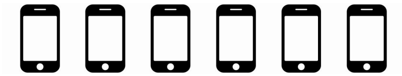
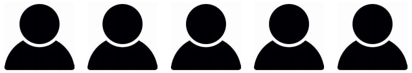
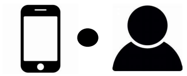
Cover Ratio Maximization

南方科技大学
SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

李可明,11612126@mail.sustech.edu.cn
学术导师: 唐 博

Problem Definition

- Given Product dataset D 
- user dataset W 
- $score(\boldsymbol{p_i}, \boldsymbol{w_j}) = \boldsymbol{p_i} \cdot \boldsymbol{w_j}$, 
- A product $\boldsymbol{p_i}$ **covers** a user $\boldsymbol{w_j}$ when its score ranks top-k respecting to $\boldsymbol{w_j}$

- How to introduce a product $\boldsymbol{p}$ that covers the most users under the constraint $C(\boldsymbol{p}) \leq B$

Cover Specific Users

	CPU	battery	w_1	w_2	w_3	a phone covers
g_1	0	1	0.219	0.531	0.817	w_3
h_1	0.695	0.606	0.675509	0.647741	0.618	w_2
h_2	1	0	0.781	0.469	0.182	w_1
h_3	0.872	0.344	0.756368	0.591632	0.44	w_1, w_2
p_1	0.506	0.566	0.51914	0.53786	0.555	$\emptyset$
p_2	0.228	0.777	0.348231	0.519519	0.677	w_3
w_1	0.781	0.219				
w_2	0.469	0.531				
w_3	0.183	0.817				

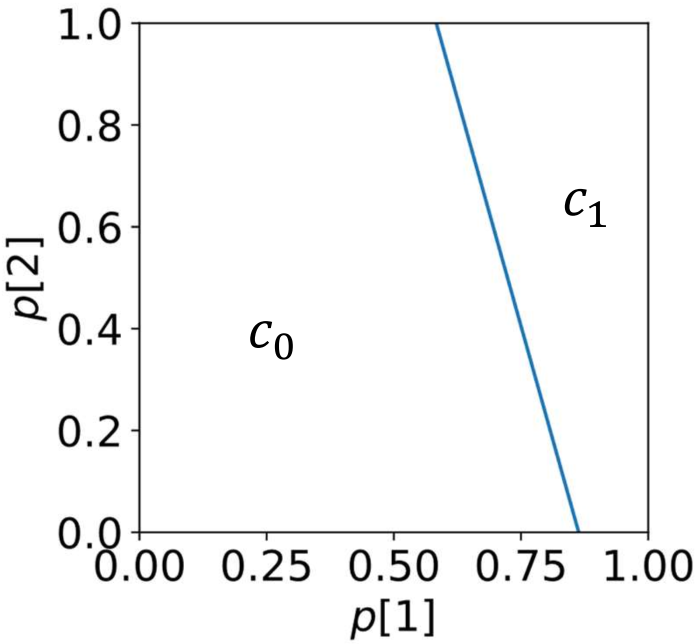
Assume the new product is p,

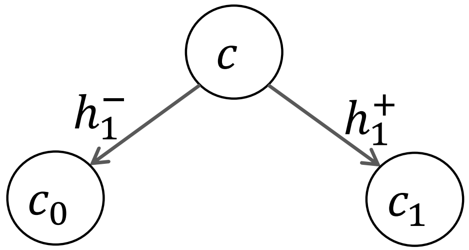
to rank top-2 for w_1, p should satisfy:

$$w_1 \cdot p \geq 0.756$$

- h_1: $0.781p[1] + 0.219p[2] = 0.756$
- h_2: $0.469p[1] + 0.531p[2] = 0.591$
- h_3: $0.183p[1] + 0.817p[2] = 0.677$

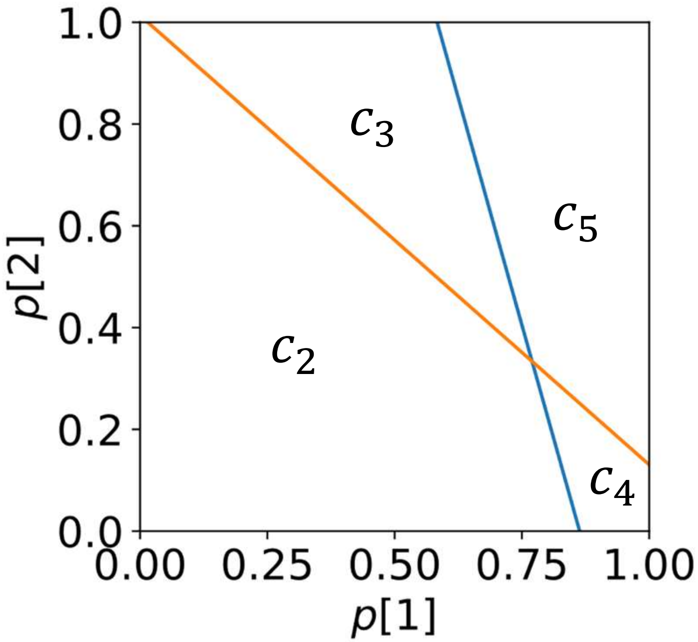
Cover w_1





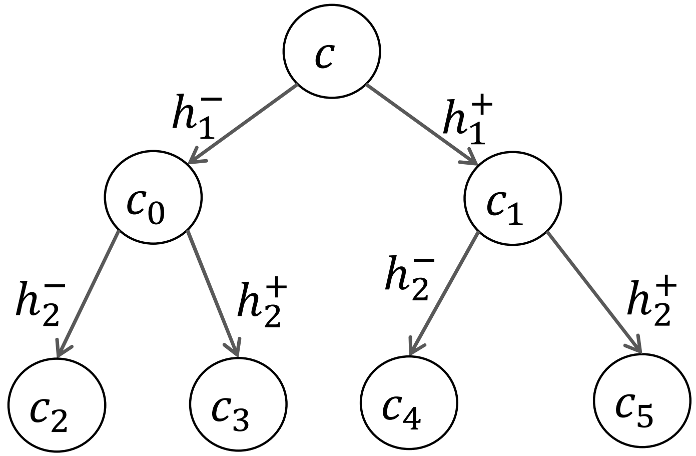
h_1: $0.781p[1] + 0.219p[2] = 0.756$

Cover w_2

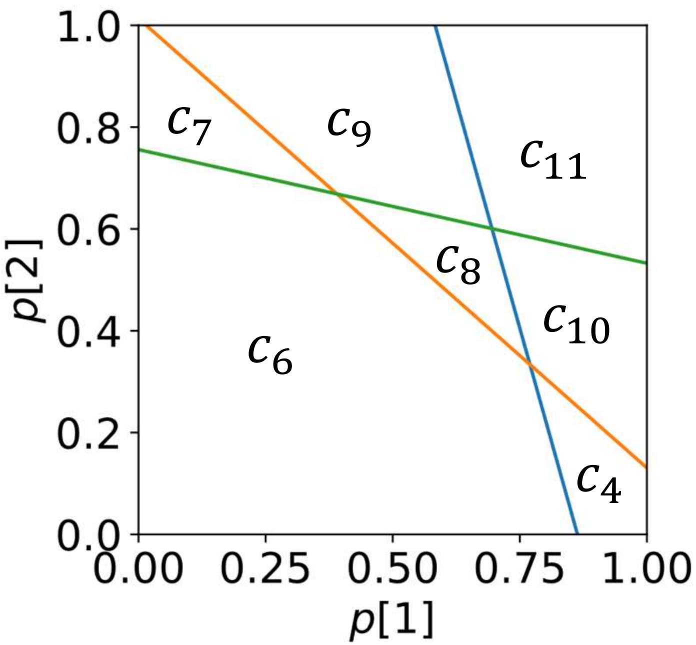


h_1: $0.781p[1] + 0.219p[2] = 0.756$

h_2: $0.469p[1] + 0.531p[2] = 0.591$



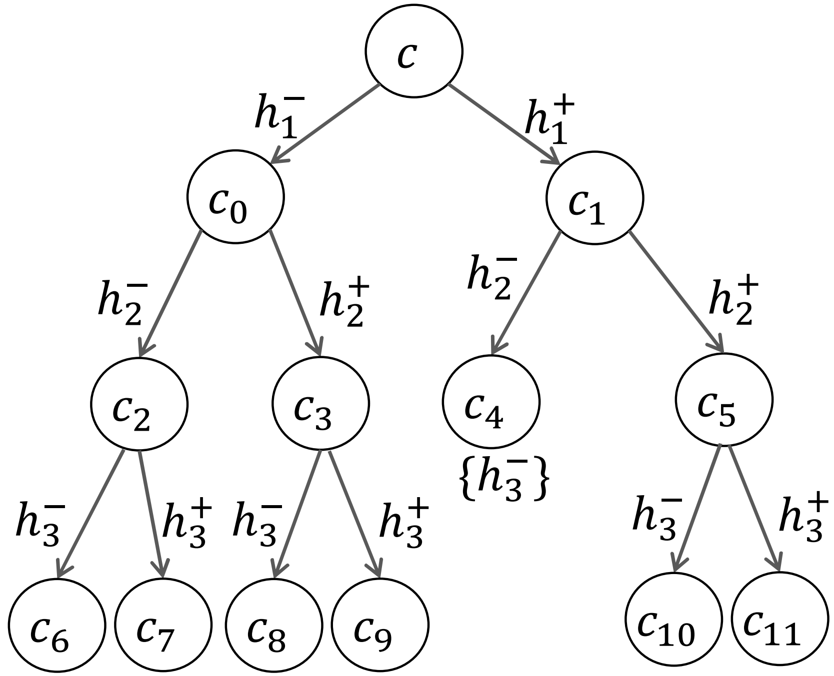
Cover w_3



h_1: $0.781p[1] + 0.219p[2] = 0.756$

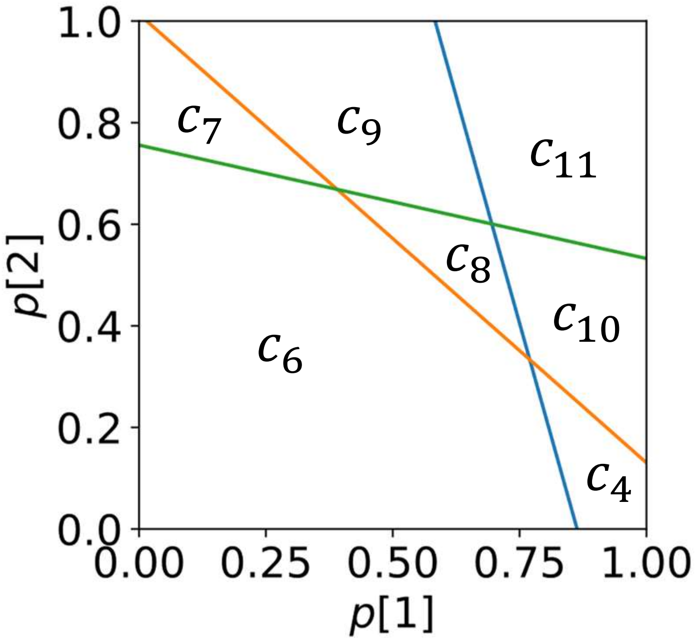
h_2: $0.469p[1] + 0.531p[2] = 0.591$

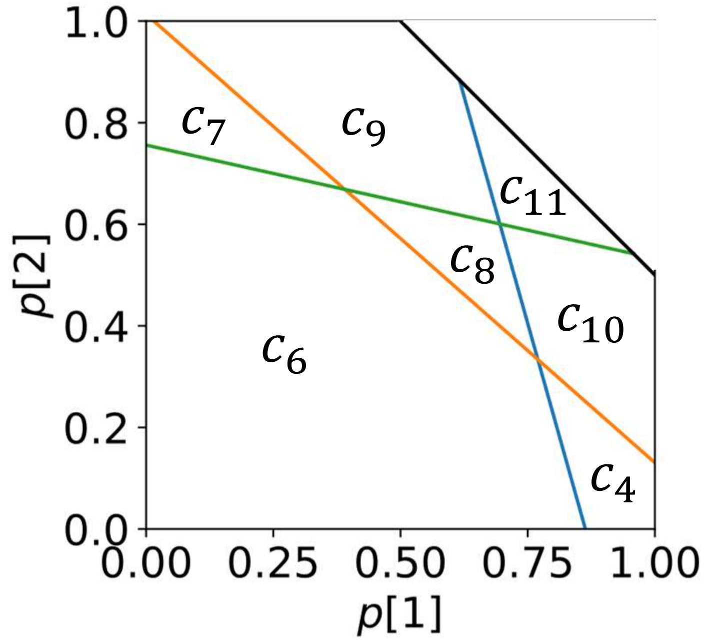
h_3: $0.183p[1] + 0.817p[2] = 0.677$



	c_6	c_7	c_8	c_9	c_4	c_{10}	c_{11}
covers	0	1	1	2	1	2	3

Constraint $C(p) \leq B$





h_1: $0.781p[1] + 0.219p[2] = 0.756$

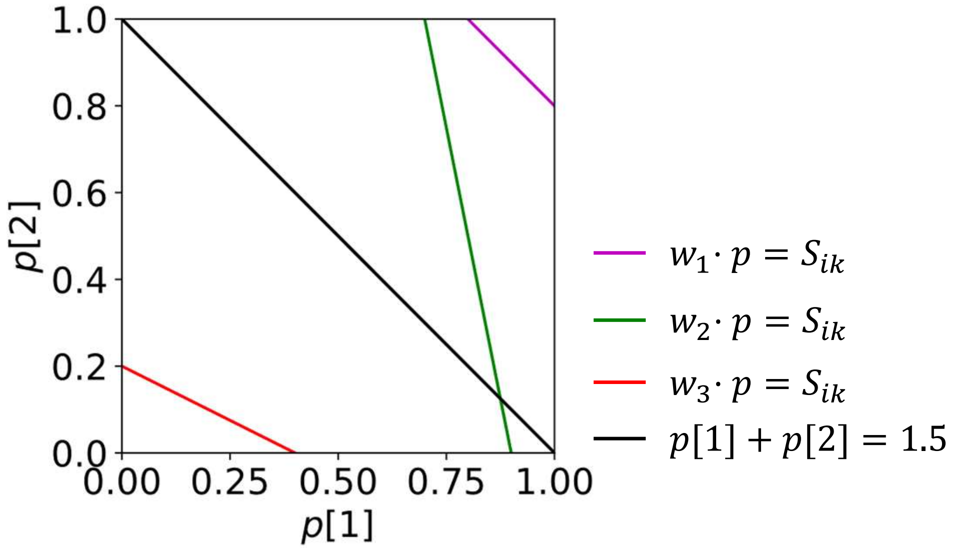
h_2: $0.469p[1] + 0.531p[2] = 0.591$

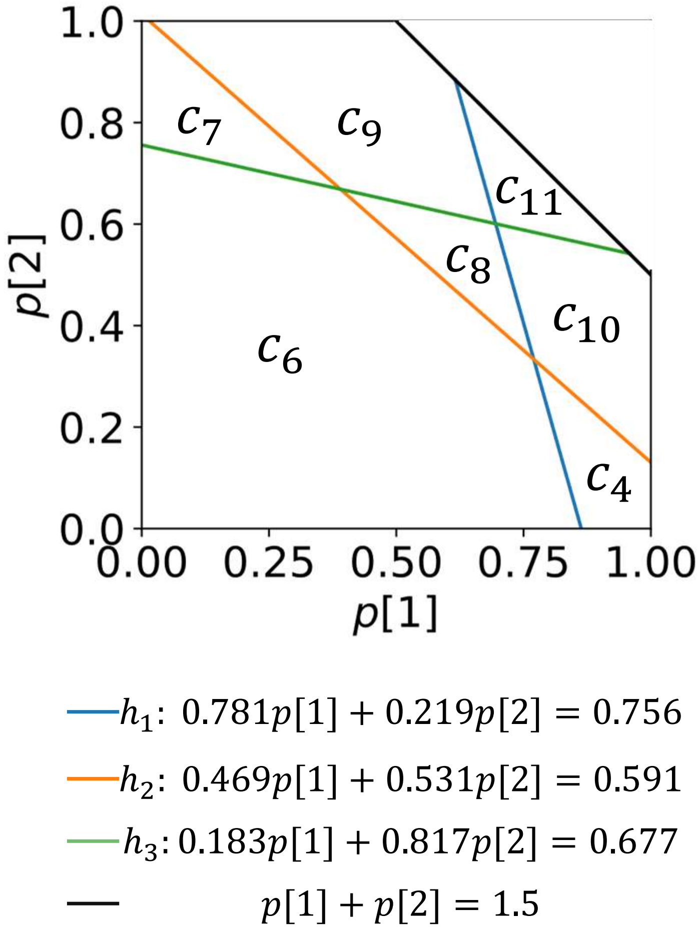
h_3: $0.183p[1] + 0.817p[2] = 0.677$

$p[1] + p[2] = 1.5$

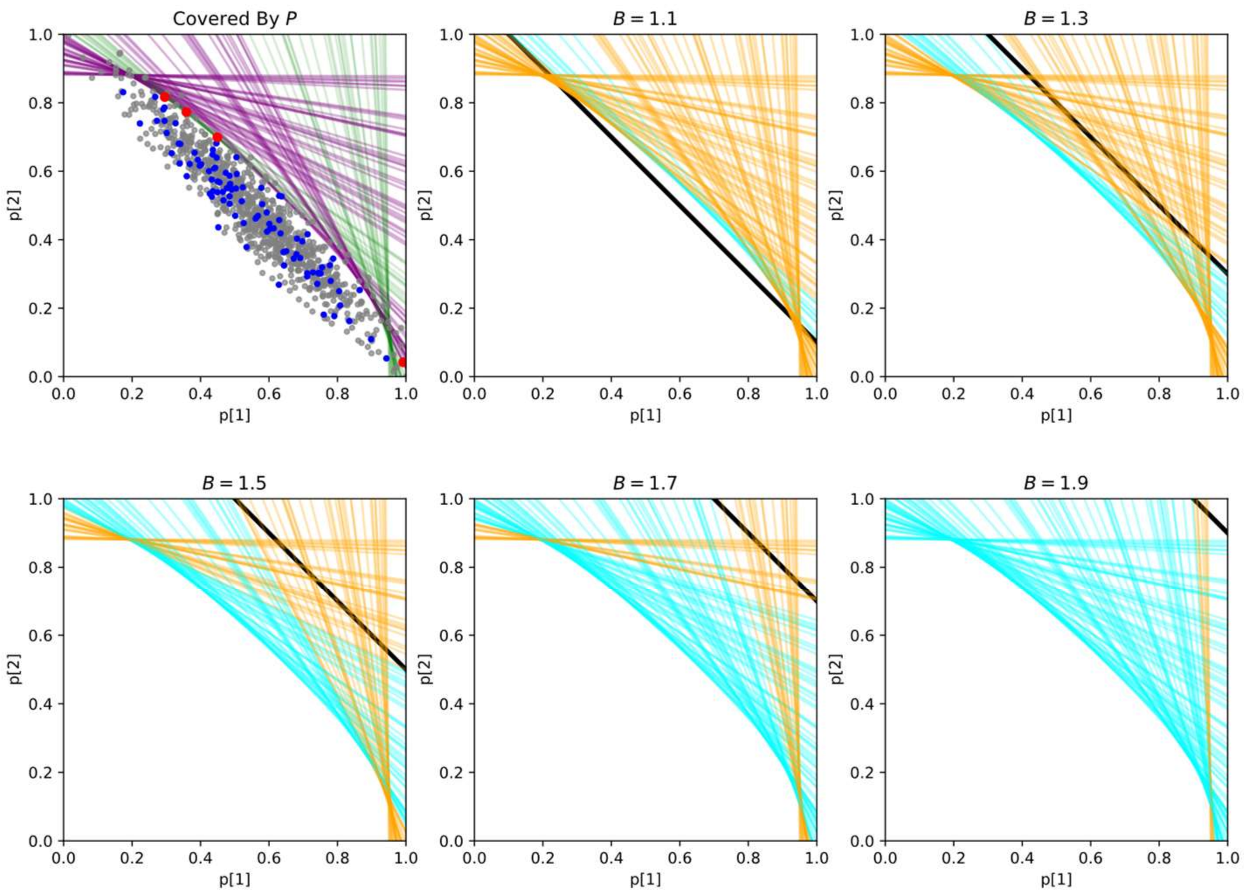
Remove Halfspaces Don't Intersect with $C(p) = B$

- **Lemma 1:** Only consider products on $C(p) = B$ as candidate solutions.
- **Lemma 2:** Remove halfspaces don't intersect with $C(p) = B$.





Experiment and Visualization of 2d Example



D: {eyepoints,
blue points,
red points},
P: {blue points,
red points}

red points: at least cover one user

green lines: halfspaces covered by

purple lines: halfspaces uncovered by P

orange lines: halfspaces intersects with constrain

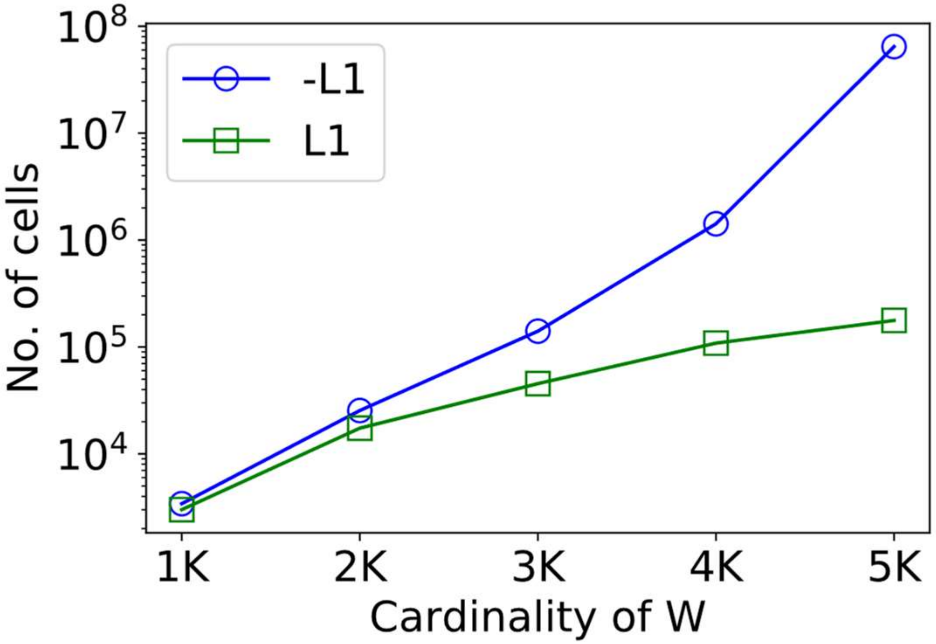
blue lines: halfspaces don't intersect with constrain

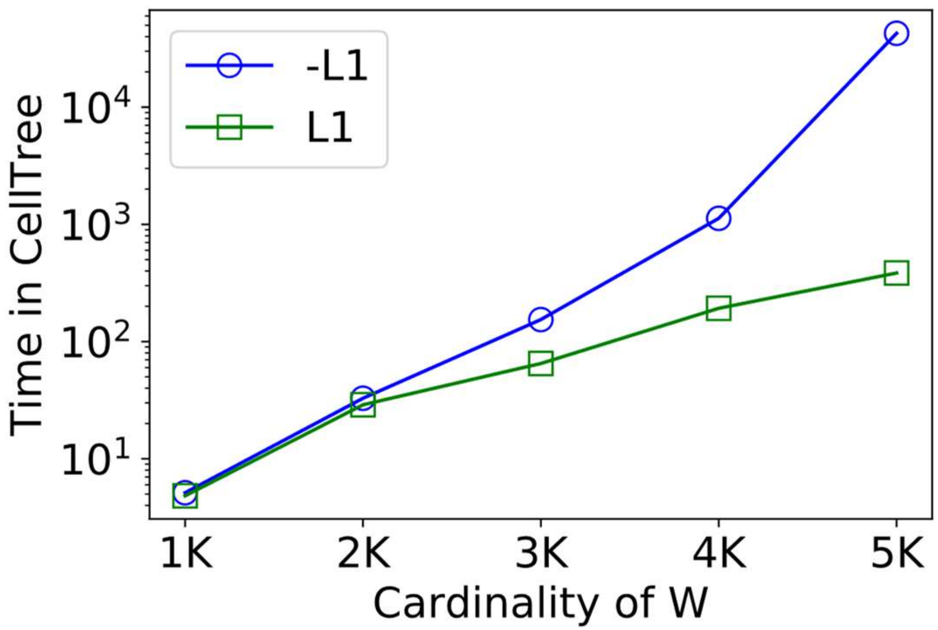
bold line: constrain boundary

Experiment Setting

- Product Dataset D:
  - HOTEL, d=4(No. of stars, No. of rooms, No. of facilities, Price), m=186637
- User Dataset W:
  - Uniformly sampling from hyperplane $\Sigma \mathrm{w[i]} = 1$
- $k = 10$, $B = 1.25$
- C++, $lp\_solve(lpsolve.sourceforge.net/5.5/)$
- Intel Xeon Gold 5122 3.60 GHz CPU, 128GB DDR4 RAM.

Effect of Lemma 1





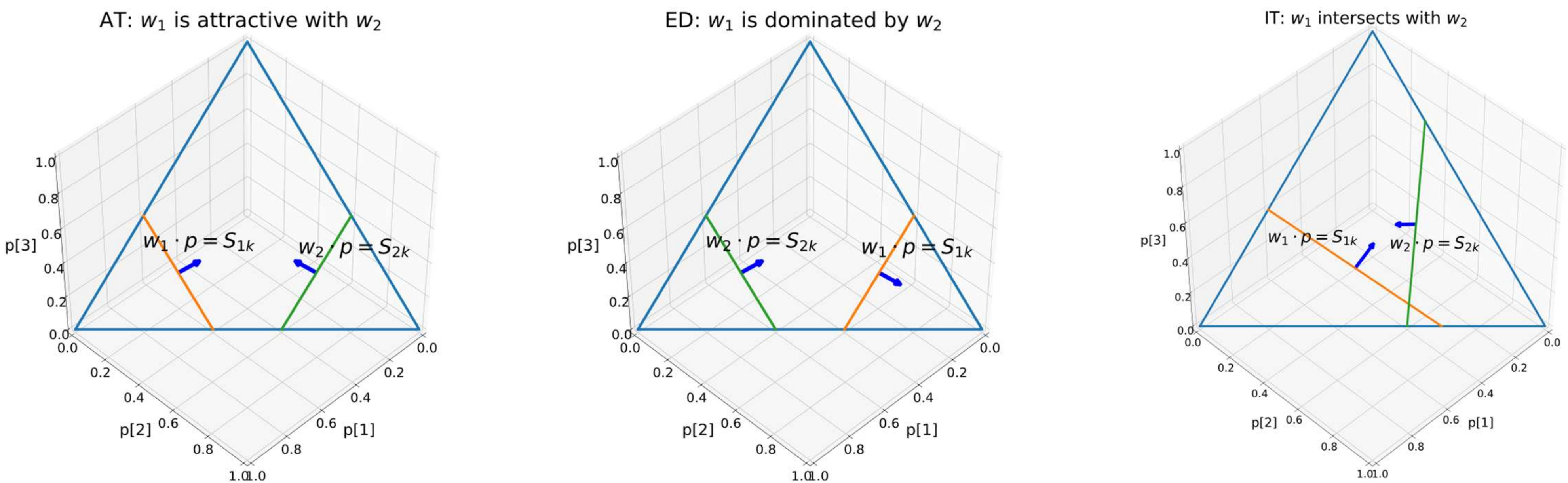
Lemma 1: Only consider the region $C(p) = B$

Upper Bounds and Lower Bounds

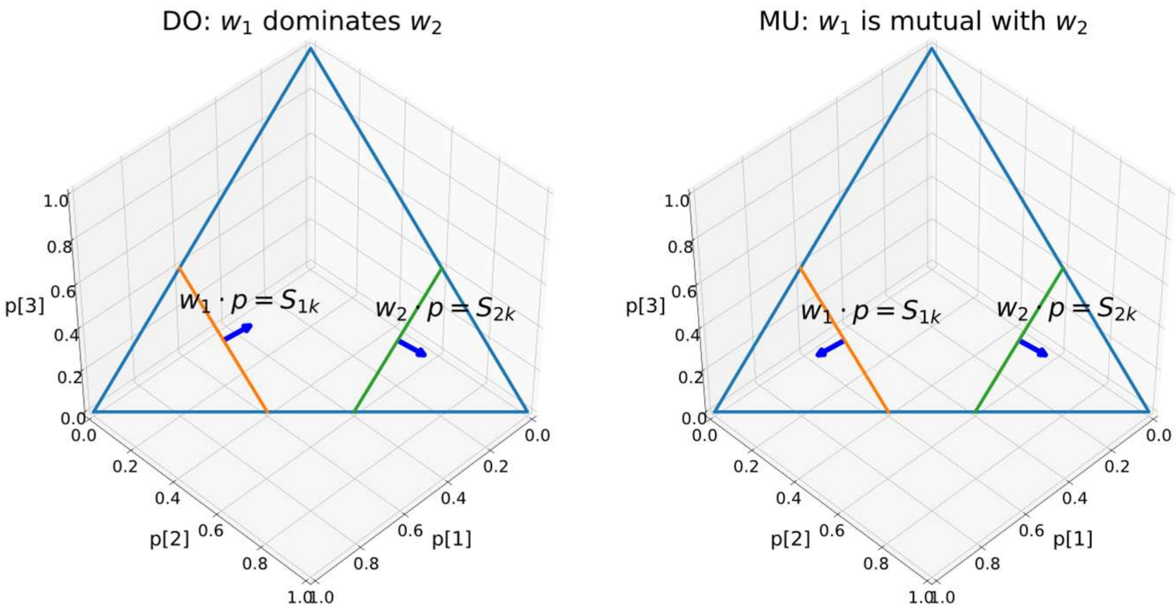
- **Definition 1:** The **lower** bound $\underline{\beta_{w_i}}$ of a user w_{i} means there is at least 1 candidate product covers $\underline{\beta_{w_i}}$ users and one of which is w_i.
- **Definition 2:** The **upper** bound $\overline{\beta_{w_i}}$ of a user w_{i} means the maximal of the cover counts of products that covers w_i.

- **Lemma 3:** Let $\beta = \max(\{\underline{\beta_{w_i}} | w_i \in W\})$, $\alpha = card(W) - \beta$, we prune the tree nodes that with more than α negative halfspaces.
- **Lemma 4:** Insert halfspaces into $CellTree$ based on ascending order of $\underline{\beta_{w_i}}$.

Get Users' Upper and Lower Bounds



Get Upper Bounds and Lower Bounds



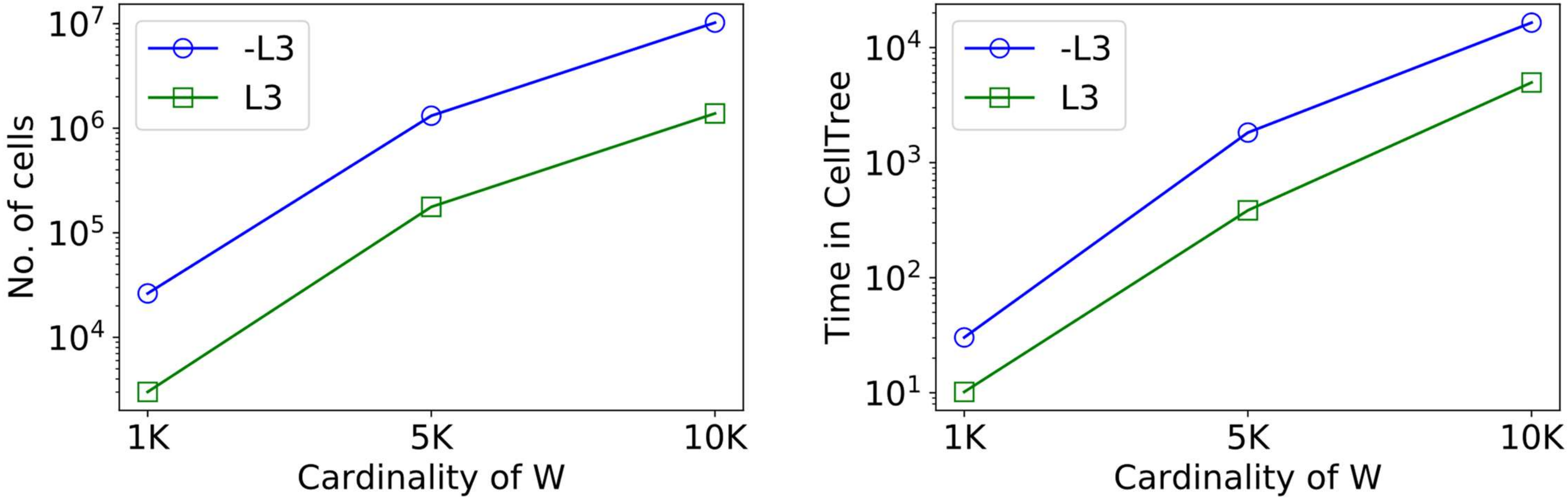
Lower Bound of w_1:
$card(ED \cup AT)$+1

Upper Bound of w_1:
$card(ED \cup AT \cup DO \cup IT)$+1

Time complexity: $O(cn^2)$,

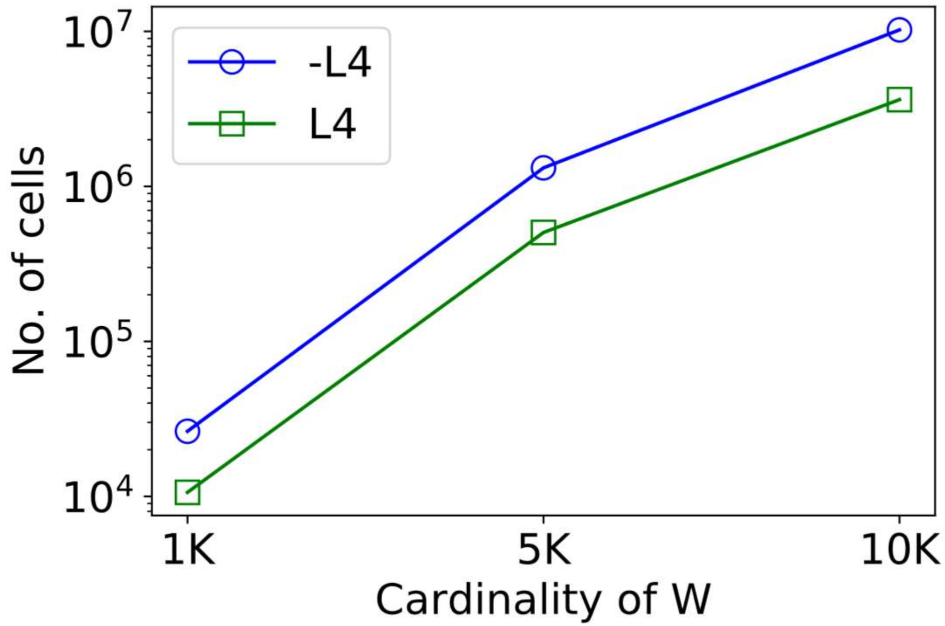
lp_solve: revised simplex method

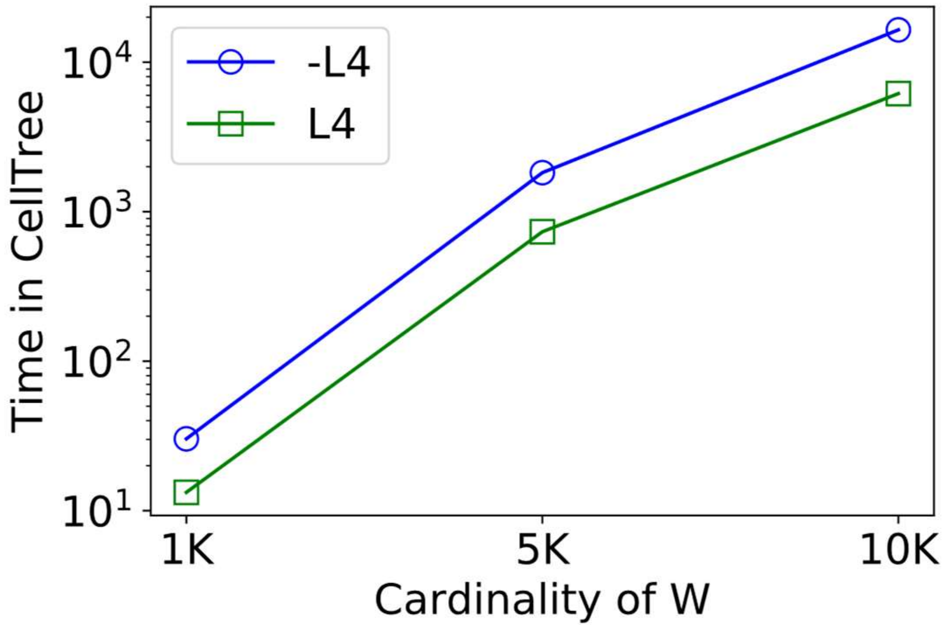
Effect of Lemma 3



Lemma 3: we prune the tree nodes that with more than α negative halfspaces.

Effect of Lemma 4





Lemma 4: Insert halfspaces into $CellTree$ based on ascending order of $\underline{\beta_{w_i}}$.

Upper Bounds and Lower Bounds

南方科技大学
SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

- **Definition 1:** the **lower** bound $\underline{\beta_{w_i}}$ of a user w_{i} means there is at least 1 candidate product covers $\underline{\beta_{w_i}}$ users and one of which is w_i.
- **Definition 2:** the **upper** bound $\overline{\beta_{w_i}}$ of a user w_{i} means the maximal of the cover counts of products that covers w_i.

- **Lemma 5:**

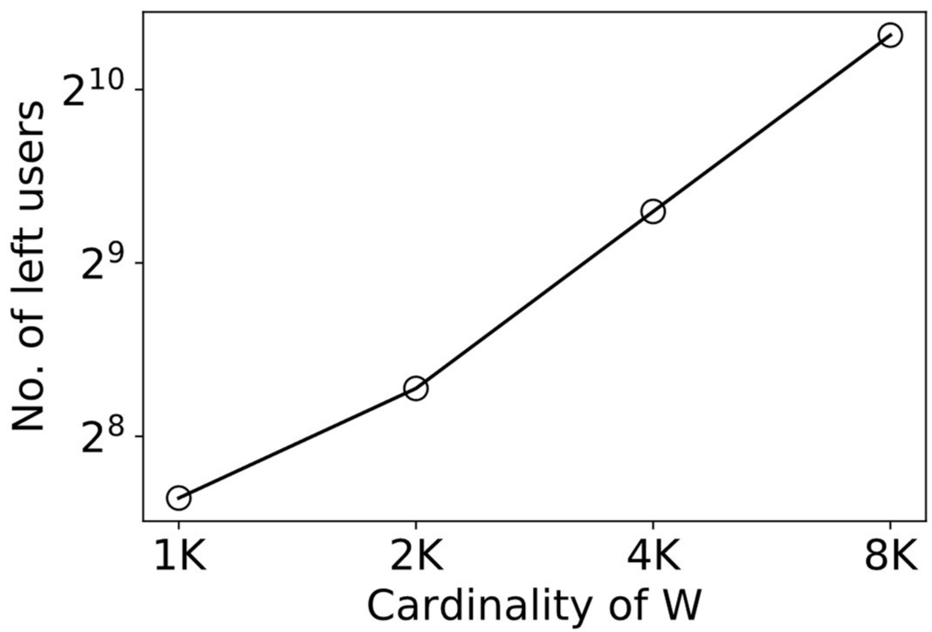
If there are 3 users, $W = \{w_1, w_2, w_3\}$ and

$$\underline{\beta_{w_1}} = 2,\ \underline{\beta_{w_2}} = 2,\ \overline{\beta_{w_3}} = 1,$$

which indicates $\underline{\beta_{w_1}} \geq \overline{\beta_{w_3}}$,

so the optimal new product p can't cover w_3 and we can remove w_3

Effect of Lemma 5



Conclusion

We use $CellTree$ mentioned in $kSPR$ as a baseline solution and propose 5 lemmas to efficiently solve $kCRM$.

Future Work:

1. Insertion order of halfspaces
2. More stringent upper bounds and lower bounds

- $kSPR$
- WHY−NOT REVERSE TOP−K QUERY
- $kCRM$

南方科技大学
SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

Q & A