

# Continuous Monitoring of Pareto Frontiers over Partially Ordered Attributes for Many Users

Afroza Sultana and Chengkai Li

*Innovative Database and Information Systems Research Lab*

*The University of Texas at Arlington*

EDBT 2018, Vienna, Austria

# Motivation

- Recommendation based on users' preferences

## Items customers view after viewing this item



★★★★★ 17 Reviews

HP Pavilion TouchSmart  
Laptop Computer With  
15.6" HD To...

~~\$549.99~~  
**\$399<sup>99</sup>** / each



★★★★★ 8 Reviews

HP Pavilion 15 Laptop  
Computer With 15.6"  
Screen & ...

**\$649<sup>99</sup>** / each



★★★★★ 4 Reviews

Toshiba Satellite® C55-B  
Laptop Computer With  
15.6"...

**\$469<sup>99</sup>** / each



★★★★★ 68 Reviews

Lenovo® Flex 2 (15)  
Dual-Mode Laptop  
Computer With 15.6&...

~~\$569.99~~  
**\$429<sup>99</sup>** / each







# Motivation

- Recommendation based on users' preferences
- Preferences with multiple attributes

*I prefer  
16" to 14"  
display*

## Items customers view after viewing this item

			
★★★★★ 17 Reviews	★★★★★ 8 Reviews	★★★★★ 4 Reviews	★★★★★ 68 Reviews
HP Pavilion TouchSmart Laptop Computer With 15.6" HD To...	HP Pavilion 15 Laptop Computer With 15.6" Screen & ...	Toshiba Satellite® C55-B Laptop Computer With 15.6"...	Lenovo® Flex 2 (15) Dual-Mode Laptop Computer With 15.6&...
\$549.99	\$649 <sup>99</sup> / each	\$469 <sup>99</sup> / each	\$569.99
<b>\$399<sup>99</sup></b> / each			<b>\$429<sup>99</sup></b> / each

# Motivation

- Recommendation based on users' preferences
- Preferences with multiple attributes
- Goal: objects that "stand out"

*I prefer  
16" to 14"  
display*

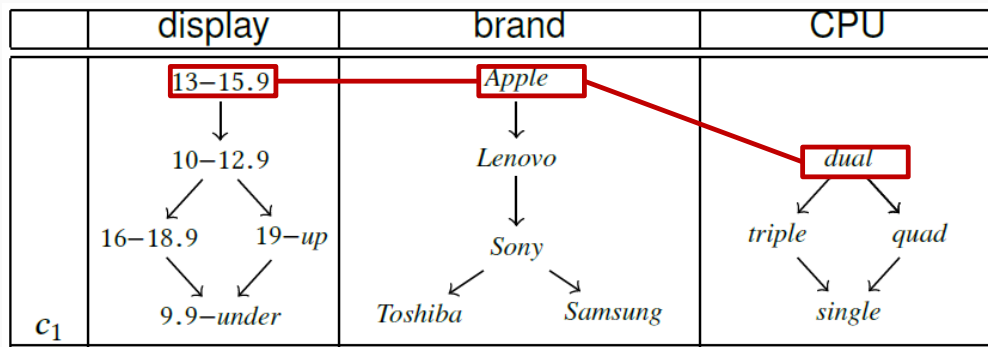


# An Example

	display	brand	CPU
$c_1$	<pre> graph TD     A["13-15.9"] --&gt; B["10-12.9"]     B --&gt; C["16-18.9"]     B --&gt; D["19-up"]     C --&gt; E["9.9-under"]     D --&gt; E           </pre>	<pre> graph TD     A["Apple"] --&gt; B["Lenovo"]     B --&gt; C["Sony"]     C --&gt; D["Toshiba"]     C --&gt; E["Samsung"]           </pre>	<pre> graph TD     A["dual"] --&gt; B["triple"]     A --&gt; C["quad"]     B --&gt; D["single"]     C --&gt; D           </pre>

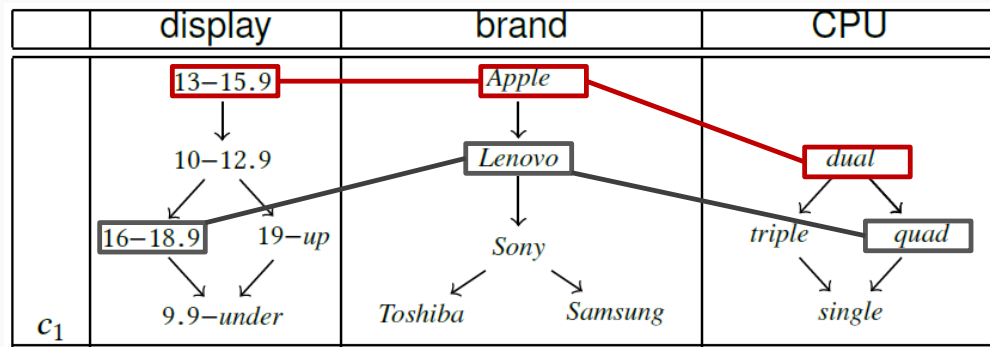
	display	brand	CPU
$o_1$	12	<i>Apple</i>	<i>single</i>
$o_2$	14	<i>Apple</i>	<i>dual</i>
...	...	...	...
$o_7$	16.5	<i>Lenovo</i>	<i>quad</i>

# An Example



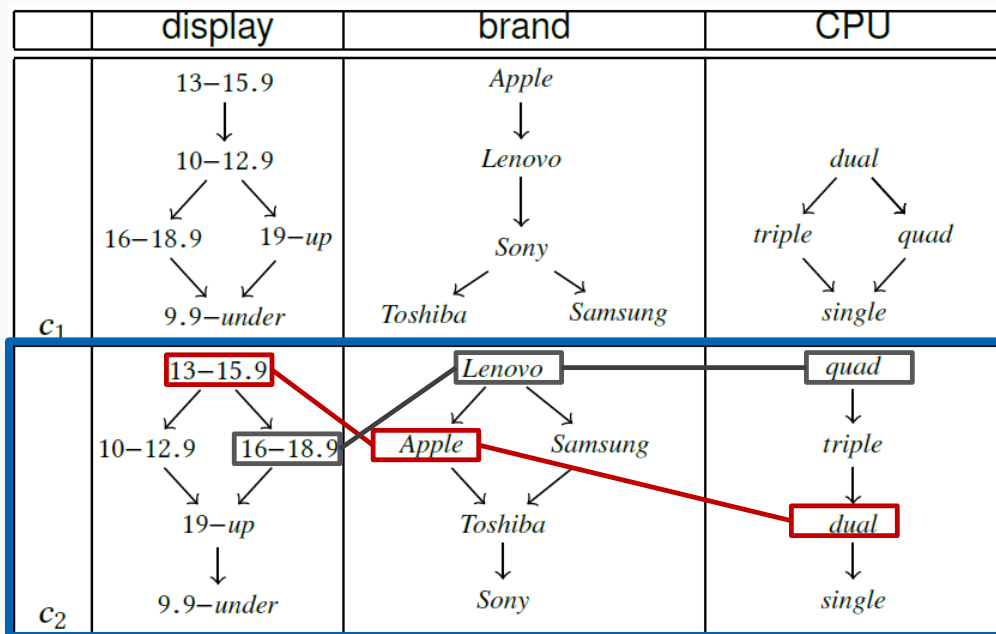
	display	brand	CPU
$o_1$	12	Apple	single
$o_2$	14	Apple	dual
...	...	...	...
$o_7$	16.5	Lenovo	quad

# An Example



	display	brand	CPU
$o_1$	12	Apple	single
$o_2$	14	Apple	dual
...	...	...	...
$o_7$	16.5	Lenovo	quad

# An Example



	display	brand	CPU
$o_1$	12	Apple	single
$o_2$	14	Apple	dual
...	...	...	...
$o_7$	16.5	Lenovo	quad



# Problem Formulation

Set of attributes  $\mathcal{D}$

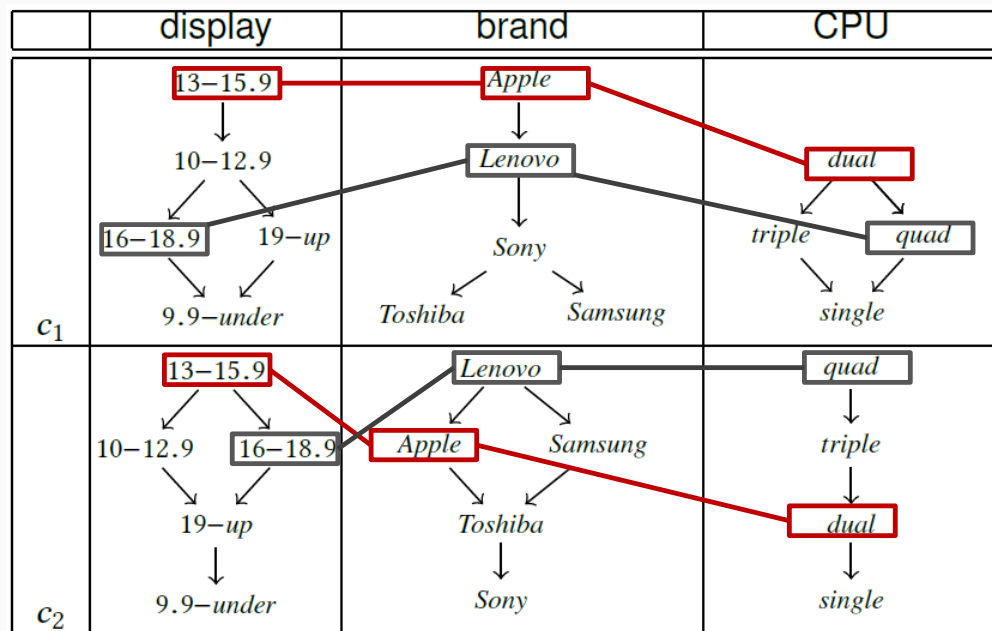
	display	brand	CPU
$c_1$	<pre> graph TD     A["13-15.9"] --&gt; B["10-12.9"]     B --&gt; C["16-18.9"]     B --&gt; D["19-up"]     C --&gt; E["9.9-under"]     D --&gt; E         </pre>	<pre> graph TD     A["Apple"] --&gt; B["Lenovo"]     B --&gt; C["Sony"]     C --&gt; D["Toshiba"]     C --&gt; E["Samsung"]         </pre>	<pre> graph TD     A["dual"] --&gt; B["triple"]     A --&gt; C["quad"]     B --&gt; D["single"]     C --&gt; D         </pre>
$c_2$	<pre> graph TD     A["13-15.9"] --&gt; B["10-12.9"]     A --&gt; C["16-18.9"]     B --&gt; D["19-up"]     C --&gt; D     D --&gt; E["9.9-under"]         </pre>	<pre> graph TD     A["Lenovo"] --&gt; B["Apple"]     A --&gt; C["Samsung"]     B --&gt; D["Toshiba"]     C --&gt; D     D --&gt; E["Sony"]         </pre>	<pre> graph TD     A["quad"] --&gt; B["triple"]     B --&gt; C["dual"]     C --&gt; D["single"]         </pre>

	display	brand	CPU
$o_1$	12	Apple	single
$o_2$	14	Apple	dual
...	...	...	...
$o_7$	16.5	Lenovo	quad

Append-only object  
table  $O$

Users' preferences

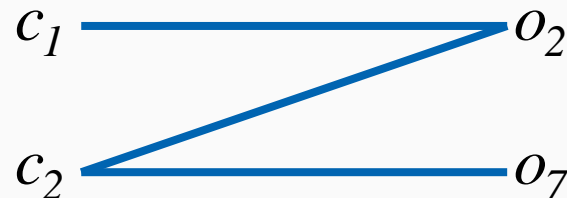
# Problem Formulation



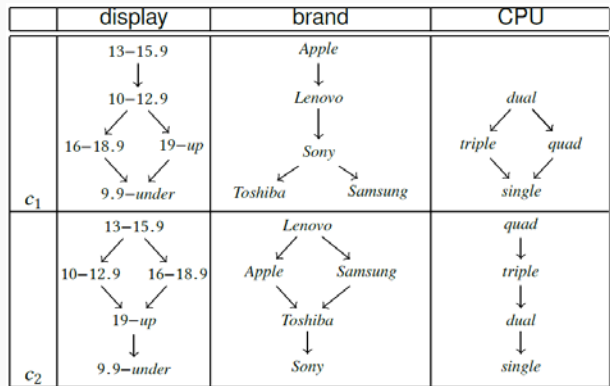
	display	brand	CPU
$o_1$	12	Apple	single
$o_2$	14	Apple	dual
...	...	...	...
$o_7$	16.5	Lenovo	quad

Target users

Pareto frontier  $\mathcal{P}_c$



# Problem Formulation; Continuous Object Dissemination



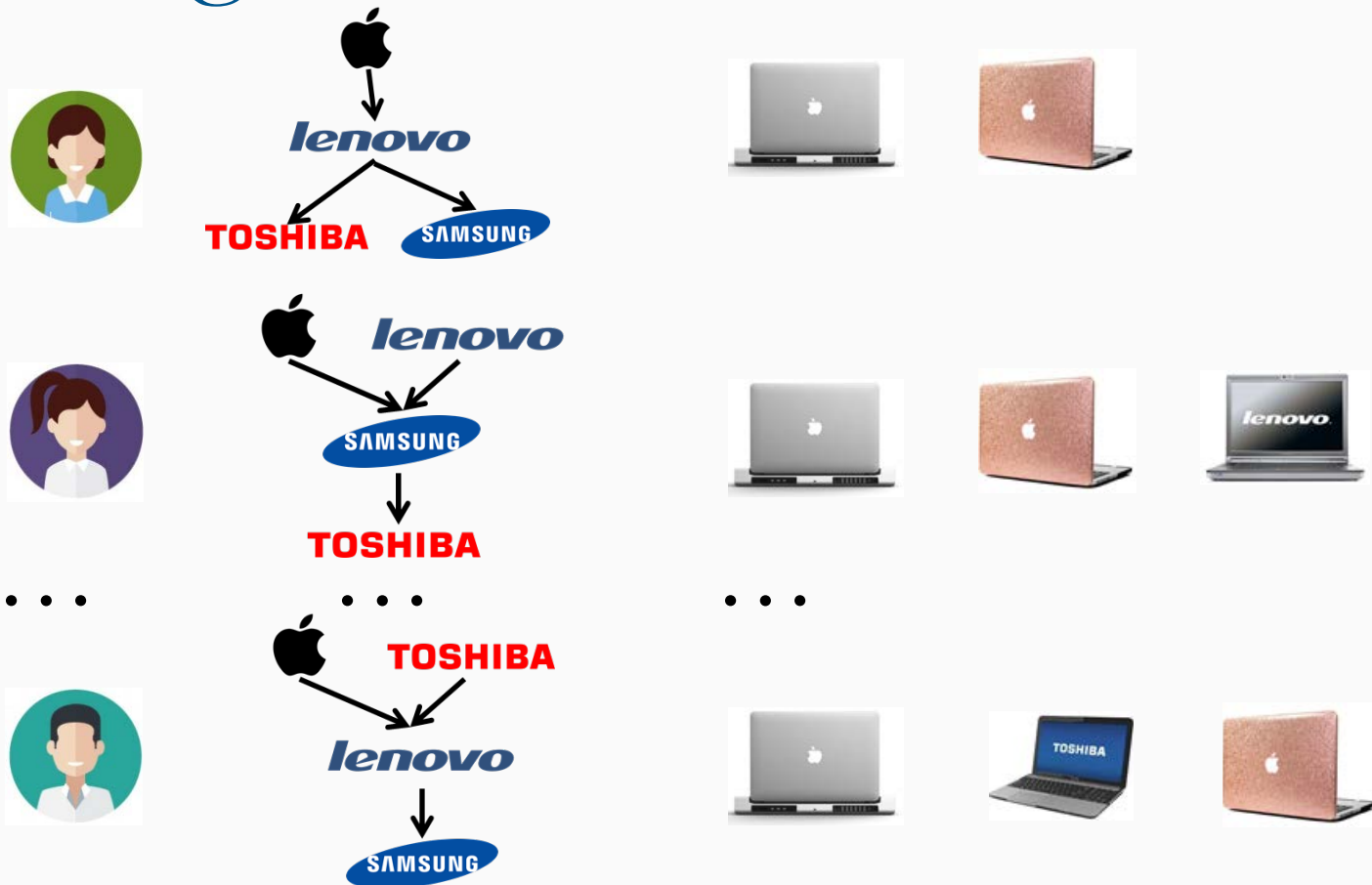
Find target users such that  $o_7$  is in the Pareto frontier.

	display	brand	CPU
$o_1$	12	Apple	single
$o_2$	14	Apple	dual
...	...	...	...
$o_7$	16.5	Lenovo	quad

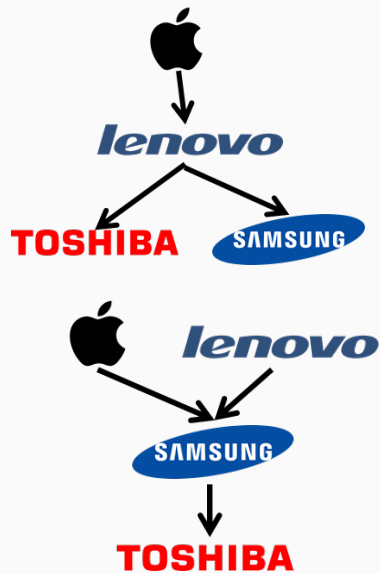


$c_2$

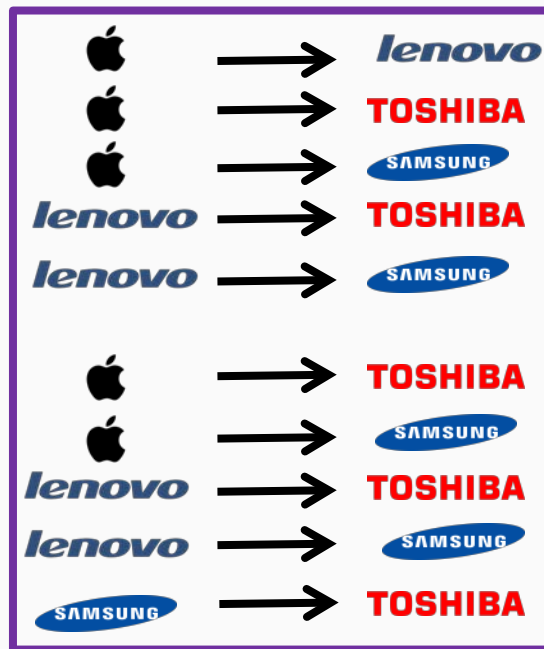
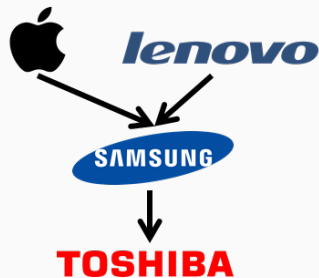
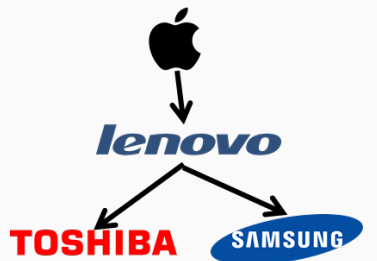
# Challenges & Ideas



# Challenges & Ideas

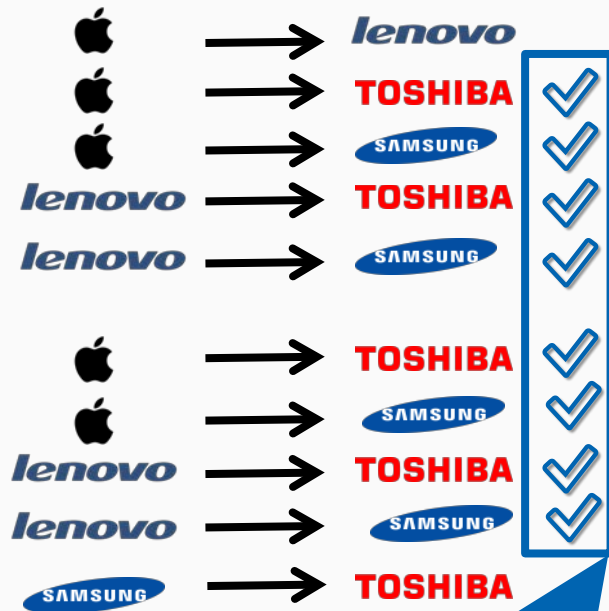
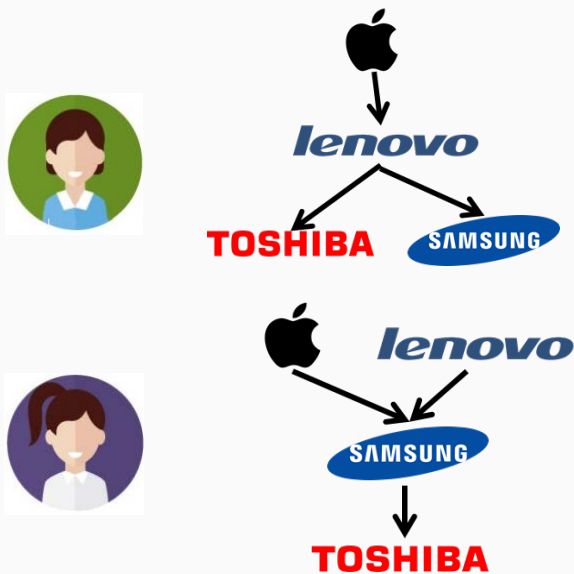


# Challenges & Ideas



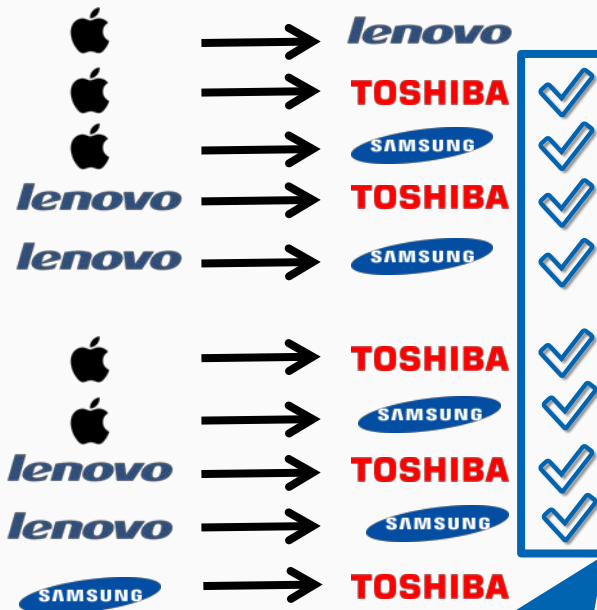
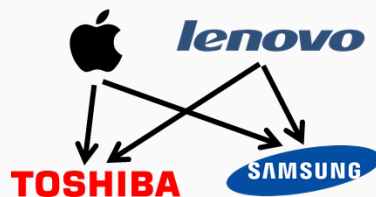
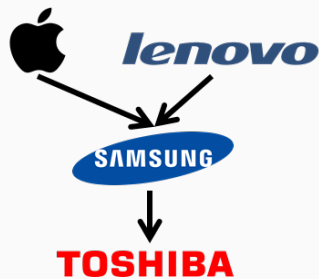
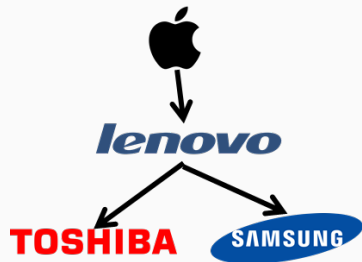
Preference tuples

# Challenges & Ideas



Common preference tuples

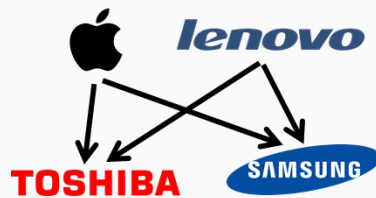
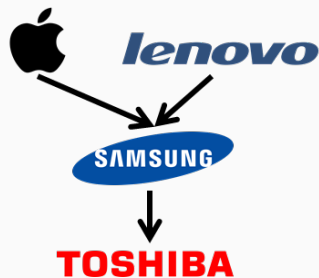
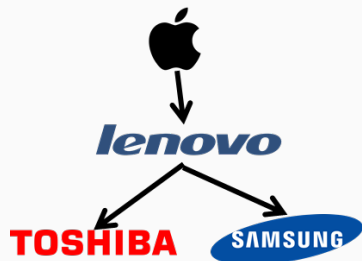
# Challenges & Ideas



Common preference tuples

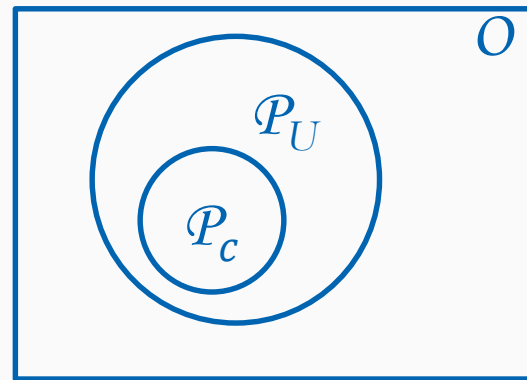


# Challenges & Ideas



# Challenges & Ideas

- ❑ Theorem 1:  $\mathcal{P}_U \supseteq \mathcal{P}_c$
- ❑ Lemma 1:  $\mathcal{P}_c$  w.r.t.  $O = \mathcal{P}_c$  w.r.t.  $\mathcal{P}_U$
- ❑ Recall & precision: 100%

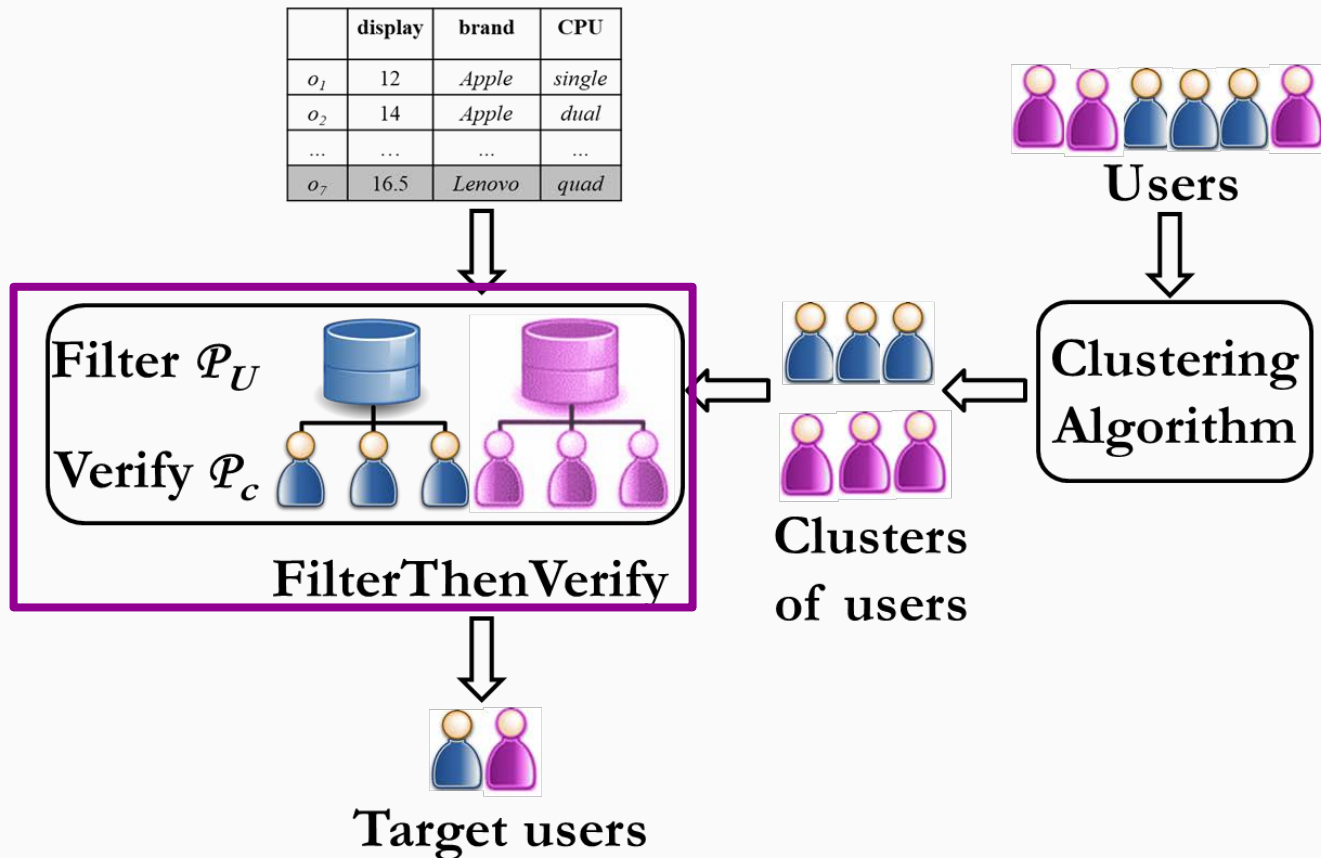


Sharing computation across users

# Challenge and Ideas

- ❑ Which users share preferences?
  - ✓ Cluster users based on preferences
- ❑ No prior study on clustering for partial orders
  - ✓ Study clustering partial orders

# System Architecture



# FilterThenVerify

For each cluster in  $\mathcal{C}$

- **Filter:** if  $U$  approve  $o$  in Pareto-optimality, stores  $o$  in  $\mathcal{P}_U$
- **Verify:** for each  $c$ , determines whether  $o$  belongs to  $\mathcal{P}_c$

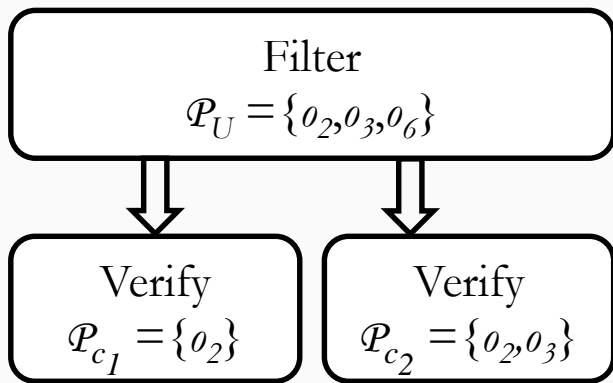


# FilterThenVerify

	display	brand	CPU
$c_1$	<pre>       13-15.9         ↓      10-12.9     ↙      ↘   16-18.9  19-up     ↘      ↙     9.9-under           </pre>	<pre>       Apple         ↓      Lenovo         ↓       Sony     ↙      ↘   Toshiba  Samsung           </pre>	<pre>       dual     ↙      ↘   triple    quad     ↘      ↙     single           </pre>
$c_2$	<pre>       13-15.9     ↙      ↘   10-12.9  16-18.9     ↘      ↙     19-up         ↓     9.9-under           </pre>	<pre>       Lenovo     ↙      ↘   Apple    Samsung     ↘      ↙   Toshiba         ↓       Sony           </pre>	<pre>       quad         ↓      triple         ↓       dual         ↓      single           </pre>
$U$	<pre>       13-15.9     ↙      ↘   10-12.9  16-18.9     ↓      ↙   19-up     ↘      ↙     9.9-under           </pre>	<pre>       Apple  Lenovo     ↙  ↘  ↙  ↘   Toshiba Sony Samsung           </pre>	<pre>   dual triple quad     ↘   ↓   ↙     single           </pre>

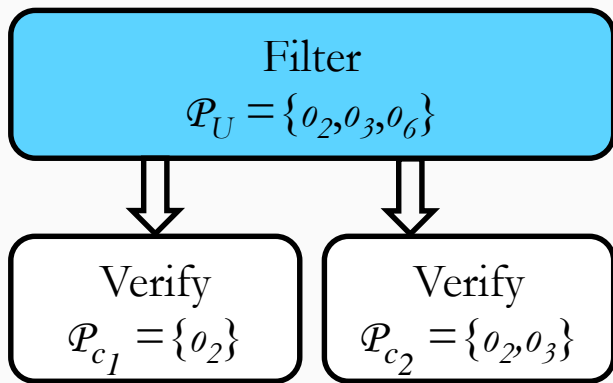
	display	brand	CPU
$o_1$	12	Apple	single
$o_2$	14	Apple	dual
$o_3$	15	Samsung	dual
$o_4$	19	Toshiba	quad
$o_5$	9	Samsung	quad
$o_6$	9.5	Lenovo	triple
$o_7$	16.5	Lenovo	quad

# FilterThenVerify



	display	brand	CPU
$o_1$	12	<i>Apple</i>	<i>single</i>
$o_2$	14	<i>Apple</i>	<i>dual</i>
$o_3$	15	<i>Samsung</i>	<i>dual</i>
$o_4$	19	<i>Toshiba</i>	<i>quad</i>
$o_5$	9	<i>Samsung</i>	<i>quad</i>
$o_6$	9.5	<i>Lenovo</i>	<i>triple</i>
$o_7$	16.5	<i>Lenovo</i>	<i>quad</i>

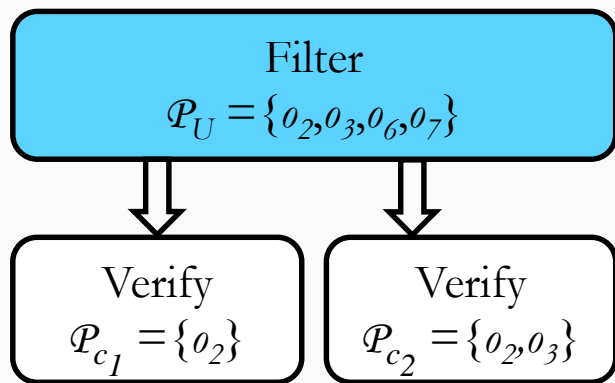
# FilterThenVerify



	display	brand	CPU
$o_1$	12	<i>Apple</i>	<i>single</i>
$o_2$	14	<i>Apple</i>	<i>dual</i>
$o_3$	15	<i>Samsung</i>	<i>dual</i>
$o_4$	19	<i>Toshiba</i>	<i>quad</i>
$o_5$	9	<i>Samsung</i>	<i>quad</i>
$o_6$	9.5	<i>Lenovo</i>	<i>triple</i>
$o_7$	16.5	<i>Lenovo</i>	<i>quad</i>

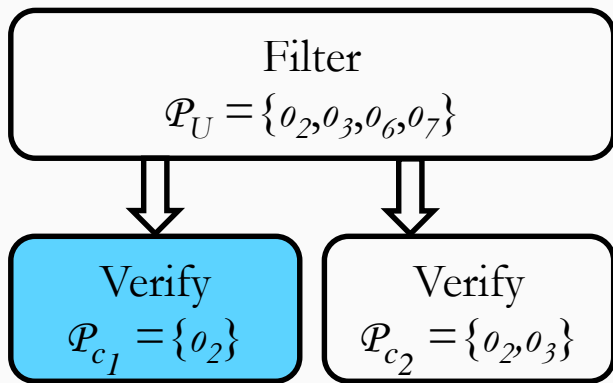


# FilterThenVerify



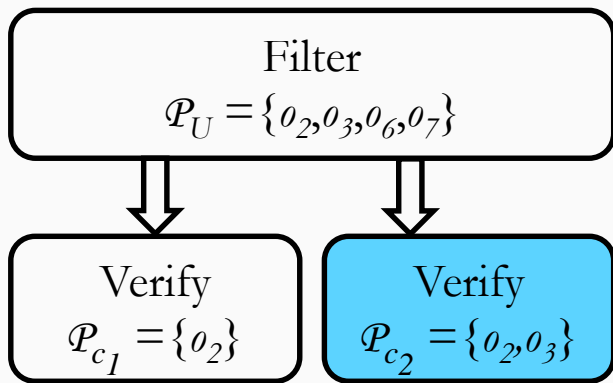
	display	brand	CPU
$o_1$	12	<i>Apple</i>	<i>single</i>
$o_2$	14	<i>Apple</i>	<i>dual</i>
$o_3$	15	<i>Samsung</i>	<i>dual</i>
$o_4$	19	<i>Toshiba</i>	<i>quad</i>
$o_5$	9	<i>Samsung</i>	<i>quad</i>
$o_6$	9.5	<i>Lenovo</i>	<i>triple</i>
$o_7$	16.5	<i>Lenovo</i>	<i>quad</i>

# FilterThenVerify



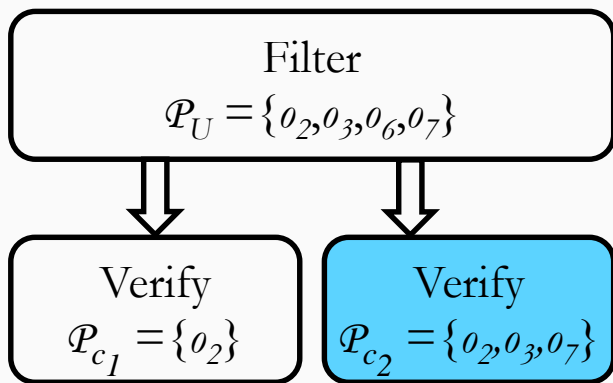
	display	brand	CPU
$o_1$	12	<i>Apple</i>	<i>single</i>
$o_2$	14	<i>Apple</i>	<i>dual</i>
$o_3$	15	<i>Samsung</i>	<i>dual</i>
$o_4$	19	<i>Toshiba</i>	<i>quad</i>
$o_5$	9	<i>Samsung</i>	<i>quad</i>
$o_6$	9.5	<i>Lenovo</i>	<i>triple</i>
$o_7$	16.5	<i>Lenovo</i>	<i>quad</i>

# FilterThenVerify



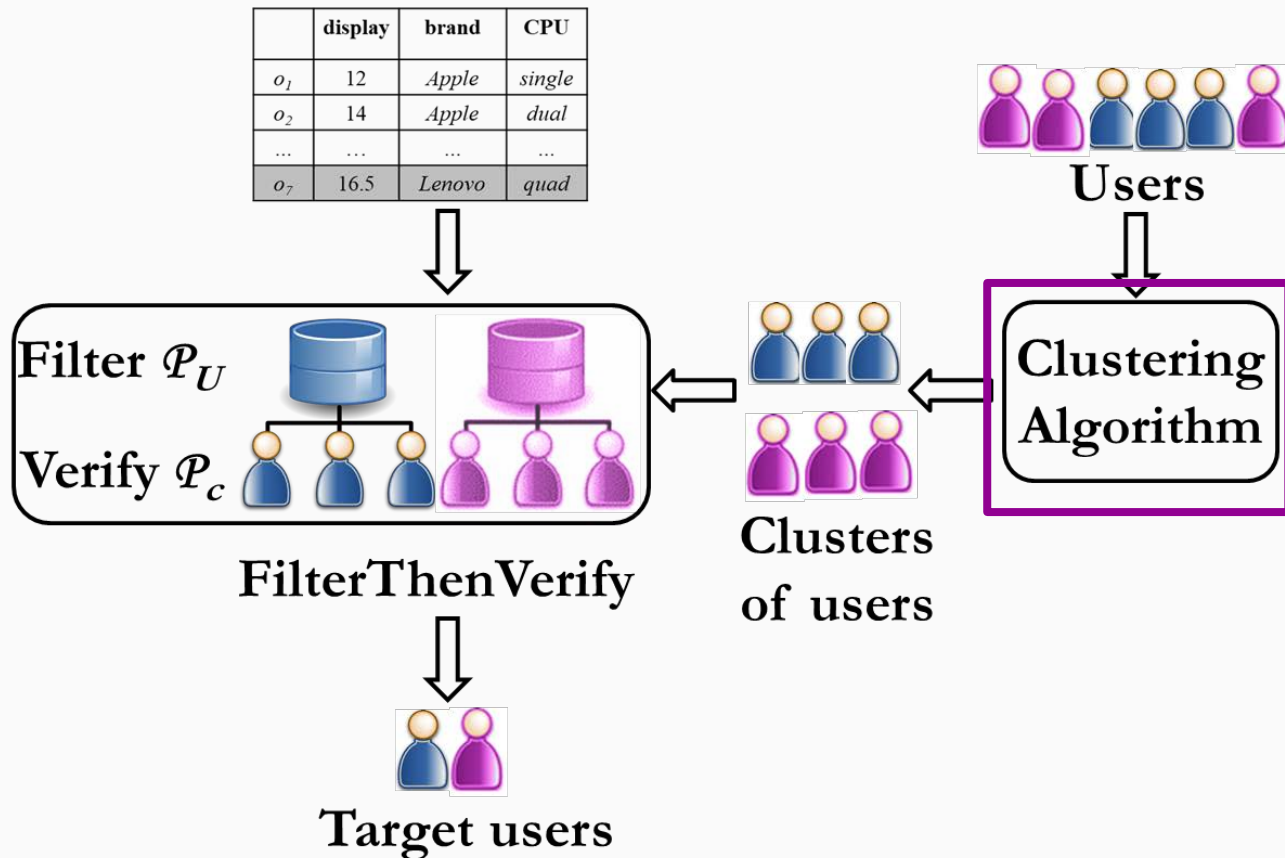
	display	brand	CPU
$o_1$	12	Apple	single
$o_2$	14	Apple	dual
$o_3$	15	Samsung	dual
$o_4$	19	Toshiba	quad
$o_5$	9	Samsung	quad
$o_6$	9.5	Lenovo	triple
$o_7$	16.5	Lenovo	quad

# FilterThenVerify

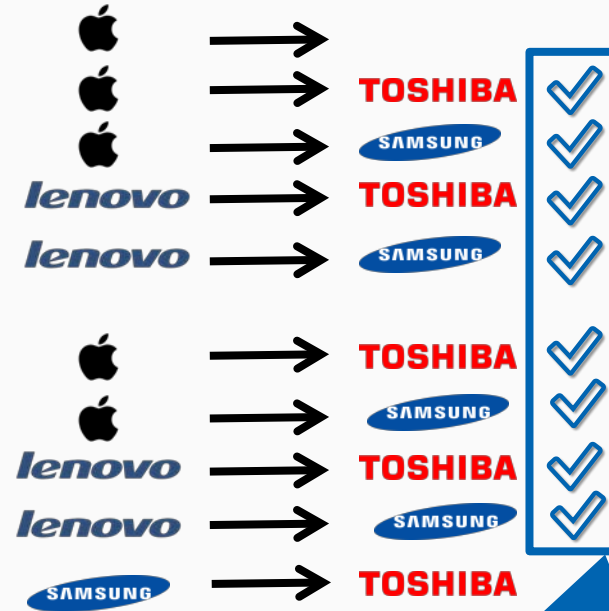
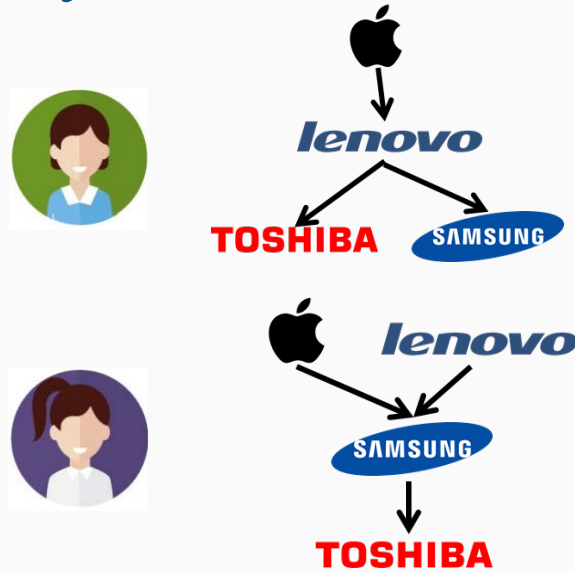


	display	brand	CPU
$o_1$	12	Apple	single
$o_2$	14	Apple	dual
$o_3$	15	Samsung	dual
$o_4$	19	Toshiba	quad
$o_5$	9	Samsung	quad
$o_6$	9.5	Lenovo	triple
$o_7$	16.5	Lenovo	quad

# System Architecture



# Similarity Function

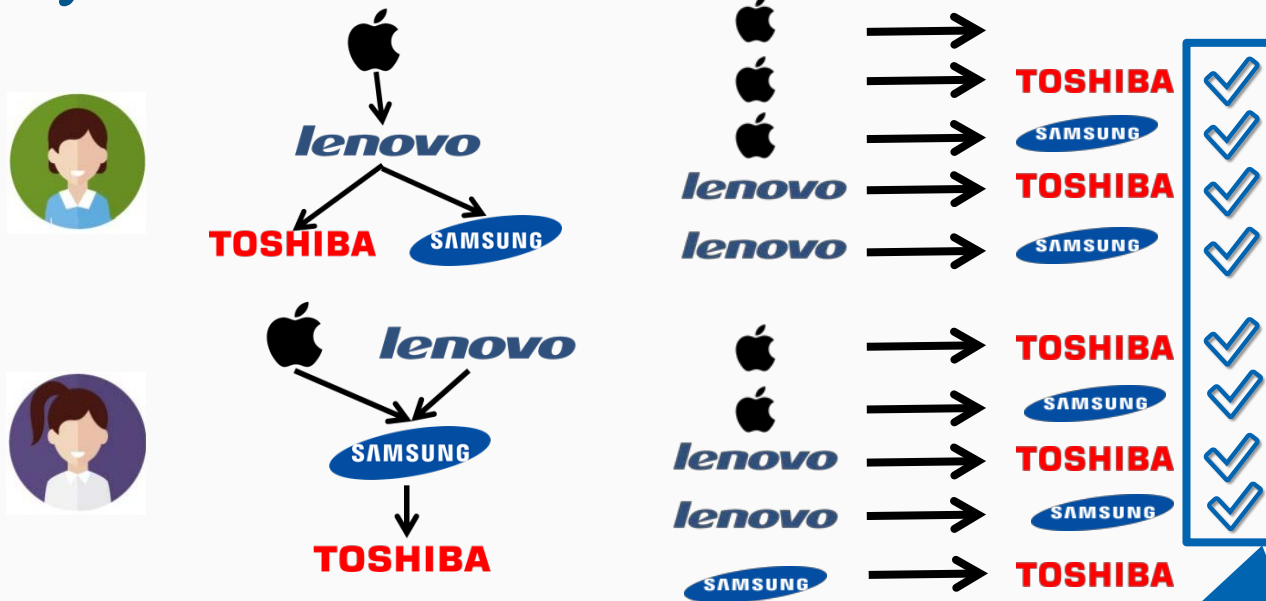


□ Jaccard similarity

$$\frac{|Common\ preference\ tuples|}{|All\ preference\ tuples|}$$

Common preference tuples

# Similarity Function



□ Weighted Jaccard similarity

▪ Locations of preference tuples

Common preference tuples

# Approx. Common Preference Tuples

- Preferences can be diverse
  - Tiny clusters

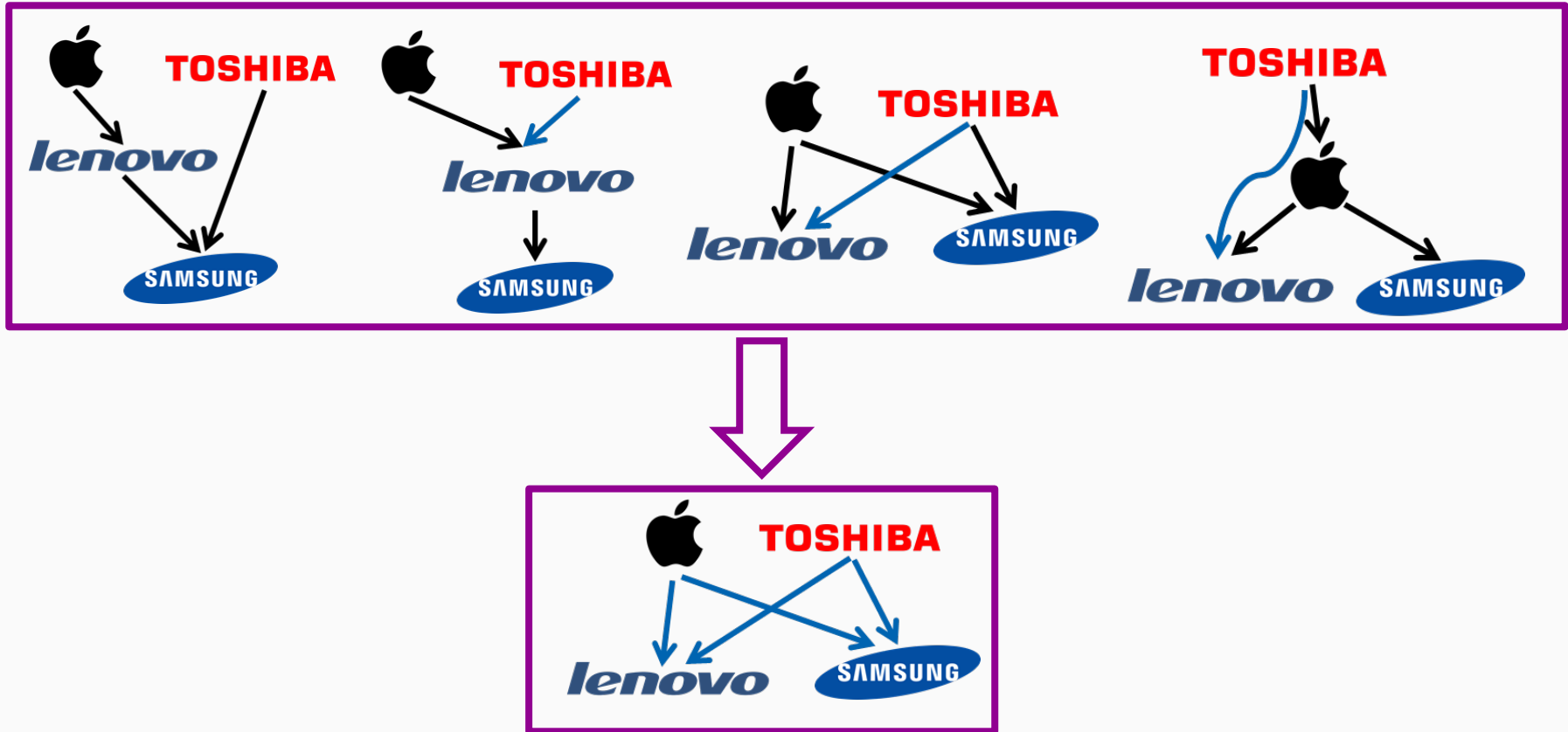


# Approx. Common Preference Tuples

- Preferences can be diverse
  - Tiny clusters
- Relax idea of common preference tuple
  - ✓ Preference polling

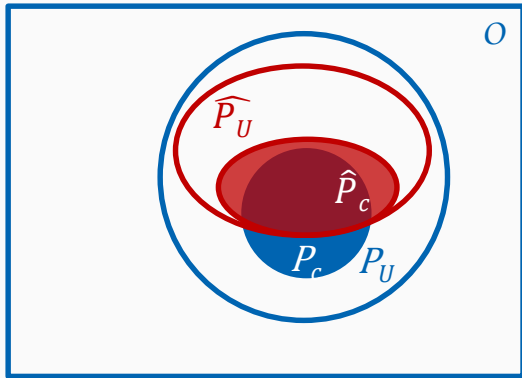


# GetApproxCommonPreferenceTuples

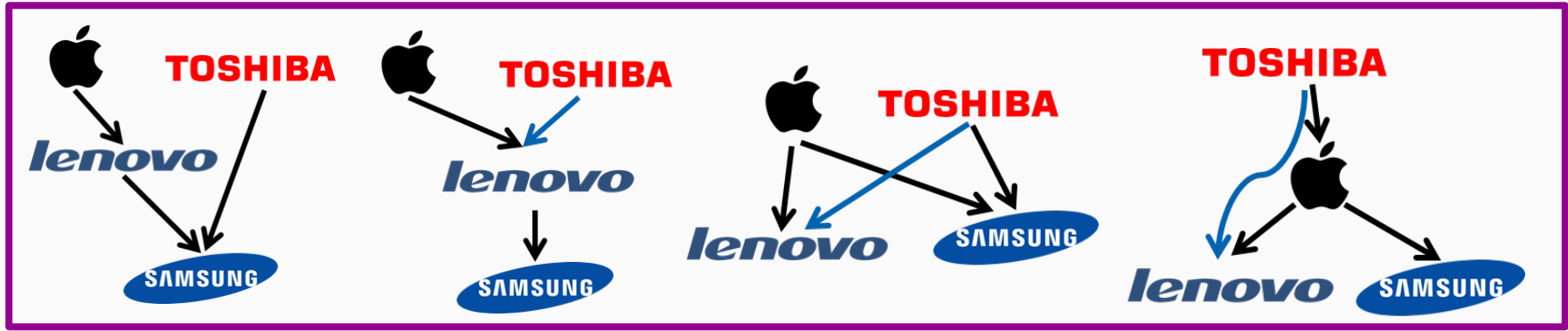


# Properties of Approx. Common Preference Tuples

- Pareto frontier w.r.t. approx. common preference tuples:  $\widehat{P}_U$
- Pareto frontier w.r.t. user upon approximation:  $\widehat{P}_c$
- Lemma 2:
  - Approx. common preference tuples  $\supseteq$  Common preference tuples
- Theorem 2
  - $\widehat{P}_U \subseteq P_U$
- Lemma 3
  - $\widehat{P}_U \supseteq \widehat{P}_c$
- Theorem 3
  - $\widehat{P}_U \cap P_c \subseteq \widehat{P}_c$



# Similarity Function



□ Percentage of preference tuples

# Related Works

- Conventional preference query (Kießling VLDB 2002)
  - Pareto frontier w.r.t. individual users, **separately**
- ✓ Our solution---
  - Share computation across **multiple** users

# Related Works

## ➤ Mining favorable facets (Wong et al. SIGKDD 2007)

### ■ Minimum disqualifying condition

	<b>brand</b>	<b>CPU</b>
$o_1$	<i>Apple</i>	<i>single</i>
$o_2$	<i>Samsung</i>	<i>dual</i>
$o_3$	<i>Toshiba</i>	<i>quad</i>



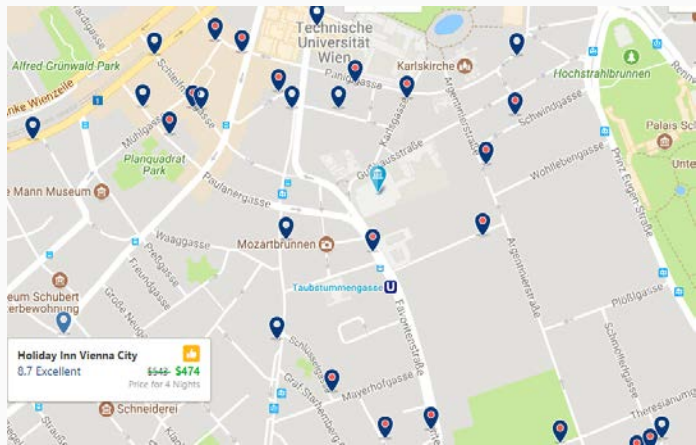
	<b>Minimum set of preferences to disqualify</b>
$o_1$	$((\text{Samsung}, \text{Apple}) \wedge (\text{dual}, \text{single})) \vee ((\text{Toshiba}, \text{Apple}) \wedge (\text{quad}, \text{single}))$
$o_2$	$((\text{Apple}, \text{Samsung}) \wedge (\text{single}, \text{dual})) \vee ((\text{Toshiba}, \text{Samsung}) \wedge (\text{quad}, \text{dual}))$
$o_3$	$((\text{Apple}, \text{Toshiba}) \wedge (\text{single}, \text{quad})) \vee ((\text{Samsung}, \text{Toshiba}) \wedge (\text{dual}, \text{quad}))$

## ✓ Our solution---

### ■ Compatible with continuously arriving objects

# Related Works

	Attribute	Order
<b>Reverse skyline query (Dellis et al. VLDB 2007)</b>	Numerical: price, distance	Total
<b>Our solution</b>	Categorical/numerical: brand, hotel/suite	Partial



# Experiment by Simulation

## □ Movie Dataset

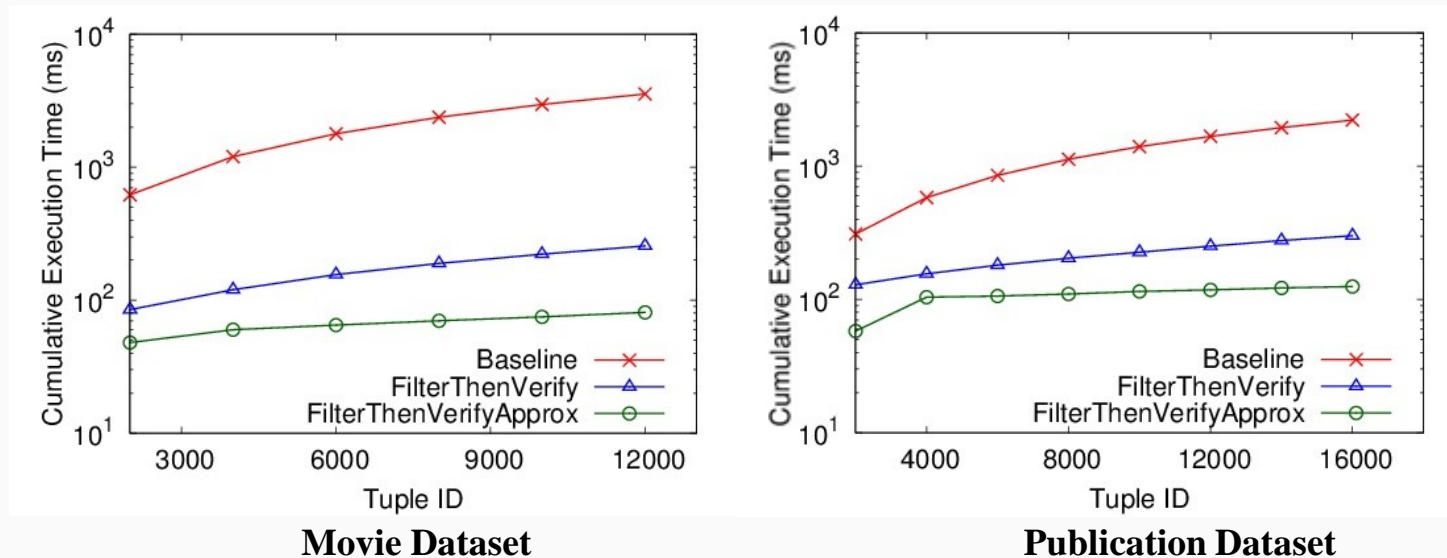
- 12,749 movies: joined Netflix dataset with data from IMDB
- 1000 users
- 4 attributes: *actor, director, genre, writer*

## □ Publication Dataset

- 17,598 publications: ACM Digital Library
- 1000 users
- 4 attributes: *affiliation, author, conference, and keyword*



# Performance of FilterThenVerify/FilterThenVerifyApprox



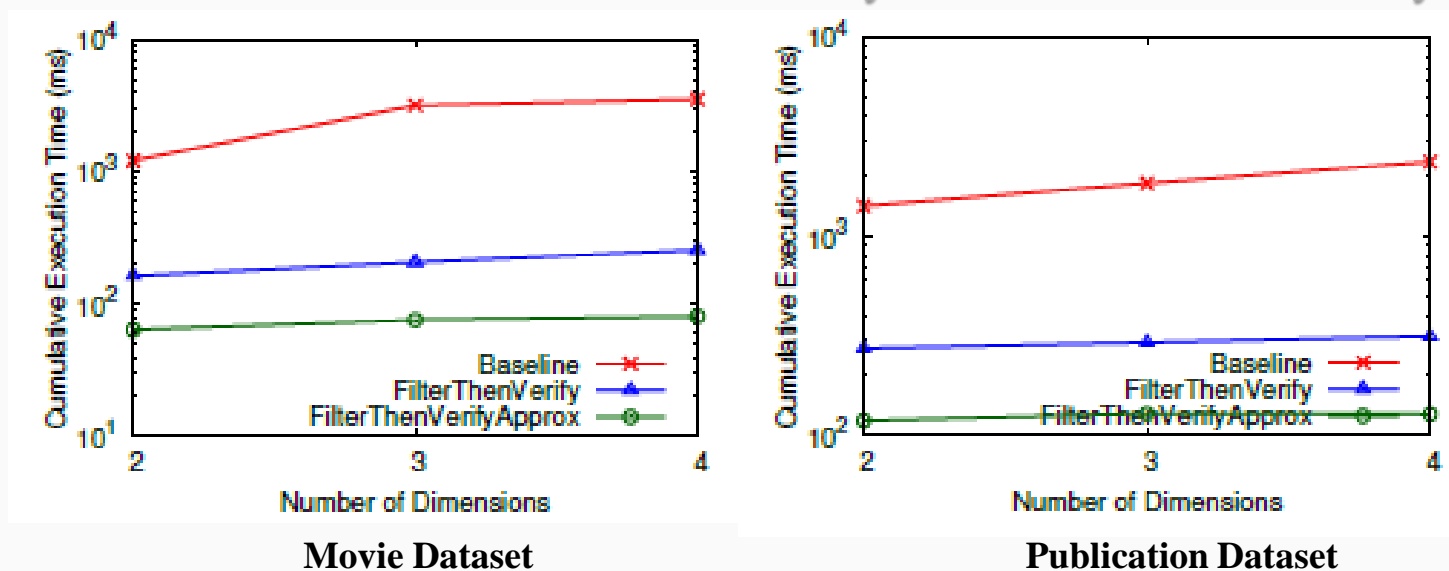
□ Baseline < FilterThenVerify/FilterThenVerifyApprox

- Fewer comparisons due to filtering

□ FilterThenVerify < FilterThenVerifyApprox

- Approx. allows more sharing

# Performance of FilterThenVerify/FilterThenVerifyApprox



□ Execution time increases with  $d$

- High  $d \Rightarrow$  large Pareto frontiers  $\Rightarrow$  more comparisons

# Efficacy of FilterThenVerifyApprox

Dataset	$h = 0.70$			$h = 0.55$		
	Precision	Recall	F-measure	Precision	Recall	F-measure
Movie	100	95.43	97.67	99.99	90.46	94.99
Publication	100	96.59	98.27	100	95.13	97.51

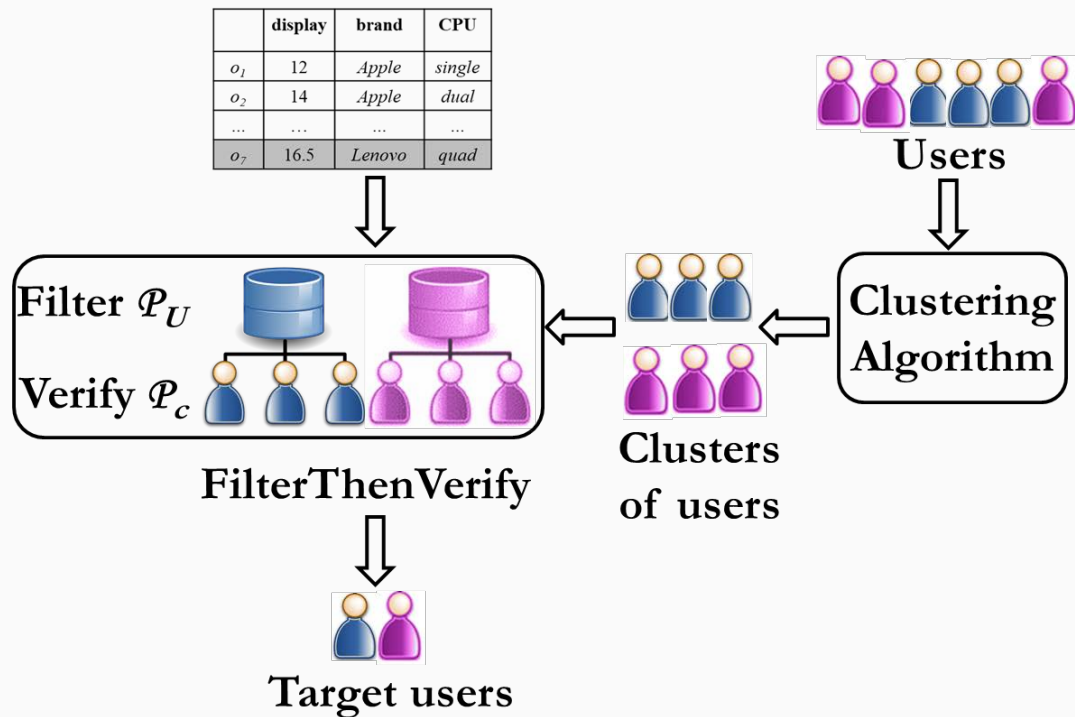
□ Recall decreases with  $h$

- Small  $h \Rightarrow$  large clusters  $\Rightarrow$  high false negatives

□ Stable precision

- Few false negatives  $\Rightarrow$  fewer false positives

# Conclusion



- ✓ Efficient algorithm to find target users
- ✓ Novel problem of clustering partial orders

*THANK YOU!*