**Problem background:**

Take smartphone market as an example, there is different models of smartphone（）. Each model has different price, pixel, battery capacity, cooling capacity and etc.（）.

Now brand has models (), and have user data set . Each present the preference weight vector of a single user. The score of a model respecting to a user is the dot product . A model covers a user when the score of it ranks top among .

How to introduce a new product such that maximizes the cover ratio and the product creation cost is no greater than creation budget ?

**Problem Definition:**

*Input:*

global model dataset , ,

Another model dataset

Model creation budget ,

a positive integer ,

a creation cost function ,

a user preference tuple set ,

The score of a model respecting to a user is the dot product

A model covers a user when the score of it ranks top among .

*Output:*

find a new product such that satisfies the constrain

and maximize ’s cover ratio

**Hit:**

*The main difference between baseline and advance-1 is:*

1. advance-1 only consider weight vectors that not covered by

2. advance-1 only consider rather than

3. advance-1 only consider the weight vectors that intersects with

*The main difference between advance-1 and advance-2 is:*

1. advance-2 considers convex hull to reduce the range of

**Baseline solution example**

Take as example,

where is the score to rank top respecting to

1. In model space, only the half-space C(p)≤B is valid

2. If we want to cover , the score of the new model should greater than

The line divides the space into 2 part.

One is , where any model in it could cover ;

Another is , where any model in it couldn’t cover .

3. If we want to cover , the score of the new model should greater than

The line divides the space into 2 part.

One is , where any model in it could cover ;

Another is , where any model in it could only cover .

The line divides the space into 2 part.

One is , where any model in it could only cover ;

Another is , where any model in it couldn’t only any user.

4. Return region

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

**Baseline solution ():**

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

Time complexity: , where is the capacity of , is the dimension of a model.

**Advance solution version-1 example:**

Take as example,

where is the score to rank top respecting to

1. discard all weight vectors such that have been cover by :
2. In model space, only consider the space
3. The line divides the line into 2 part.

One is , where could cover ;Another is , where couldn’t cover .

1. The line divides the space into 2 part.

One is , could cover ; Another is , where could only cover .

1. Return line

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

**Advance solution version1:**

1. Preprocessing , discard all user that P already covers:
2. we only consider
3. Preprocessing , discard all such that doesn’t intersect with
4. For each in , based on using cell tree divide model space , where is the score to rank top respecting to
5. count each divided region’s user cover number
6. Return the region that satisfies and with the greatest cover count

Time complexity: ,

where is the number of such that not cover by and intersect with ;

is the dimension of a model.

**Advance solution version-2 example:**

Suppose and

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

As shown below, we only need the solid part of , which is

1. From line , we could know all of is in , which means could cover
2. The line divides the line into 2 part.

One is , could cover ; Another is , where couldn’t cover .

1. Return segmentation

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |

**Advance solution version-2:**

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

Time complexity: ,

where is the number of such that not cover by and intersect with ;

is the dimension of a model;

Compare with version1, version2’s is no more than version1’s .

**Proof 1:**

To proof:

if a model is covered by a convex hull and this model could cover weight vector ,

then one of the models that form this convex hull can also cover .

Proof:

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.

**Proof 1’s corresponding original lemma in** :

*Lemma 4. Let be a tuple in which is inside but is not on the surface of . Then, .*

*Proof of Lemma 4:*

*Firstly, we show that for any point and any (linear) function L*

*associated with its weight vector , is equal to the length of the projection of the vector from the origin to point on vector . This can be shown as follows. We denote the vector from the origin to point by . Let be the angle between vector and vector . . Since the norm of the weight vector is (by deﬁnition) (i.e., ), we have . Note that is equal to the length of the projection of vector on vector , which completes the proof.*

*Secondly, we show that . For any weight vector where its corresponding utility function is in L, there exists a point on the surface of such that the length of the projection of the vector from the origin to point on vector is greater than the length of the projection of the vector from the origin to point on vector . This means that . In other words, for any utility function in , there exists a point on the surface of such that . This implies that .*