I_T is used for seller data, while a histogram is used for customer data.

Search Engine. To conduct filtering efficiently, we propose a progressive strategy, which scans for candidate points layer by layer. Besides, when we are seeking for superior customers, it is a wiser method to roughly estimate ranks with lower and upper bounds of the buckets.

Please refer to our recent research work [4] for more details about indexing design and query algorithm.

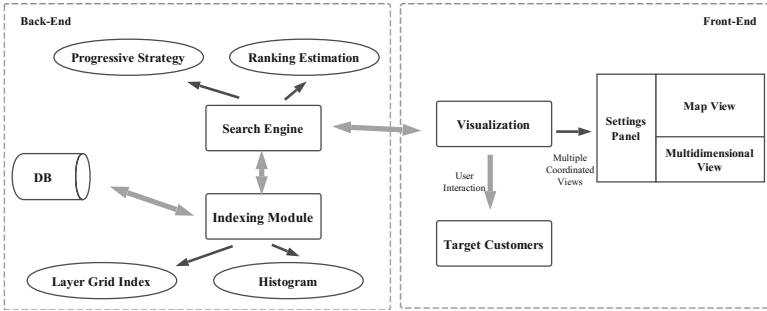


Fig. 1. System architecture

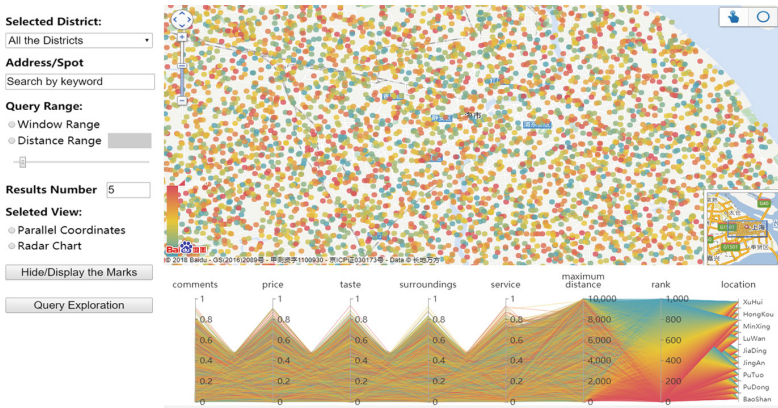


Fig. 2. An overview of the customers

4 Demonstration and Scenarios

We will demonstrate CusFinder based on real-world datasets and scenarios. We crawled restaurant information in Shanghai from Dianping¹, which is one of the biggest lifestyle and group buying websites in China. Each restaurant has descriptive attributes (such as price, taste, surrounding) and location attribute.

Specifically, our demonstration is based on the scenario that *Tom* is trying to find customers who favour his restaurant more than other people, which we have described in *Example 1.1*.

¹ <http://www.dianping.com>.

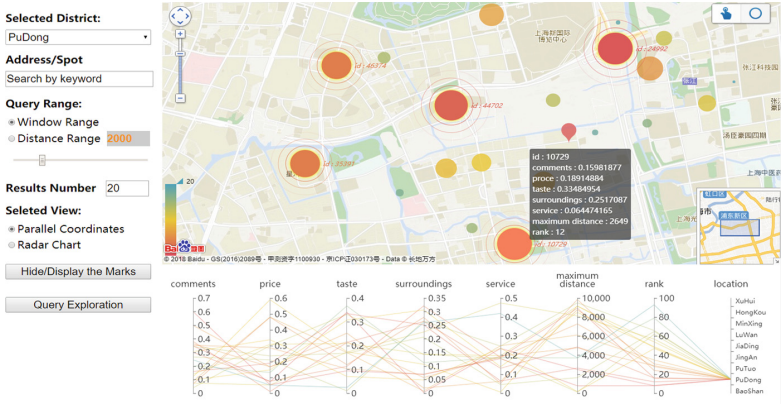


Fig. 3. An example of interactive exploration

Exploration Query 1. Interactive Exploration upon Dynamic Windows. Our first exploration gives users an overview of customers, and helps them understand the distribution and requirements of customers, so that users could retrieve a certain number of customers according to their actual needs.

As shown in Fig. 2, when Tom enters the main page of our system, he notices that there is an enormous quantity of colourful points displayed on the map, which reflect the distribution of customers, and the colour implies the different requirements for distance. Considering that it is overwhelming and meaningless to display all customer data in a limited screen with coarser granularity [1], only those customers from current map window and in the RSO-*k*R Query results will be displayed. All interactions with the map such as zooming in/out and panning will triage a new RSO-*k*R Query within current window range, and all the result views change simultaneously to keep consistency. As shown in Fig. 3, current exploration is based on a restaurant marked with pin in PuDong district, aiming to retrieve twenty customers with optimum rankings within current window range. Finally, 20 results with top-20 rankings have been marked with circles in different sizes and colours.

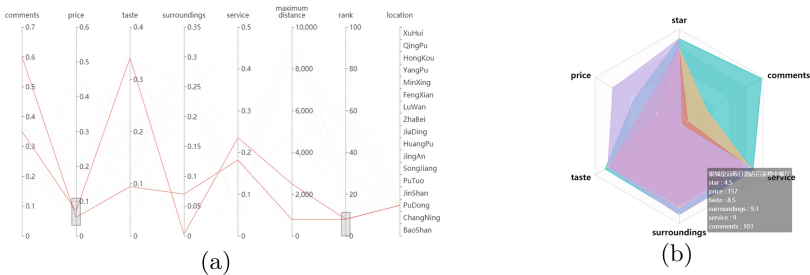


Fig. 4. Detailed comparison: (a) in parallel coordinates; (b) in radar chart.

Exploration Query 2. Detailed Comparison of Selected Customers and Restaurants. After users obtain the filtering results, they want to compare the preference of customers, or discover strengths and weaknesses of their restaurants and competitors. Our second exploration query can help them achieve this goal.

On the basis of interactive exploration, *Tom* intends to know not only the detailed comparison of these selected customers, but also the differences between his restaurants and rivals. With visual assistance of parallel coordinates, he could drag boxes on those coordinate axes indicating properties that he cares about more, thereby observing the corresponding values of these selected customers from each dimension. As shown in Fig. 4(a), although his restaurant is given same ranks by two customers, there are notable differences between the preference. Consequently, he could choose the better one according to his requirements. Besides, *Tom* is able to compare his restaurant with his rivals' restaurants in the radar chart. As shown in Fig. 4(b), he clearly notices that the green coloured restaurant has more favourable comments and lower price compared with others.

Acknowledgement. This research is supported by NSFC 61373015, 61728204, and the Project of State Grid Corporation of China (Storage and Processing of Distributed Parallel Database Based on Big Data Technology).

References

1. Guo, T., Feng, K., Cong, G., Bao, Z.: Efficient selection of geospatial data on maps for interactive and visualized exploration. In: SIGMOD, pp. 567–582. ACM (2018)
2. Wang, S., Li, M., Zhang, Y., Bao, Z., Tedjopurnomo, D.A., Qin, X.: Trip planning by an integrated search paradigm. In: SIGMOD, pp. 1673–1676. ACM (2018)
3. Zhang, Z., Jin, C., Kang, Q.: Reverse k-ranks query. PVLDB **7**(10), 785–796 (2014)
4. Zhou, Y., Qin, X., Xie, X., Guo, C.: A reverse k-ranks query based on clusters. J. Chin. Comput. Syst. (2018, in press)
5. Zou, L., Chen, L.: Dominant graph: an efficient indexing structure to answer top-k queries. In: ICDE, pp. 536–545. IEEE Computer Society (2008)