Efficient General Dominant Relationship Analysis based on Partial Order Models

Zhenglu Yang Lin Li Masaru Kitsuregawa

Department of Information and Communication Engineering

University of Tokyo

Abstract

Due to the importance of skyline query in many applications, it has been attracted much attention recently. Given a k-dimensional dataset *D*, a point *p* is said to dominate another point *q* if *p* is better than *q* in at least one dimension and equal to or better than *q* in the remaining dimensions. However, in some real applications, users are more interested in the detail of the general dominant relationship in a business model, i.e., a point *p* dominates how many other points and is dominated by how many others, which is called general dominant relationship (GDR). In this paper, we explore how to efficiently analyze the GDR query. We show that the framework proposed before can not efficiently solve this problem. We find the interrelated connection between the partial order and the dominant relationship. Based on this discovery, we propose efficient algorithms to answer the GDR queries by querying the partial order representation of spatial datasets. Extensive experiments illustrate the effectiveness and efficiency of our methods.

1. Introduction

Recently, the skyline query has attracted considerable attention because it is the basis of many applications, e.g., multi-criteria decision making, user-preference queries and microeconomic analysis. Fig. 1 shows one classic example of skyline query that customers are always interested in those ``best'' hotels that are better than others at least at one of the two criteria, the distance and the price, with smaller values. The skyline of the example dataset in Fig. 1 consists of *a* and *c*.

Efficient skyline querying methodologies have been studied extensively [2, 4, 5]. All the papers concerned only the pure dominant relationship among a dataset, i.e., a point *p* is whether dominated by others or not, and got those non-dominated ones as results.

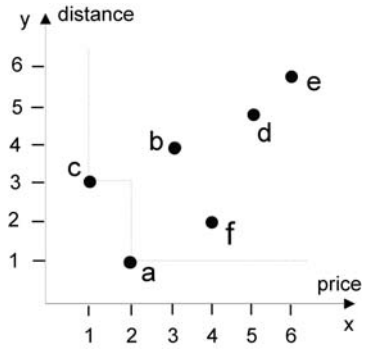

Fig. 1: Example of skyline query to find "best" hotels

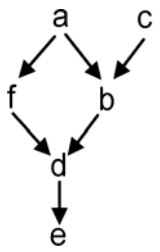

Fig. 2: DAG representation in 2-d space

However, in some real applications, users are more interested in the detail of the general dominant relationship in a business model, i.e., a point *p* dominates how many other points and is dominated by how many others. In Fig.1, although the hotel *b* is not a skyline, its manager also wants to know how many hotels *b* dominates (i.e. 2), how many hotels dominates *b* (i.e. 2) and whom they are, from where the manager can know the business position of *b* in the local area. Obviously, this kind of dominant relationship analysis requires more information explored than the original one of skyline query [2].

To illustrate our proposed core idea, here we show a simple example. Fig. 2 is the corresponding partial order (encoded as DAG format) of Fig. 1. We can know that item *b* dominates items *d* and *e* and is dominated by items *a* and *c*, by checking the *out-link* and *in-link* of *b*, respectively. Moreover, no *in-link* nodes such as *a* and *c* are candidate items (*skyline*). From this example, we know that the

general dominant relationships of a dataset can be represented into their corresponding partial order representation (i.e., DAGs). In this paper we explore how to efficiently find such succinct representative partial orders.

2. Proposed Strategies

2.1 *ParCube* Construction

We find that the dominant relationship between two items in a k-dimensional dataset (i.e., a dominants b) can be represented as a frequent sequence pattern in the corresponding k-customer sequence dataset. Because the small-large pair (dominant) relationship in the spatial dataset is equivalent to the early-late pair (dominant) relationship in the converted sequence dataset. For example, the example spatial dataset in Fig. 1 can be converted to a sequence dataset, by considering each dimension as a customer in the sequence dataset. The result dataset is D1: <cabfde>, D2: <afcbde>. By employing sequential pattern mining algorithms, we can get the frequent patterns, i.e., (afde), which indicates that a dominates f and dominates d and dominates e. The order in the pattern describes the dominance relationship between items. After we get the frequent sequence pattern, we merge them into partial orders, which is the concise model of the dominance relationship representation.

Next we explain the detail how to construct the partial order data cube (*ParCube*) with a spatial dataset input. In this paper, we propose to apply strategies from another research context, sequential pattern mining [1], to get the partial order representation from a spatial dataset. The whole work flow is shown in Fig.3. We propose a simple method of converting the spatial dataset to the corresponding sequence dataset in the first process and then, apply existing strategies such as that used in [3] with modification in the second and third processes to generate DAGs from the transformed sequence dataset. Note that we mainly illustrate how to compute the cube for a dominating set since computation of a dominated set can be done in a similar fashion.

Among the three processes of partial orders finding as illustrated in Fig. 3, the second one,

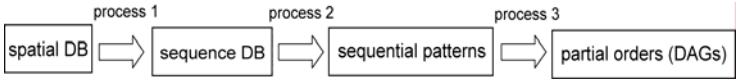

Fig. 3: Work flow of *ParCube* constructing

sequential pattern mining, is the slowest process although the state-of-the-art algorithm is used. To improve the efficiency of the whole system, we aim to develop an optimized algorithm to fasten the mining process by considering the special property of the converted sequence datasets.

We find that the converted sequence dataset has one important characteristic: for each customer sequence (dimension), one item appears and only appears once. In other words, there is no two same items existing in the same customer sequence (dimension). This is very different from general sequence, i.e., Web log sequence, customer shopping history or DNA sequence. Based on this discovery, we have largely improved the performance of the mining process.

2.2 Efficient *ParCube* Querying

We introduce the strategy to efficiently answer the general dominant relationship query. The semantic meaning kept in the *ParCube* data cube is the key used to extract the general dominant relationship.

An important observation in this case is that, if P_{query} is in D, all the general dominant relationship related to P_{query} can be easily discovered by traversing the DAG in a specific subspace. The example of querying is already described in Section 1.

We have done extensive experiments to illustrate the effectiveness and efficiency of our methods.

Reference

[1] R. Agrawal and R. Srikant. Mining Sequential Patterns. In *ICDE*, pp. 3-14, 1995.

[2] S. Borzsonyi, D. Kossmann, and K. Stocker. The skyline operator. In *ICDE*, pp. 421-430, 2001.

[3] G. Casas-Garriga. Summarizing sequential data with closed partial orders. In *SDM*, pp. 380-391, 2005.

[4] D. Kossmann, F. Ramsak, and S. Rost. Shooting stars in the sky: An online algorithm for skyline queries. In *VLDB*, pp. 275-286, 2005.

[5] D. Papadias, Y. Tao, G. Fu, and B. Seeger. An optimal and progressive algorithm for skyline queries. In *SIGMOD*, pp. 467-478, 2003.