International Journal of Engineering Research and Technology. ISSN 0974-3154 Volume 10, Number 1 (2017)

Implementing and Applying an Indexing Structure Dominant Graph for Dominant Relationship Analysis

Mr. S S Dhawale
Research Scholar, Department of Computer Science and Engineering,
Dr. D. Y. Patil College of Engineering and Technology,
Kolhapur, Maharashtra, India.

Dr. V R Ghorpade
Guide, Dr. D. Y. Patil College of Engineering and Technology,
Kolhapur, Maharashtra, India.

Abstract

Dominant relationship concept is used for answering preference queries. The concept of dominant relationship has been extended here for the purpose of doing business analysis. New kind of analysis presented here which is called as dominant relationship analysis. This type of analysis can help product manufacturing companies to design new products, compare company's products with products of competitor company and to find interesting subspaces of the product which can be helpful for promoting the products. Algorithms which use Dominant Graph(DG) to answer dominant relationship queries are presented here.

Keywords: dominant relationship analysis, dominant graph, dominant relationship queries.

Introduction

Dominant relationship concept has recently attracted much interest to answer preference queries. Here, the concept is extended for the purpose of doing business analysis using DG. Definition (Dominate). If there are two records p and q in a multidimensional space, we can say that p dominates q if following two conditions are satisfied 1) in any dimension I i , the value of p is larger than or equal to q , i.e. p.Ii >= q.Ii 2) there must exist at least one dimension I j , such that p.Ij > q.Ij .

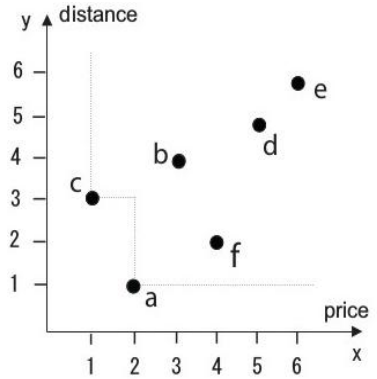

Figure 1: Example of skyline query

Preference queries are used to find those points, which are not dominated by others, in a n-dimensional dataset. Fig. 1 shows example of preference query where customers are always interested in hotels which are better in minimum in one of two dimensions, the distance and the price as compared to other hotels. In example shown in figure 1 a and c are products which are not dominated by any other products. Like dominance concept is useful for customer to find products which they like, it can also be useful for manufacturing companies to determine that how popular are their products among customers in comparison to their competitor company's product. Suppose that you are a manager of hotel company. You want to know the position of hotel b in market with respect to preference like hotel's price and hotel's distance from the beach by finding number of hotels that are better or worst than hotel b. From the fig 1 , it can be easily identified that hotel b is better than 2 other hotels but worse than another 2 hotels with respect to preference given. Here, five types of queries are presented for dominant relationship analysis. Queries are as follows (i) Linear Optimization Queries (LOQs), (ii) Subspace Analysis Queries (SAQs), (iii) Comparative Dominant Queries (CDQs), (iv) Skyline Product Query (SPQ) and (v) Skyline Subspace Query. This dominant relationship analysis can help product manufacturing companies to design new products, to compare company's products with competitor company's products and to find subspaces of a product which can be helpful for promoting the products. Algorithms are designed for answering these queries using DG. The paper is organized as follows: In section 2 the preliminaries of this paper are presented. Section 3 discusses related work. Section 4 defines problem statement. In section 5 algorithm for computation of DG is presented. Section 6 presents algorithms for dominant relationship queries. Conclusion is presented in section 7.

Preliminaries

This section defines some terms used in this paper and also makes some assumptions. After that four types of dominant relationship queries are presented which are used for analysis .

Assumptions

There is an assumption made here that there are two manufacturing companies A and B, they produce a set of

products PA = {A 1 , . . . , A s } and PB = {B 1 , . . . , B t }, respectively. There is also a set of customer preferences C = {C 1 , . . . , C n }. Products or customer preferences can be represented by a point in an N -dimensional space, D, with dimensions D 1 , . . . , D N are the attributes of the products and customer preferences. It is also assumed here that product s or customer preferences are having minimum attributes.

Definitions

Definition 1 dominating(p, C, D') If an object p, a set of objects C and a set of dimensions D' ⊆ D are given, then dominating(p, C, D') is defined as the set of objects from C which are dominated by p in the subspace D' of D. analysis)
Definition 2 dominated(C, p, D') If an object p, a set of objects C and a set of dimensions D' ⊆ D are given, then dominated(C, p, D') is defined as the set of objects from C which dominate p in the subspace D ' of D.

Dominant Relationship Queries

***Linear Optimization Query (LOQ)* :** Given a plane L, and a set of objects, C, in an N -dimensional space of D, we define LOQ(L, C, D) as the aggregate max(|dominating(p, C, D)|), such that p is any point in the plane L. This query can product manufacturing companies in designing new products that satisfy most of the customer preferences while remaining profitable.

Subspace Analysis Query (SAQ) : Given a set of points C and a point p in the N -dimensional space of D', SAQ finds: 1. |dominating(p, C, D')| and 2. |dominated(C, p, D')| where D' ⊆ D. Products have different attributes like a smart phone have attributes like processor speed, ram size, internal memory, camera, battery, price. When all attributes of a product are considered then it may not satisfy many customer preferences but if we consider few attributes of the product then same product may satisfy more number of customer preferences. By using subspace analysis query it will be easier for product manufacturing company to identify subspaces of a product where product dominates more customer preferences in the selected subspace.

Comparative Dominant Query (CDQ): Since there is tremendous competition among product manufacturing companies these days, these companies may be interested in comparing their own products with products of competitor company, doing so that they will be able to assess the position of their products in market. CDQ can help companies to achieve this goal. CDQ helps product manufacturing companies to identify customer preferences which are dominated by products of both a company and it's competitor company. CDQ also helps in identifying customer preferences that are dominated by a company's products and not by competitor company's products. gdominating(A, C, D):- If set of objects A and C in an N -dimensional space of D are given, then gdominating(A, C, D) is defined as the set of objects in C which are dominated by some object from A. CDQ- (A, B, C, D):- If three sets of objects A, B, C in the N -dimensional space of D are given, then CDQ- (A, B, C, D) is defined as: |gdominating(A, C, D) − gdominating(B, C, D)|. Similarly if three sets of objects A,B,C in the N -dimensional space of D are given, then CDQ∩(A, B, C, D) is defined as as: | gdominating(A, C, D)∩ gdominating(B, C, D)|.

Skyline Product Query (SPQ): It is useful for product manufacturing companies in designing new products which are not dominated by any existing product in market. Given a set Te of existing products in market, SPQ creates a set of best possible products from source tables T1, T2,...Tn of sub-products such that the newly created products are not dominated by any existing products.

Skyline Subspace Query (SSQ):

SSQ can help product manufacturing companies to find subspaces of a product where the given product is not dominated by any customer preferences or any existing product in market. Given a set C of customer preferences and a point p in the N -dimensional space D. Subspace Sky-line Query can be defined as determining all possible subspaces where point p is not dominated by any customer preference.

Related Work

Many techniques have been proposed for dominant relationship analysis. Here relevant work is described. Work is divided into three categories: data mining methods for market analysis, dominant graph method, skyline query method.

Data Mining Techniques for Business Analysis

This work is motivated from the work in [1] where authors say that interestingness of knowledge being discovered depends on it's usefulness to the organization. Examples in [1] discusses utility oriented mining which includes sensitivity analysis, market segmentation. Work in [2] includes profit oriented association rule discovery for product assortment. [3] discusses customer oriented catalog segmentation problem. [4] discusses profit oriented pattern discovery . [5] discusses maximum-profit item selection with cross-selling consideration. [6] discusses data mining as sensitivity analysis. In [7] dominant relationship analysis model is presented.

Dominant Graph

The DG index was proposed in [8]. In dominant graph records are linked according to dominant relationship. Based on off-line built the DG, Traveler algorithm has proposed to answer a top-k query.

Skyline Queries

Skyline queries [7] and Top-k queries [6] are very popular to answer preference queries, and there are many indexing structures are available to find skyline and subspace skyline efficiently on static data set .

Dominant Graph

This section defines dominant graph and presents algorithm for it.

Defining DG

Set S of records in a multi-dimension space is given, S does have k nonempty maximal layers L i , i=1 . . . n. The records r in i th layer and records r' in (i+1)th layer form a bipartite graph gi , i =1 . . . (m-1). Directed edge will be there from r to r ' in gi if and only if record r dominates r' . This directed edge represents "parent-children relationship" . All bipartite graphs gi are joined to obtain DG. The maximal layer L i is called i th layer of DG. There are two types of dominant graphs dominated dominant graph and dominating dominant graph. Dominated dominant graph shows dominated relationship among records in graph. Dominating dominant graph shows dominating relationship among records in graph.

Algorithm for Dominant Graph

Input: i) Dominant graph of the database. ii) r = record to be inserted.

1) If r is not dominated by any record in first layer of DG then

2) set n=0

3) else

4) All record Pi at first layer of DG that dominates r are collected to form the set P.

5) Do DFS search from each Pi to find the longest path L, and each record in dominates r.
6) Set n = |L|.
7) Insert r into the (n+1)th layer of DG.
8) If r dominates some records Ci in the (n+1)th layer of DG then
9) All the descendents records of Ci (including Ci) are collected to form the set S.
10) For each record O in S do
11) O is degraded into it's next layer.
12) Build parrent children relationship between O and records in current next layer.
13) if O has some other parrent A that is not in set S then
14) Delete the directed edge from A to O.
15) Build parrent children relationship between records in nth layer and r.
16) Build parrent children relationship between records in r and n+2 th layer.
17) Report the updated DG.

TID	X	Y
000	1	563
001	193	808
002	585	479
003	350	895
004	822	809
005	174	858
006	710	513
007	303	14
008	91	364
009	147	165
010	100	800
011	351	810

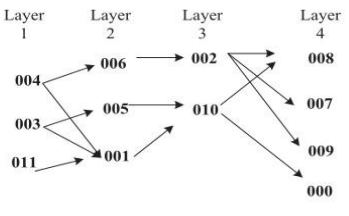

Figure 2: Dominant Graph for given dataset

Figure 2 shows a sample dataset consisting of 11 records and DG built for it using algorithm described above.

Problem Statement

A Dominant Relationship Analysis Tool is presented here for product manufacturing companies. This analysis tool can be used for helping product manufacturing companies to design new products and to extracts important information from large dataset. Extracted information will be very helpful to manufacturers to do the business analysis, to determine the position of their products in the market more effectively, and to identify interesting subspaces of a product which can be used for promoting the products .

Answering Dominant Relationship Queries

In this section, algorithms are proposed to answer dominant relationship queries using DG.

Linear Optimization Query

Given a plane L, a set of objects C in space D, we wish to find some cells which are intersecting plane L and dominate the most points in C.

Algorithm

1)Input: A plane L, set of objects C .
2)Output: Points on plane L which dominates maximum number of objects from C.
3) Method:
- i) Start from the point 1,1,...1 in N-dimensional space.
- ii) At any stage if the cell is at the bottom left of plane L, iterate continually to it's children till we find the cell on the plane L.
- iii) If the cell is on the plane L add it to the result cell set.
- iv)Obtain the cells from result cell set which dominate maximum number of objects from set C using LOQ Max Dominating function.

4) Function LOQ Max Dominating (result cell set, DG for set C)
- i) Find the position of the cell in DG for set C.
- ii) Determine number of objects in set C are dominated by the cell.
- iii) Repeat step 1 and 2 for each cell in result cell set.
- iv) Return the cells which dominate more objects in set C than any other cell in result cell set.

Subspace Analysis Query

If we are given a set of points C and a point p in the N-dimension of D, the most basic SAQ is to find for each subspace D' , how many points dominating or dominated by p.

Algorithm

1) Input: i) C = Set of objects in N-dimensional space D.
- ii) D' = A subspace of dimension D.
- Iii) A point P.

2) Output: i) Number of points in C dominated by P.
- ii) Number of points in C dominating P.

3) Method:
- i) Build dominating dominant graph DG1 for set of objects in C.

ii) Find the position of product p in DG1 and determine how many objets form C are dominated by point p.
iii) Build dominated dominant graph DG2 for set of objects in C.
iv) Find the position of product p in DG2 and determine how many objets form C are dominating point p.

Comparative Dominant Query

Comparative dominant queries are those queries which are used to compare the set of dominated objects between competitive products. There are two kinds of CDQs:CDQ−(A,B,C,D), which retrieves |gdominating(A,C,D)−gdominating (B, C, D)|; and CDQ∩ (A,B,C,D), which retrieves |gdominating(A,C,D) ∩ gdominating(B,C,D)|.

Algorithm

1) Input: i) Three sets of objects A, B, C.
ii) Dominant graph for C.
2) Output: i) Set of objects in C which are dominated by some objects in A and not by any object in B.
ii) Set of objects in C which are dominated by some objects in A and some objects in B.
3) Method:
i) Take an object x from set A.
ii) Find the position of that object x in dominant graph.
iii) Determine objects in C which are dominated by the object x.
I v) Repeat steps 1, 2, 3 for each object in set A.
v) Let S1 be the set of objects in C which are dominated by objects in A.
vi) Take an object y from set B.
vii) Find the position of the object y in dominant graph.
viii) Determine objects in C which are dominated by the object y.
ix) Repeat steps 6, 7, 8 for each object in set B.
x)Let S2 be the set of objects in C which are dominated by objects in B.
xi) CDQ- (A, B, C, D) = S1 — S2.
xii) CDQ ∩ (A, B, C, D) = S1 ∩ S2.

Skyline Product Query

Given a set Te of existing products in market, a set of best possible products is to be created from source tables T1, T2,...Tn of sub-products such that the newly created products are not dominated by any existing products in Te.

Algorithm

1) Input: i) Dominant graph for existing products.
ii) Source tables of sub products T1, T2, ...Tn.
2) Output: Table of newly created skyline products.
3)Method:
i) Build dominant graph for existing product dataset Te.
ii) Generate new product table Tp by combining subproducts in tables T1', T2'...Tk'.
iii) Find a set Tp' of all products from Tp which dominate all the products in skyline of DG of Te.
vi) Return Tp'.

Skyline Subspace Query

Given a set C of customer preferences and a point p in the N - dimensional space D. Subspace Sky-line Query can be defined as determining all possible subspaces where point p is not dominated by any customer preference.

Algorithm

1) Input: i) C: An N dimensions set of customer preference
ii) A point p.
2) Output: All the subspaces of product P where p is in skyline.
3)Method:
i) Create DG for each dimension in D for dataset C.
ii) Read point p.
ii) Obtain the sets S1,S2....Sn such that each point in Si dominates p in the dimension Di
iv) if S1 ∩ S2....∩ Sk == NULL then Point p is a skyline point in subspace {D1,D2,...Dk}.
vi) For all combinations of dimensions in D check whether p is in skyline or not.

Experimental Setup and Result Analysis

To evaluate the efficiency and effectiveness of dominant graph and dominant relationship queries extensive experiments were conducted. All algorithms implemented using Net-beans 8.2 and Oracle 10g database, and conducted the experiments on a PC with Intel core i3 processor, 3GB main memory and 80 G hard disk,running on Ubuntu 14.04 Edition.

Efficiency of DG computation

To evaluate the efficiency of algorithm for computing Dominant Graph,the number of records fixed to 1000 and varied the number of attributes from four to seven. Figure 3 show the run time of the algorithm.

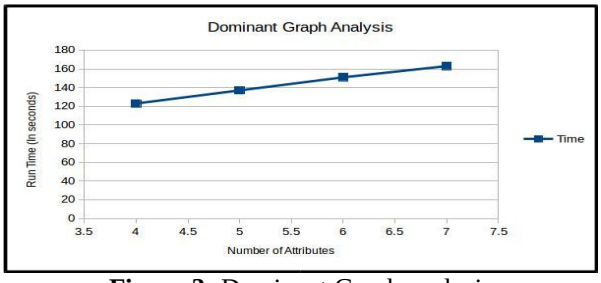

Figure 3: Dominant Graph analysis

Next, we look at the run time of the algorithms as the number of attributes fixed to 7 and the number of records varied from two hundred to one thousand. Figure 4 shows the run time of the algorithm.

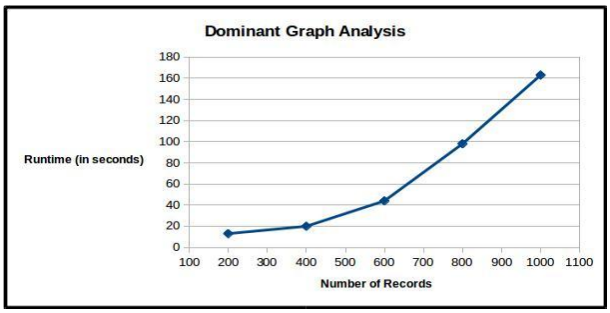

Figure 4: Dominant Graph analysis

Query Answering Performance

Below figures show performance of five types of queries. Runtime of all five queries is measured against number of attributes. Here number of attributes are fixed to seven and number of records varied from two hundred to one thousand.

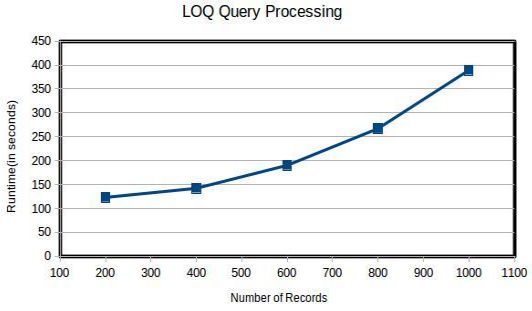

Figure 5: LOQ analysis

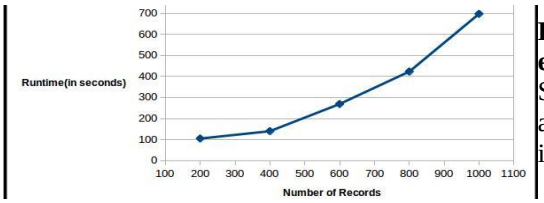

Figure 6: SAQ analysis

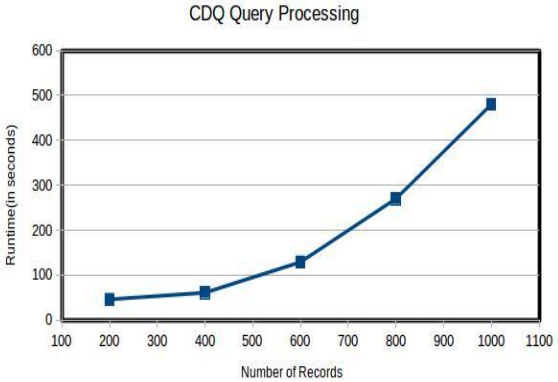

Figure 7: CDQ analysis

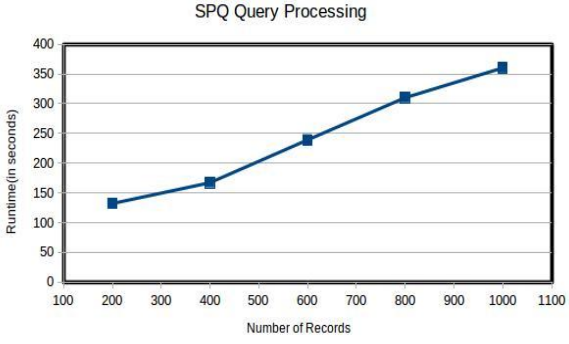

Figure 8: SPQ analysis

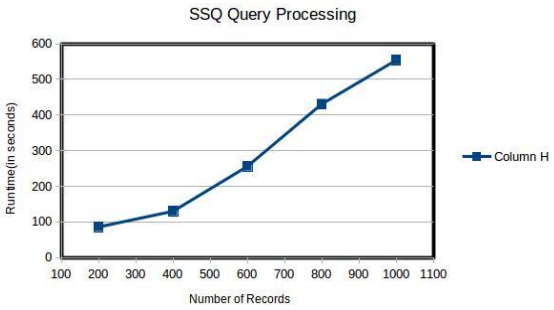

Figure 9: SSQ analysis

Conclusion

In this paper, five types of queries are presented for Dominant Relationship Analysis. To support these queries, an indexing structure called dominant graph is used. In the future work, it would be interesting, how to further improve the efficiency while solving the dominant relationship queries.

References

[1] Kleinberg J, Papadimitriou C, Raghavan P "A microeconomic view of data mining", Data Min Knowl Discov 2(4):311–322, 1998.

[2] Brijs T, Swinnen G, Vanhoof K, Wets G , "Using association rules for product assortment decisions: a case study", In: Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining, pp 254–260,1999.

[3] Ester M, Ge R, Jin W, Hu Z, "A microeconomic data mining problem: customer-oriented catalog segmentation", In: Proceedings of the 10th ACM SIGKDD international conference on knowledge discovery and data mining, pp 557–562, 2004.

[4] Wang K, Zhou S, Han J, "Profit mining: from patterns to actions", In: Proceedings of the 8th international conference on extending database technology, pp 70–87, 2002.

[5] Wong R, Fu A, Wang K, "MPIS: maximal-profit item selection with cross-selling considerations", In: Proceedings of the third IEEE international conference on data mining, pp 371–378, 2003.

[6] Yao J "Sensitivity analysis for data Mining", In: Proceedings of the 22nd international conference of the North American fuzzy information processing society, pp 272–277, 2003.

[7] C. Li, B.C. Ooi, A.K.H. Tung and S. Wang. DADA: A data cube for dominant relationship analysis. In SIGMOD'06: Proceedings of the ACM SIGMOD International Conference on Management of Data, pp. 659-670, 2006.

[8] Lei Zou, Lei Chen, "Pareto-Based Dominant Graph: An Efficient Indexing Structure to Answer Top-K Queries", IEEE transactions on Knowledge and Data Engineering, Vol.23,5, 2011.