Determining the new product option that maximizes total cover ratio on users

Keming Li

Computer science and technology Supervisor: Bo Tang

**ABSTRACTION:**

Decision making problem is popular among these days. In rank aware processing, a user will choose the option that ranks tops for himself. Concretely, preferences of users are usually represented as weight vectors. Each attribute of weight vector means how important is that attribute to that user. The score of an option respecting to a user is the dot product between the user’s preference weight vector and the option. Only the options with top-k scores can attract the user. An option covers a user if and only if it can rank top-k for that user. Usually, a company has many products (options), each of which covers some of the users. A user covered by a company means at least one of its products covers this user. The company has to develop new product that satisfies a constrain and make its all products including this new product cover as more users as possible. In this problem we study how to determining which newly added option can maximize the cover ratio of the company. This problem is essential in developing new product, market decision, advertising, etc. We refer this problem as k-Cover Ratio Maximization (). In this paper, we begin from top-k problem’s computational geometric nature using tree to represented option spaces, and then from the relationship among constrain, options and user preference weight vectors to more efficiently solve this problem.

**KEY WORDS**: Cover ratio, Introduce new option, Top-k query

**1. INTRODUCTION**

Take smartphone market as an example, there are different models of smartphones（）. Each model （）has different prices, pixels, battery capacities, cooling capacities and etc. Now a company owns models () and a user data set . Different users prefer in different aspects, for example some of them may prefer smartphone that with large battery capacity but some prefer better quality of screen, so each presents the preference weight vector corresponding to a single user. The score of a model respecting to a user is the dot product . Usually a user will only make a choice from his own view of top-k, so a product ranks top-k for the users is quite important. A product covers a user only when the score of it ranks top among all existing products (). With developing of company and market, company has to develop a new product to cover more user and make more profit. But with the limitation of technology, money and other factors, one can not develop a perfect product to cover all users. It can only develop a product that covers as more user as possible under a constrain. For a company, some of its products have covered some users, so what it wants to do is how to make this new product cover more users that uncovered before.

In real life, k-Cover Ratio Maximization () can solve the problems that how to decide the next generation product for companies. In advertising industry, it can help merchants how to cover specific group such as students, pregnant women and children. Besides it can tell advertiser where to set up new advertising board and if do so it can cover which group of people. Data analysts can use it to discover which group of people is ignored by the market.

Generally speaking, it is not a product covers a user but a group of products covers a user. And for different users, there are different groups of products. Among these groups of products, there are uncertain number of identity products. In our problem, we are aim to find this new product in continuous product space, which means there are infinite candidate products, so it is difficult to tell which product covers the most users.

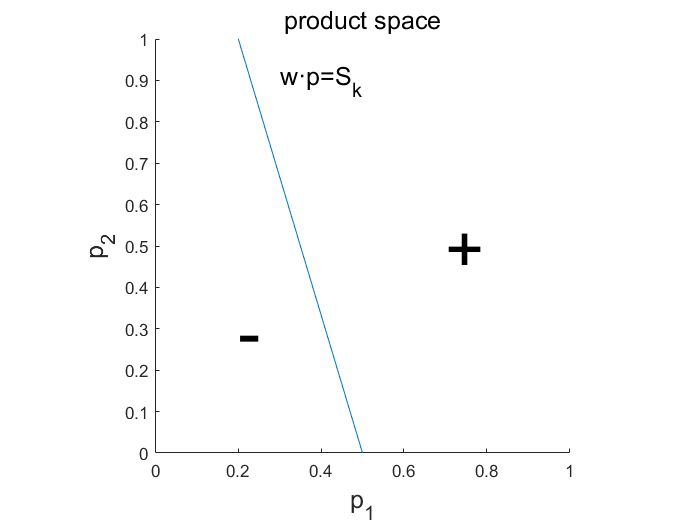
In this report, we will from computational geometric nature of to explain how to find the exact optimal options (products) with the data structure mentioned in [1]. Besides, from some observations of this problem, we propose advance method that ignore irrelevant users and candidate products to save time. At last we sample products that satisfy the constrain and use their maximal cover count to prune the .

**2. PROBLEM DEFINITION**

Given user dataset , , and a user preference weight vector set

**3. GEOMETRIC COMPUTATION NATURE**

Because the dataset is with limited number of products, for a user , it is easy to calculate the top-k score of among . Suppose is the score that rank top-k for , then if a new product wants to rank top-k for this user, in other words, covers this user, should follow . If we draw it when dimensionality , it would be shown as Figure 1.



**Figure 1**

From Figure 1, we can see that divides the candidate product space into 2 half-space, one is marked as “+”, which represents ; the other is marked as “-”, which represents . If and only if the product in the region “+” can cover .

**4. CELL TREE REPRESENTATION**

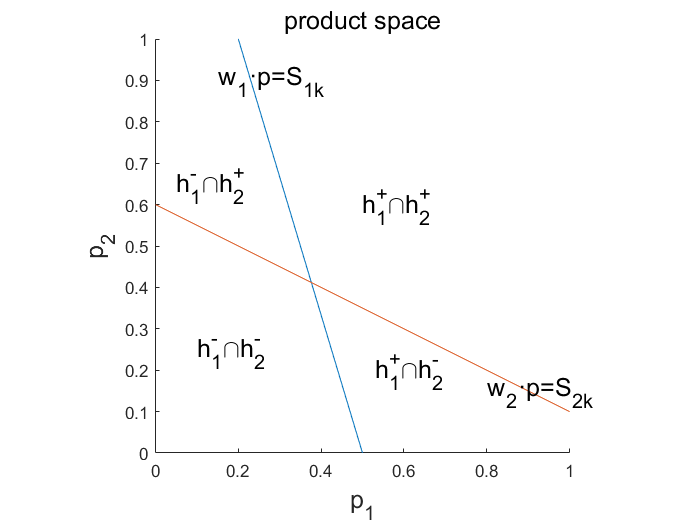
**4.1** **From One User to Many Users**

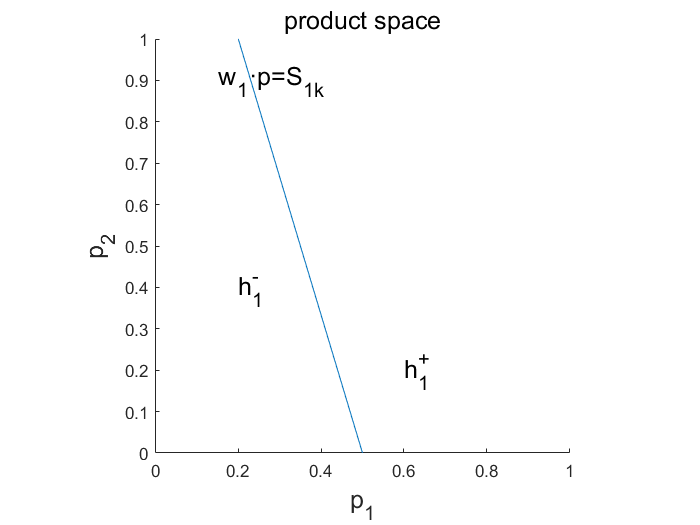
We have seen the example when there is 1 user, which will divide the space into 2 half-space. When it comes to multi users, such as 2 users (), firstly divides product space into 2 half-space as shown in Figure 2; on the basis of figure 2, divides and into 2 half-space respectively as shown in figure 3.

Region covers ; Region could only cover ; Region could only cover ; Region couldn’t cover any user.

For example, Region covers , so the cover count of is 2.

Problem 1 actually is finding the region which with the maximal cover count.





**Figure 2**

**Figure 3**

**4.2 Cell Tree Representation and Cell Insertion**

is a binary tree. We built a based on the example mentioned in 5.1

The root of the tree is the candidate option space which is limited by the constrain . Node represents the region . In practice, node will only use to present to save memory.

**5. BASELINE SOLUTION**

**5.1 Baseline Solution Example**

|  |  |
| --- | --- |
| 1. Deal with the constrain 𝐶(𝑝)≤𝐵 | |
|  |  |
| 2. to cover user , the space is divided into and . In CellTree, node generates two children, one is and the other is . | |
|  |  |
| 3. to cover user , the space is divided into and the space is divided into and . In , node generates two children, one is and the other is. | |
|  |  |
| 4. return region | |

**5.2 Baseline Solution Algorithm**

1. For each in , based on using cell tree divide model space , where is the score to rank top respecting to

2. count each divided region’s user cover number

3. Return the region that satisfies and with the greatest cover count

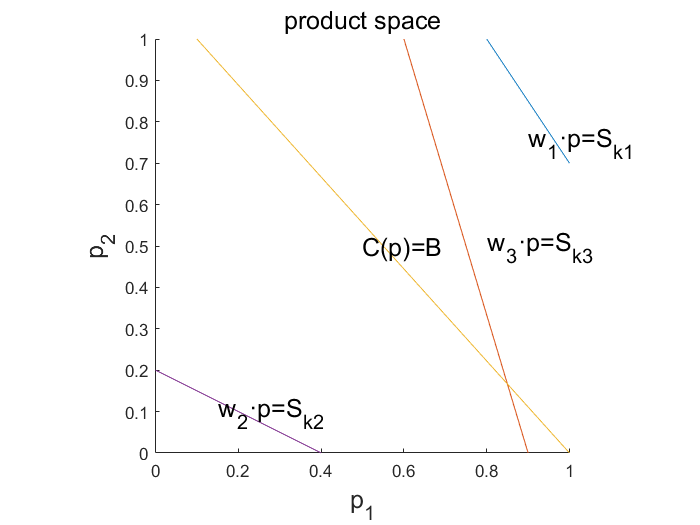
**6. OBSERVATIONS**

**6.1 Observation 1:**

The constrain is , we can only consider . This means whatever option found in there will always be a better option in .

**6.2 Observation 2:**

There are users that we don’t need to consider because we may never cover them under the constrain or we can always cover them if we choose the option on the constrain. Figure 4 shows the example.



**Figure 4**

If we decide to choose the new product from , we can see that also divides the product space into 2 half-spaces, but all the products on is in the “-” half-space, which means all of them can not cover . Different from , the products that on can always cover . From this observation, we can move out all the users that doesn’t intersect with .

**6.3 Observation 3:**

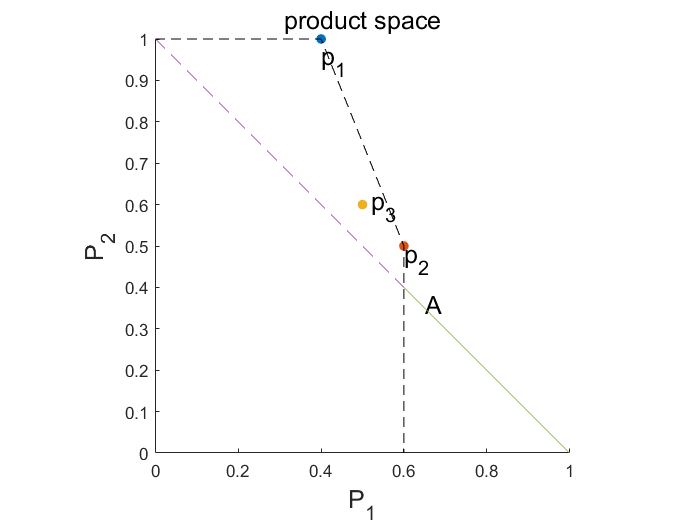
As we mentioned before, a product will cover some users and dataset will also do so, so we can ignore the products that already covered by .

**6.4 Observation 4:**

*1:*

Suppose is the convex hull defined by original point , point set and the projection points from points of to coordinate plane, then if a product is in convex hull and could cover weight vector , then one of can also cover .

Here we list an example for lemma 1. Suppose , , , original point (0, 0), points of projection point of coordinate plane is (0.4, 0), (0, 1), (0.6, 0), (0, 0.5), (0.5, 0), (0, 0.6). These points form the convex hull : as shown in figure 5.



**Figure 5**

We can see from figure 5 such that some part of the space is in (the dashed part).

Lemma1 says that for any product in , if covers a user , then one of {} (here more exactly is ) will also cover .

Because we already have the product dataset for and the points in will form a convex hull and covers some part of , we can ignore the part that covered by , which means when we built cell tree, because of the decreasing numbers the candidate options, the root of cell tree would become smaller.

Lemma 1 is inspire from ’s lemma 4. The proof of lemma 1 can be found in appendix proof 1.

**7. ADVANCE SOLUTION**

The different between base solution and advance solution is the 4 observations mentioned in part 7.

**7.1 Advance Solution Example**

|  |  |
| --- | --- |
| 1. discard all those 𝑤 that covered by 𝑃 | |
| 2. only consider the such tha intersects with constrain 𝐶(𝑝)=𝐵 | |
| (suppose after step 2, we only left ) | |
| 3. only consider | |
|  |  |
| 4. pruning for 𝐶(𝑝)=𝐵 with lemma 1, which means we discard the dashed part, only left part as shown in figure below. | |
|  |  |
| 5. consider , the intersect part is in convex full, directly mark region as | |
|  |  |
| 6. insert | |
|  |  |
| 7. return | |

**7.2 Advance Solution**

1. Preprocessing , discard all user that already covers:
2. we only consider
3. Find the biggest convex hull of point set (name it as
4. Preprocessing , on , discard the part which is in convex hull **(*see proof 1*)**
5. Preprocessing , discard all such that doesn’t intersect with
6. For each in , based on using cell tree divide model space , where is the score to rank top respecting to
7. count each divided region’s user cover number

8. Return the region that with the greatest cover count

**8. CONCLUSION**

For the final project we have done all of the theory part including formal problem definition, baseline solution from geometric computation nature, advance solution with observations and proof of observations. The next step is to complete the experiment part.

**9. REFERENCE**

1. Bo Tang, Kyriakos Mouratidis, Man Lung Yiu. [Determining the Impact Regions of Competing Options in Preference Space](https://acm.sustech.edu.cn/btang/pub/SIGMOD17_kSPR.pdf). Proceedings of the ACM Conference on Management of Data (SIGMOD), Chicago, Illinois, USA, May 2017.

2. Peng P, Wong RC-W (2015) k-hit query: top-k query with probabilistic utility function. In: Proceedings of the 2015 ACM SIGMOD international conference on management of data (SIGMOD). ACM, pp 577–592

**APPENDIX**

**A: PROOF OF LEMMAS/THEOREMS**

**Proof of Lemma 1:** Suppose point is the original point; point is inside convex hull; is the crossover point between and convex hull; on convex hull, is on the hyperplane . Then

Next，

,

Without loss of generality, suppose

Then

That is

Proof ends.