Commercial Application of Most Diverse Result set using Preference Queries

Erik Ringdahl, Carl Eriksson and Johannes Moberg

School of Computer Science and Communication (CSC), Royal Institute of Technology KTH
{erikrin, jmobe, carerik}@kth.se

Abstract. This paper examines a proposed method of returning the most diverse set of products based on user preferences e.g. reaching the largest possible audience with an advertising campaign. We evaluated the algorithms proposed by the authors and their usage in a commercial setting and found some issues which we attempted to resolve. We present a novel approach to adapting the algorithm to create more intuitive results that are more suitable for a commercial application.

1 Introduction

Preference queries is the notion of selecting products based on customers personal preferences. The top-*k* set of these queries represents the *k* number of products that is the best match for that customer as shown in Table 1. A reverse top-*k* set is, as the name suggests, a query for preferences based on products as shown in Table 2. Diversifying the resulting set of products would then identify the *r* number of products that would attract the customers with least similar preferences, thus increasing the market impact and attracting more new customers. In [1] the authors suggests several algorithms for solving this problem and focuses on performance rather than accuracy making the resulting set approximately most diverse. However, they also present an optimal solution which is slower but does not return approximate results. We implemented the optimal solution and focused on the feasibility of this approach in a commercial setting and as a proof of concept.

Table 1: Top-*k* set

User	Preferences		Top-*k*
Bob	0.1	0.9	p1
Tom	0.2	0.8	p1
Jack	0.8	0.2	p2

Table 2: Reverse top-*k* set

Product	Attributes		Reverse top-*k*
p1	1	7	Bob, Tom
p2	4	4	
p3	6	2	Jack

2 Implementation

Our algorithm works on a database of user and product data and the key factor of the result is accuracy. With this in mind the implementation of the algorithm is not optimized for efficiency but rather to deliver optimal results. Imagine a scenario where you are planning for a national or large advertising campaign. The time it takes to run the algorithm on a large set of users with a high *k*-value[1] is not even comparable to the time it takes to prepare an advertising campaign or even get to your office. You would rather have the calculation take a bit longer and have correct results, rather than running the risk of a failed campaign. The proposed algorithm in [1] takes the results of a reverse top-*k* query and approximates a centroid preference for the resulting customers. The centroids of all products is then

[1] In [1], test results show (Figure 9 and 10, page 11) that a top-30 query on 500k products takes approximately 100 sec with close to 80 % accuracy.

evaluated to create the most diverse set. We take a slightly different approach as shown later in this chapter.

2.1 Preferences and Product Weighting

The vectors of percentage values that defines the user preferences can be found or calculated in several different ways. Since preferences are context dependent and highly individual, there is no general method to describe this process. Furthermore, as human cognition is very unpredictable, a user may describe herself as price-conscious but in reality, always buy the most expensive items. So surveys where users are asked to put a number on their preferences are probably not very useful and time consuming. A straightforward approach is relative ranking [3] where we simply take the ratio of an attribute of a product relative to the rest of the products. Obviously, there must be a context for the products at hand since we can't compare the price of a house to the price of a cell phone. However, with a large database of information on specific users (which we can assume many advertising companies keep) we might be able to combine the user preferences which were calculated in their respective context. If a user is always very conscious about the price of a purchase, no matter what the product, we might be able to draw some conclusions about their future purchases. But ranking relying on purchase history brings up several more issues. How can we possibly know that the user made a conscious decision about the price? It might just be the case that all the other attributes of the product where the most important ones and the user had been satisfied with a higher price.

Search history, product ranking and active participation in discussions, reviews and other consumer related activities might be a more useful data source. The keyword we are looking for here is active. If a user made a conscious, active decision about a product related to a certain attribute we can confidently build a preference vector with those values. The objective here is not to try to reinvent the field of marketing but rather stress the importance of the preference implementation and weighting of attributes since commercial use of this algorithm is very dependent on accurate preferences and weights. Expert opinions, ideal values and all the options mentioned above must also be considered when products are evaluated and ranked based on various attributes. Collecting evidence and data is crucial.

Weighting. We implemented a system for weighting that creates uniform vectors for product attributes and user preferences i.e. all vectors consists of percentage values that adds up to 1. By doing this, we can utilize all algorithms for selection in both directions i.e. the top-k and reverse top-k are the same algorithm and are rather defined by their input. Now, we can plot products and preferences in the same graph, making the results more intuitive and the different stages of the algorithm can be easily observed which would be helpful in a commercial application of the algorithm. This also helps in building indexes in the database for products and preferences.

2.2 Top-k Results

As stated earlier, we took a slightly different approach to the implementation in [1] since we favored accuracy over speed. Since we are not dependent on approximation of centroid preferences for a user group as a result of reverse top-k queries, we can work directly with the top-k results of the users.
This is done by generating a score for each product-user pair. Thereafter the top-k products with the highest scores are selected for each user. In the case where a large portion of the preferences are matched with the same product we are faced with a new issue. Simply choosing the most diverse set from the top-k result would give products with large sets of users equal importance to a product preferred by only one user. In [1] this problem is ignored since they use centroid preferences for selecting the most diverse products. In a commercial application of the method this is not very desirable, as catering to one single customer with one of the products in an advertising campaign

defeats the purpose of the campaign and personalized marketing is more suitable in that case. We propose a novel solution to this problem by selecting the top-*k* products with the highest total score i.e. the products with attributes that satisfies the needs of the largest target audience.

2.3 Diversity

To get the *r* most diverse products from the top-*k* results, the products with the largest total distance between them are picked. The distance is calculated between each vector in the top-*k* set. We then start of with the most distant pair and then iteratively add the next vector that maximizes the distance until *r* products are chosen. The distance between two vectors can be calculated in different ways. For our algorithm we used two different measurements, cosine similarity which is not dependent on magnitude and the euclidean distance; both described in detail in the next section.
As long as the set of top-*k* products are reasonably small finding the set of *r* most diverse products is not that computationally heavy it does however not scale that well with the growth of the top-*k* set and the number of vertices *r* to maximize distance in between but it should still be within polynomial time complexity.

Cosine similarity. The similarity score is has a range between -1 to 1. Where -1 means that the vectors u and v are the opposite to each other and 1 that they are exactly the same.

$$similarity(u,v) = cos(\theta) = \frac{u \cdot v}{\|u\|\|v\|} \tag{1}$$

Euclidean distance. The euclidian distance is the distance in a straight line between two vectors in euclidian space. It produces more intuitive results than cosine similarity when graphed since magnitude it is based on magnitude i.e. the longest distance to travel between points.

$$distance(p,q) = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2} \tag{2}$$

In our application we used three dimensional vectors as preferences, thus making the Euclidian distance suitable to use and easy to visualize. Though if the algorithm would be used on data with larger vectors it's worth taking the curse of dimensionality[2] into account.

3 Results

Since this project's main purpose was to implement an already known algorithm we can draw the conclusion that the result would not differ much from [1]. Instead, we focused on the feasibility of implementing the algorithm and if it could presents any valuable facts. Since we are not doing any approximations, the algorithm is slow but since optimization was not the issue but rather the usability, this had no effect on the results. In fact, since we are returning the optimal solution (on small sets) we can ignore the issues of approximated results and their impact.

The result can be evaluated in two steps, the first, which is selecting the top-*k* products from all customers preferences. This is shown in Figure 1, where the red signs represent a set of products with different attributes. In the next step, shown in Figure 2, we go through the preferences of each customer and create a top-*k* result set excluding those products that are not popular enough. The top-*k* result set is shown as the red signs marked with a green cross in Figure 2. In the last step we calculate which *r* entities within the top-*k* result set that are most diverse from each other. The result of this step is visualized in Figure 3 as the red signs surrounded by a blue circle.

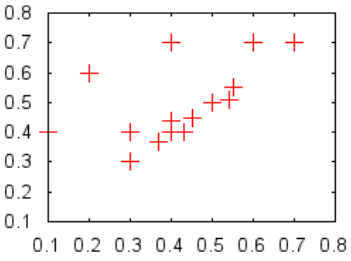

Fig 1: Products

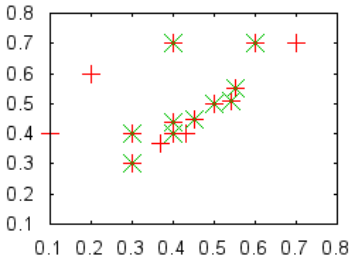

Fig 2: Top-*k* results marked with x

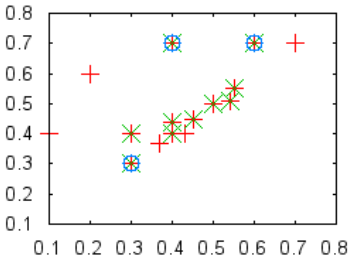

Fig 3: *r*-diverse products in circles

We suggest that using the optimal algorithm is a more viable solution for examining market impact of a product or for selecting a set of products for a campaign. We found that the solution presented in [1] is more suitable for diverse product selection where approximations are adequate and a fast result is more important e.g. an online electronics store where a user specifies a number of attributes they look for in a product and wants a diverse selection of candidates instantly.

4 Future work

In our evaluation, we have been working with small datasets in the attempt to establish if the proposed algorithms and implementation models. We can clearly see that the concept is functional on our synthesised data so a next step would be to try it on commercial data coming from a live marketplace. Here we propose a data set containing search history, user rankings or user reviews where preferences can be built on a set of active choices.

References

1. Gkorgkas, O., Vlachou, A., Doulkeridis, C., & Nørvåg, K.: Finding the Most Diverse Products using Preference Queries. In: Proc. International Conference on Extending Database Technology, Brussels, Belgium (March 2015)
2. Ernest, Bellman R.: Dynamic programming. Princeton University Press (1957)
3. Wang, P.: The interpretation of fuzziness. IEEE Transactions on Systems, Man, and Cybernetics, 26:321-326. (1996)