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

Afroza Sultana and Chengkai Li
The University of Texas at Arlington

IDiR idir.uta.edu   UNIVERSITY OF TEXAS ARLINGTON

Motivation

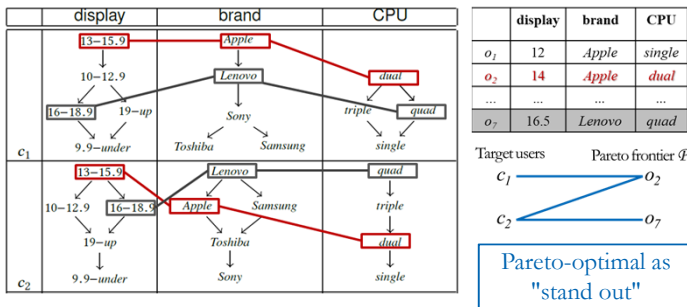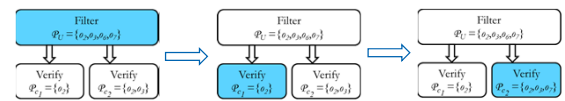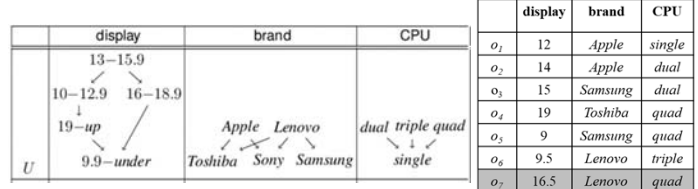| Recommendation based on users' preferences | Preferences with multiple attributes | Objects that "stand out" |

Genre
Author
Publisher

Meet your next favorite book.

An Example

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

Target users   Pareto frontier $\mathcal{P}_c$

c_1 ———————— o_2

c_2 ———————— o_7

Pareto-optimal as "stand out"

Problem Formulation

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

Find target users such that o_7 is in Pareto frontier. ⟶ c_2

Challenges & Ideas

- Exhaustive comparisons
  - For every user
  - With every Pareto-optimal object
- A number of users
- Objects streaming
- Efficient dissemination
- ✓ Sharing computation across users

Common preference tuples

- 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%

O
$\mathcal{P}_U$
$\mathcal{P}_c$

- Which users share preferences?
  - ✓ Cluster users based on preferences
- No prior study on clustering for partial orders
  - ✓ Define similarity functions w.r.t. partial orders

System Architecture

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

Users
Clustering Algorithm
Clusters of users

Filter $\mathcal{P}_U$
Verify $\mathcal{P}_c$
FilterThenVerify

Target users

Algorithm FilterThenVerify

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

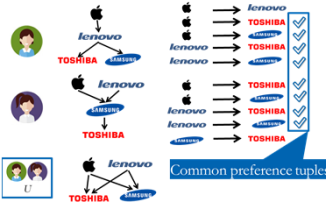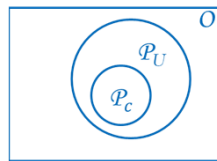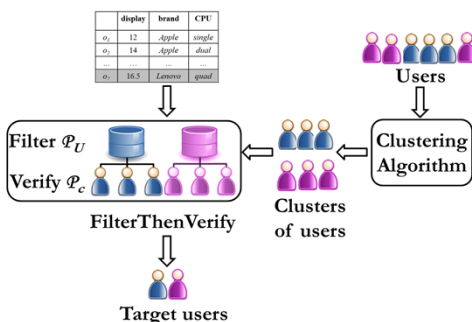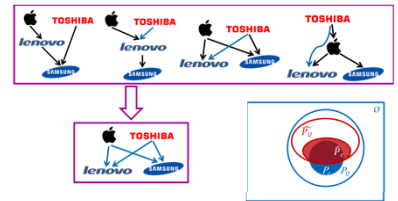|	display	brand	CPU
o_1	12	Apple	single
o_2	14	Apple	dual
o_3	15	Samsung	dual
o_4	19	Toshiba	quad
o_5	9	Samsung	quad
o_6	9.5	Lenovo	triple
o_7	16.5	Lenovo	quad

Similarity Functions

- Jaccard similarity
  - |Common preference tuples| / |All preference tuples|
- Weighted Jaccard similarity
  - Values near top have more impact
  - Maximal values: no other value is preferred over
  - Weighted preference tuples

Approx. Common Preference Tuples

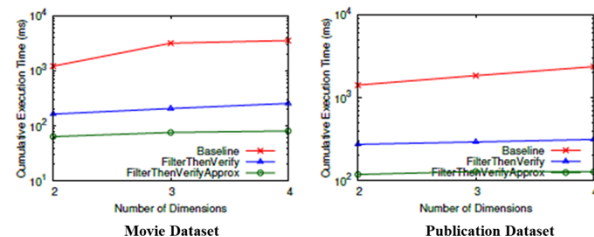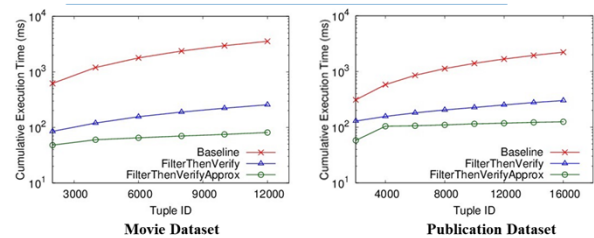| Preferences can be diverse |
|---|
| - Tiny clusters |

Relax idea of common preference tuple
✓ Preference polling

Experimental Evaluation

- Movie Dataset: 12,749 movies: joined Netflix dataset with data from IMDB
- Publication Dataset: 17,598 publications: ACM Digital Library

Movie Dataset   **Publication Dataset**

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