Week 8: Knowledge Based RSs – Part II

CSX4207/ITX4207: Decision Support and Recommender Systems

Asst. Prof. Dr. Rachsuda Setthawong

Objectives

- To be familiar with a widely used knowledge-based recommendation algorithms
- To present and discuss the applications of collaborative filtering approaches

Outlines

- Conjoint Analysis
- Knowledge based recommendation algorithm
  - Critiquing
- Pros and Cons of Knowledge based RS
- Articles' Presentation and Discussion

Basic Conjoint Analysis

- A statistical technique that marketers usually apply to measure *customer satisfaction* on available products to determine *desirable* product's *features* and *price*
- Common steps:
  1. Marketers **prepare choices** of product **as a set of cards**
  2. User(s) **ranks** the cards **wrt their preferences**
  3. Based on the ranks in step 2, **calculate a utility function** (representing user's preference) and/or relative importance (of available features)

Ref: Joseph Curry, **Sawtooth Software *RESEARCH PAPER SERIES:*** Understanding Conjoint Analysis in 15 Minutes, Sawtooth Technologies, Inc., 1996
https://sawtoothsoftware.com/resources/technical-papers/understanding-conjoint-analysis-in-15-minutes

An Example

- To market a new golf ball with three important product *features*:
  - Average driving distance
  - Average ball life
  - Price
- **Goal:** What is the desirable features' value for a new golf ball?

Ref: Joseph Curry, **Sawtooth Software *RESEARCH PAPER SERIES:*** Understanding Conjoint Analysis in 15 Minutes, Sawtooth Technologies, Inc., 1996

An Example

1. Marketers **prepare choices** of product **as a set of cards**
   - The no. of choices = 3 x 3 x 3 = 27. In this example will simplify the step.

Average Driving Distance	*Average Ball Life*	*Price*
275 yards	54 holes	$1.25
250 yards	36 holes	$1.50
225 yards	18 holes	$1.75

Ref: Joseph Curry, **Sawtooth Software *RESEARCH PAPER SERIES:*** Understanding Conjoint Analysis in 15 Minutes, Sawtooth Technologies, Inc., 1996

Asst. Prof. Dr. Rachsuda Setthawong

An Example

2. User(s) **ranks** the choices (the cards) **wrt their preferences**

Figure 2a

Buyer 1

		Average Ball Life		
		54 holes	**36 holes**	**18 holes**
Average Driving Distance	**275 yards**	1	2	4
	250 yards	3	5	6
	225 yards	7	8	9

Figure 4a

Buyer 1

		Average Ball Life		
		54 holes	**36 holes**	**18 holes**
Price	**$1.25**	1	4	7
	$1.50	2	5	8
	$1.75	3	6	9

Ref: Joseph Curry, **Sawtooth Software** ***RESEARCH PAPER SERIES:*** Understanding Conjoint Analysis in 15 Minutes, Sawtooth Technologies, Inc., 1996

Asst. Prof. Dr. Rachsuda Setthawong

An Example

3. Use user's ranks to generate the utility values of each attribute (could also be derived from a linear regress model)

From Figure 2a to **Figure 3**

Buyer 1

		Average Ball Life		
		54 holes **50**	**36 holes** **25**	**18 holes** **0**
Average Driving Distance	**275 yards** **100**	sum (1) 150	(2) 125	(4) 100
	250 yards **60**	(3) 110	(5) 85	(6) 60
	225 yards **0**	(7) 50	(8) 25	(9) 0

From Figure 4a to **Figure 4b**

Buyer 1

		Average Ball Life		
		54 holes **50**	**36 holes** **25**	**18 holes** **0**
Price	**$1.25** **20**	(1) 70	(4) 45	(7) 20
	$1.50 **5**	(2) 55	(5) 30	(8) 5
	$1.75 **0**	(3) 50	(6) 25	(9) 0

Find a set of values for Average Driving Distance and a second set for Average Ball Life for Buyer 1 so that when adding these values together for each ball, they reproduce Buyer 1's rank orders

Figure 5

Average Driving Distance		*Average Ball Life*		*Price*	
275 yards	100	54 holes	50	$1.25	20
250 yards	60	36 holes	25	$1.50	5
225 yards	0	18 holes	0	$1.75	0

utility values

Ref: Joseph Curry, **Sawtooth Software *RESEARCH PAPER SERIES:*** Understanding Conjoint Analysis in 15 Minutes, Sawtooth Technologies, Inc., 1996

An Example

- Apply the obtained utility values* to calculate utility values for the user.

Figure 5*

Average Driving Distance		*Average Ball Life*		*Price*	
275 yards	100	54 holes	50	$1.25	20
250 yards	60	36 holes	25	$1.50	5
225 yards	0	18 holes	0	$1.75	0

Figure 7

	Choice 1		Choice 2	
Buyer 1	*Distance Ball*		*Long-Life Ball*	
Distance	275	100	250	60
Life	18	0	54	50
Price	$1.50	5	$1.75	0
Total Utility		105	<	110

Buyer 1 prefers choice 2 more than choice 1.

*: Could also be derived from a ***linear regression model***

S = B0 + B11(AD1) + B12(AD2) + B13(AD3) + B21(AB1) + B22(AB2) + B23(AB3) + B31(P1) + B32(P2) + B33(P3)

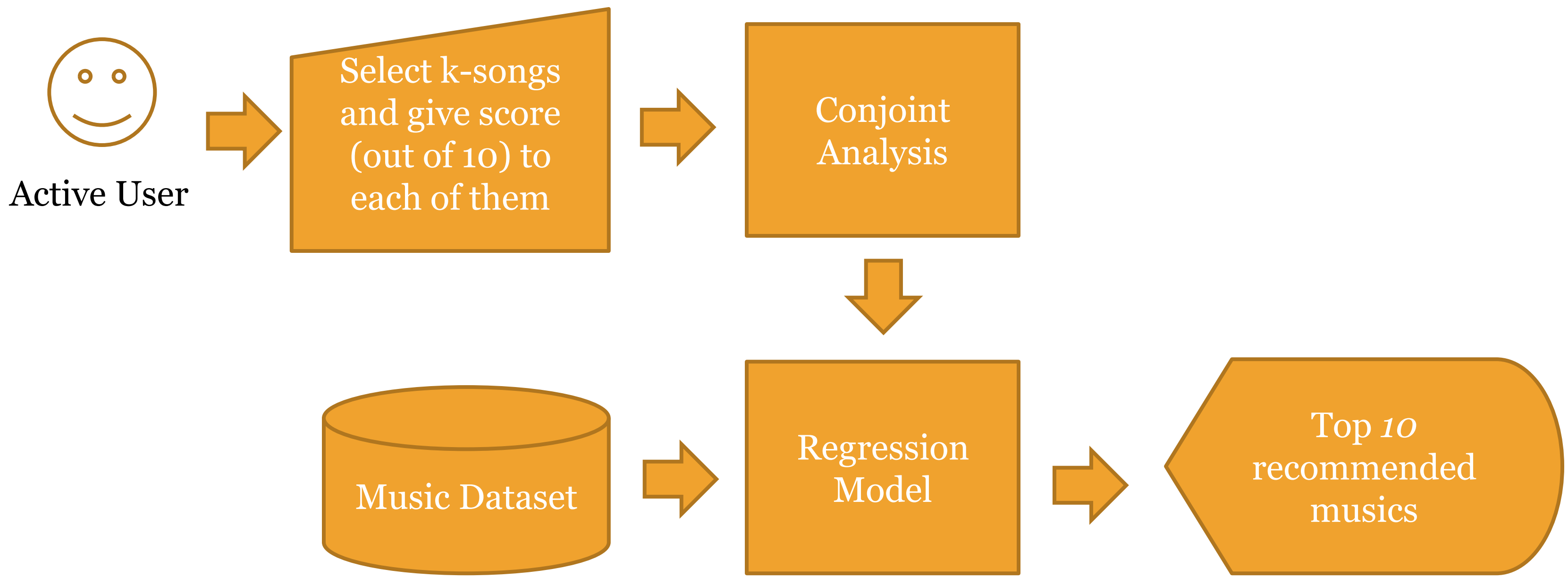
S = 9.08 -4.75(AB1) – 2.17(AB2) – 4.44(AD1) – 2.5(AD2) – 2.78(P1) – 1.78(P2)

Ref: Joseph Curry, **Sawtooth Software *RESEARCH PAPER SERIES:*** Understanding Conjoint Analysis in 15 Minutes, Sawtooth Technologies, Inc., 1996

Asst. Prof. Dr. Rachsuda Setthawong

Applying Conjoint Analysis in RS – 1/3

Example 1: Music Recommender System using Conjoint Analysis



Ref: https://medium.com/@pum.crab/music-recommender-system-using-conjoint-analysis-f20fdc9cd8ec

Asst. Prof. Dr. Rachsuda Setthawong

Applying Conjoint Analysis in RS – ½

Example 2: analyze **ranks** of variables that affects user's decision

- Given
  - Ranges of **price**: $price_1$[100-159], $price_2$[160-199], $price_3$[200-300]
  - Ranges of **mpix**: $mpix_1$[5.0-8.0], $mpix_2$[8.1-11.0]
- Then, a target user **ranks** of all possible combination of price's and mpix's choices

	$price_1$ [100-159]	**$price_2$ [160-199]**	**$price_3$ [200-300]**
$mpix_1$ [5.0-8.0]	4	5	6
$mpix_2$ [8.1-11.0]	2	1	3

Applying Conjoint Analysis in RS – 2/2

Table1	$price_1$ [100-159]	$price_2$ [160-199]	$price_3$ [200-300]	avg_rank($mpix_x$)
$mpix_1$ [5.0-8.0]	4	5	6	5
$mpix_2$ [8.1-11.0]	2	1	3	2
avg_rank($price_x$)	3	3	4.5	**3.5**

Table2	d = avg(ranking) – avg_rank($attr_x$)
avg(ranking) – avg_ranking($mpix_1$) (3.5 - 5)	-1.5
avg(ranking) – avg_ranking ($mpix_2$) (3.5 - 2)	1.5
avg(ranking) – avg_ranking ($price_1$) (3.5 - 3)	0.5
avg(ranking) – avg_ranking ($price_2$) (3.5 - 3)	0.5
avg(ranking) – avg_ranking ($price_3$) (3.5 – 4.5)	-1.0

The **relative importance** of each attribute is calculated by considering *how much difference* each attribute could make in the total utility of a product (the **range** in the **attribute's utility values**.)

avg($mpix_x$) **span**
$= |max\_d_{mpix} - min\_d_{mpix}|$
$= |1.5 - (-1.5)| = 3$

avg($price_x$) **span**
$= |max\_d_{price} - min\_d_{price}|$
$= |0.5 - (-1.0)| = 1.5$

Conclusion: Mpix changes **<u>has a higher effect</u>** on the overall utility (for this customer) than price changes.

Asst. Prof. Dr. Rachsuda Setthawong

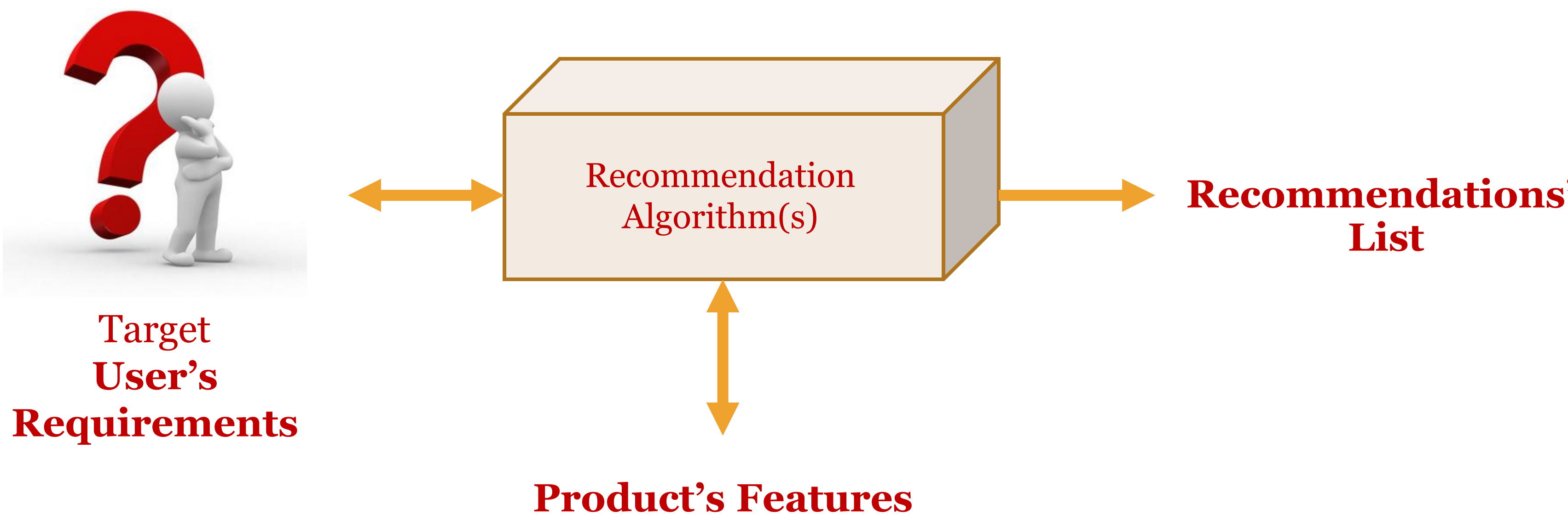
Revising Knowledge-based Recommendation

Main Idea

- To match *user requirements* with item's features.
- To use *user interaction* to refine recommendations.
- To predict which items the current user will most probably like.

Asst. Prof. Dr. Rachsuda Setthawong

How to Generate Recommendation Using Knowledge Based Filtering Approach



Knowledge based: "Tell me what fits based on my needs."

Asst. Prof. Dr. Rachsuda Setthawong

Outlines

- Conjoint Analysis
- Knowledge-based recommendation algorithm
  - Critiquing
- Pros and Cons of Knowledge based RS
- Articles' Presentation and Discussion

Asst. Prof. Dr. Rachsuda Setthawong

Basic Types of Knowledge-based Recommender Systems

- Constraint-based
- Case-based

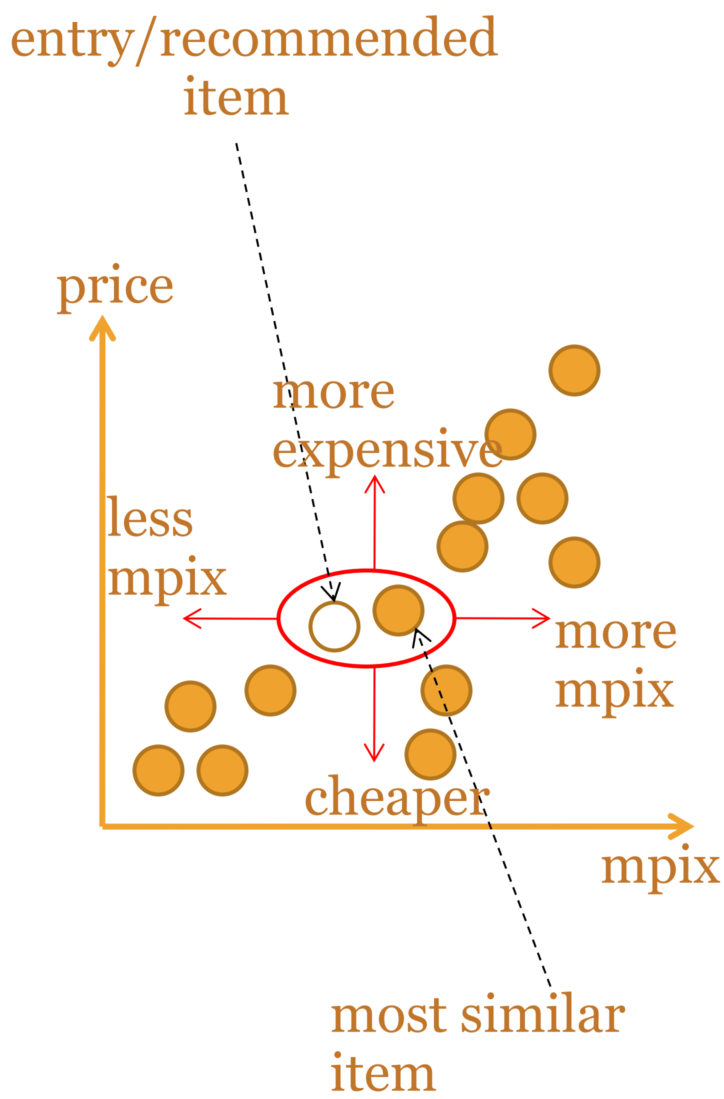
Interacting with Case-based Recommenders

- Motivation: a pure query-based approach requires
  - Respecifying user's requirements until a target item has been identified
  - A substantial domain knowledge to perform well
- Alternatives: browsing-based approaches to item retrieval
  - By *navigating* in the item space with the goal to *find useful alternatives* (initially unknown)
  - E.g., Critequing

Asst. Prof. Dr. Rachsuda Setthawong

Critiquing

- Idea: users specify their ***changes*** requests in the form of goals (critiques) for the currently unsatisfied items recommended.
  - E.g. of critiques (in the level of *technical properties*)
    - `cheaper` (on `price` feature)
    - `more mpix` (on `mpix` feature)
  - E.g. of critiques (in the level of *abstract dimensions*)
    - The apartment should be *more modern-looking*.
    - The hotel location should be *nearer to the sea*.



Benefit and Major Goals of Critiquing based Recommender Systems

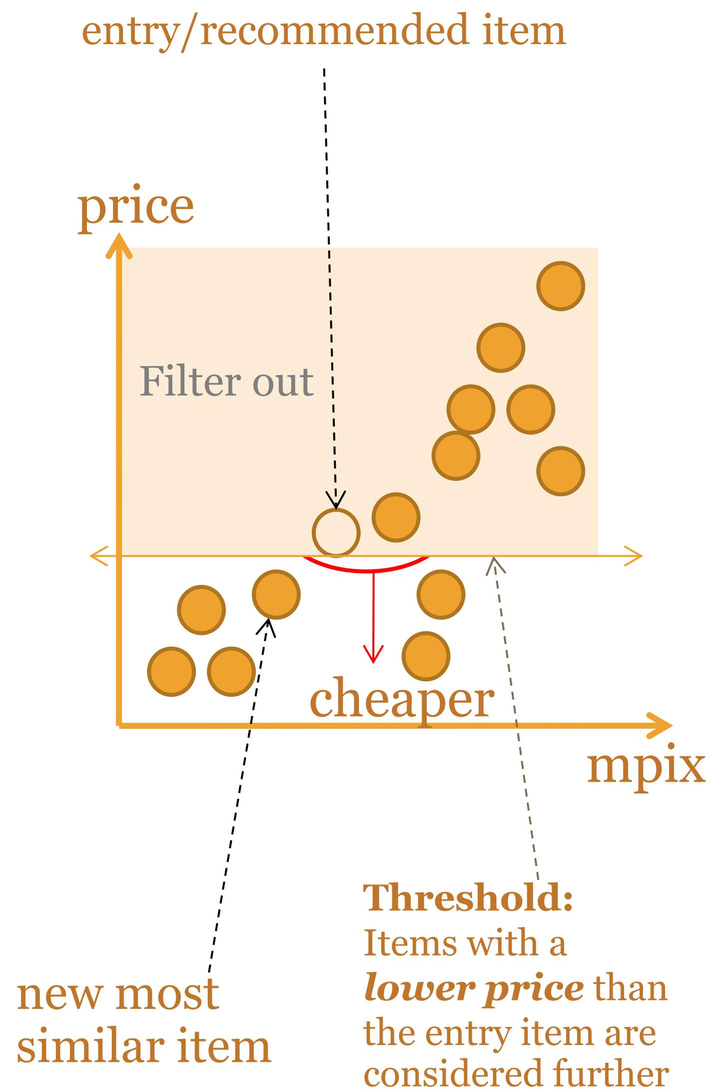
- Benefit: Allow users to ***easily articulate*** preferences instead of *specifying concrete* values for item properties.

- Major Goals:
  - Saving item selection time
  - Obtaining the same recommendation quality as standard query-based approaches

Asst. Prof. Dr. Rachsuda Setthawong

Item Recommendation

- Apply the algorithm SIMPLECRITIQUING (slide 21) to obtain entry item (recommended items)
- **Idea:** Iterate the two tasks until Critique is null.
  - ITEMRECOMMENDED()
    - Select an item r to be presented to the user
      - First round: entry item r = the ***most similar*** item between the **user's query q** and the candidate items
      - Next rounds: recommended item r = the ***most similar*** item between **currently recommended item** and the candidate items
  - USERREVIEW()
    - ***Review*** the recommended (entry) item and ***either accept it or selects another critique*** (modify q).



Asst. Prof. Dr. Rachsuda Setthawong

SimpleCritiquing(q, CI)

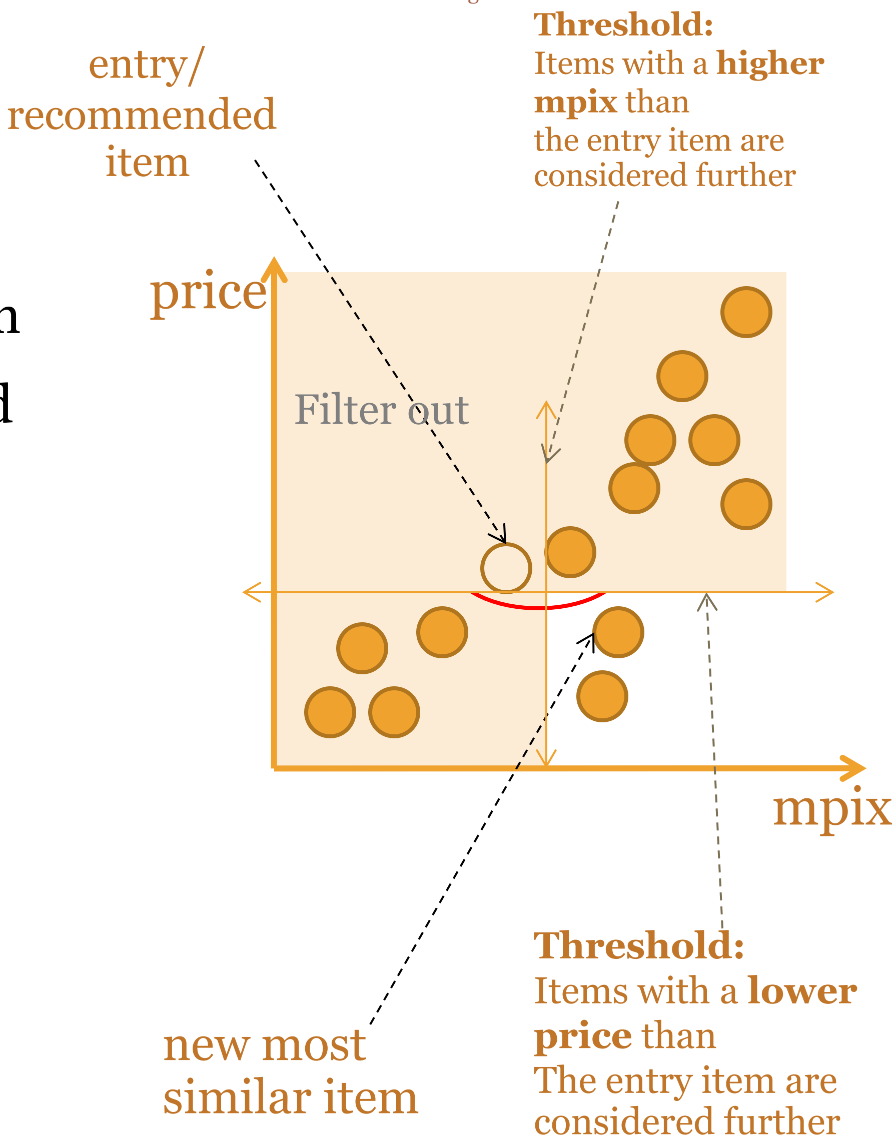
```
Input: Initial user query q; Candidate items CI
procedure SimpleCriquing(q, CI)
 repeat
  r ← ItemRecommend(q, CI);
  q ← UserReview(r, CI);
 until empty(q)
end procedure

procedure ItemRecommend(q, CI)
 CI ← {ci ∈CI: satisfies(ci, q)};
 r ← mostsimilar(CI, q);
 return r;
end procedure

procedure UserReview(r, CI)
 q ← critique(r);
 CI ← CI – r;
 return q;
end procedure
```

Asst. Prof. Dr. Rachsuda Setthawong

Unit Critiques vs Compound Critiques

- Unit critiques allows the definition of change requests that are related to **a single item property.**

- Compound critiques allows the definition of change requests that are related to **multiple item properties.**
  - E.g., `cheaper` and `more mpix`



Asst. Prof. Dr. Rachsuda Setthawong

An Advantage and a Disadvantage of Compound Critiques

- Advantage: provide a faster progression through the item space.
- Disadvantage: must formulate critiques statically.
  - All critique alternatives are available beforehand.
  - E.g.,

User requirements	(Static) critiques
fastest CPU available on the market and *maximum available storage capacity*	*faster CPU* and *more storage capacity* *(more efficient)*

Asst. Prof. Dr. Rachsuda Setthawong

Dynamic Critiquing

Pattern = ***Generic descriptions*** of differences between the recommended (entry) item and the candidate items

- Exploits patterns.
- Use the patterns in deriving compound critiques on the fly (in each critiquing cycle).
  - Use the concept of association rule mining in calculating dynamic critiques
- Apply the algorithm DYNAMICCRITIQUING (slide 26) to obtain entry item (recommended items).

Item Recommendation (DYNAMICCRITIQUING())

- ITEMRECOMMEND()
  - Select an item r to be presented to the user
- COMPOUNDCRITIQUES()
  - Based on r, identify critique patterns (CRITIQUEPATTERNS())
  - Calculate and select compound critiques $cc_i \in CC$ (APRIORI() and SELECTCRITIQUES())
- USERREVIEW()
  - Display the identified compound critiques in CC to the user.
    - The critiquing will stop if the user does not select any critique.

Asst. Prof. Dr. Rachsuda Setthawong

DynamicCritiquing(q, CI)

Input: Initial user query q; Candidate items CI;
number of compound critiques per cycle k;
minimum support for identified association rules σ_{min}

```
procedure DynamicCriquing(q, CI, k, σ_min)
  repeat
    r ← ItemRecommend(q, CI);
    CC ← CompoundCritiques(r, CI, k, σ_min);
    q ← UserReview(r, CI, CC);
  until empty(q)
end procedure

procedure CompoundCritiques(r, CI, k, σ_min)
  CP ← CritiquePatterns(r, CI); //slide 27
  CC ← Apriori(CP, σ_min); //slide 28
  SC ← SelectCritiques(CC, k); //slide 29
  return SC;
end procedure

procedure ItemRecommend(q, CI)
  CI ← {ci ∈CI: satisfies(ci, q)};
  r ← mostsimilar(CI, q);
  return r;
end procedure

procedure UserReview(r, CI, CC)
  q ← critique(r, CC);
  CI ← CI – r;
  return q;
end procedure
```

Identification of Critique Patterns

- Critiques patterns are a ***generic representation of the differences*** between the currently recommended item (entry item) and the candidate items.

	id	price	mpix	opt-zoom	LCD-size	movies
entry item (EI)	ei_8	278	9.1	9x	3.0	yes
candidate item (CI)	ci_1	148	8.0	4x	2.5	no
	ci_2	182	8.0	5x	2.7	yes
	ci_3	189	8.0	10x	2.5	yes
	ci_4	196	10.0	12x	2.7	yes
	ci_5	151	7.1	3x	3.0	yes
	ci_6	199	9.0	3x	3.0	yes
	ci_7	259	10.0	10x	3.0	yes
critique patterns (CP)	cp_1	$<$	$<$	$<$	$<$	$\neq$
	cp_2	$<$	$<$	$<$	$<$	$=$
	cp_3	$<$	$<$	$>$	$<$	$=$
	cp_4	$<$	$>$	$>$	$<$	$=$
	cp_5	$<$	$<$	$<$	$=$	$=$
	cp_6	$<$	$<$	$<$	$=$	$=$
	cp_7	$<$	$>$	$>$	$=$	$=$

compare

to generate

Asst. Prof. Dr. Rachsuda Setthawong

Mining Compound Critiques from Critique Patterns

- Identify ***compound*** critiques that ***frequently co-occur*** in the set of critique patterns (CP) wrt. Confidence level.
  - Apply APRIORI algorithm
    - Output: A set of association rules: p → q

	id	mpix	opt-zoom
critique patterns (CP)	cp_1	<	<
	cp_2	<	<
	cp_3	<	>
	cp_4	>	>
	cp_5	<	<
	cp_6	<	<
	cp_7	>	>

	id	price	mpix	opt-zoom	LCD-size	movies
entry item (EI)	ei_8	278	9.1	9x	3.0	yes

Examples of association rules (AR) -- *output*	compound critiques (CC)	SUPP	CONF
ar_1: $>_{mpix} \rightarrow >_{zoom}$	cc_1: $>_{mpix(9.1)}, >_{zoom(9x)}$	2/7 = 28.6	2/2 = 100.0
ar_2: $>_{zoom} \rightarrow <_{price}$	cc_2: $>_{zoom(9x)}, <_{price(278)}$	3/7 = 42.9	3/3 = 100.0
ar_3: $=_{movies} \rightarrow <_{price}$	cc_3: $=_{movies(yes)}, <_{price(278)}$	6/7 = 85.7	6/6 = 100.0

Ranking Compound Critiques

- Rank compound critiques according to the ***support values*** of association rules.
  - E.g., Rank of CC: $\{cc_1, cc_2, cc_3\} = cc_3, cc_2, cc_1$

association rules (AR)	compound critiques (CC)	SUPP	CONF
ar_1: $>_{mpix} \rightarrow >_{zoom}$	cc_1: $>_{mpix(9.1)}, >_{zoom(9x)}$	28.6	100.0
ar_2: $>_{zoom} \rightarrow <_{price}$	cc_2: $>_{zoom(9x)}, <_{price(278)}$	42.9	100.0
ar_3: $=_{movies} \rightarrow <_{price}$	cc_3: $=_{movies(yes)}, <_{price(278)}$	85.7	100.0

Outlines

- Conjoint Analysis
- Knowledge-based recommendation algorithm
  - Critiquing
- Pros and Cons of Knowledge based RS
- Articles' Presentation and Discussion

Asst. Prof. Dr. Rachsuda Setthawong

Pros and Cons of Knowledge based RS

Pros	Cons

• Address a cold start problem.
  ▫ Capable of suggesting items to new users.
  ▫ Capable of suggesting new items to users.
• Quick access and easy to manipulate data.
• Capable of refining suggestion.
	• Require details of items to construct item profile.

Outlines

- Conjoint Analysis
- Knowledge-based recommendation algorithm
  - Critiquing
- Advanced item recommendation
- Critique Diversity
- Pros and Cons of Knowledge based RS
- Articles' Presentation and Discussion