COMP9727: Recommender Systems

Lecture 4: Knowledge-Based Recommender Systems

Wayne Wobcke

e-mail:w.wobcke@unsw.edu.au

This Lecture

Focus on recommender "agents" as decision support tools

- Conversational/Critiquing Recommender Systems
  - Consumer Buying Behaviour
  - Adaptive Decision Making
- Data Mining Techniques
  - Association Rule Mining
  - Frequent Pattern Mining
  - Expert Knowledge Acquisition
- User Evaluation
  - User Scenarios
  - Questionnaires

Knowledge-Based Recommender Systems

Useful when users don't know exactly what they want

- Also known as "conversational" recommender systems
- Users cannot articulate their needs as a search query
- Suitable for casual and naive users, for infrequent purchases
- Start with a single example then provide "critiques"
- Helps users navigate a complex search space
- Helps users understand tradeoffs in a complex domain
- Users learn their preferences during search
- Functions as a decision support tool
- Based on consumer buying behaviour models

Satisficing

Simon (1956, 1947) – bounded rationality

- Satisfy and suffice to reach "good enough" decision
- Example: Elimination by aspects in job hiring, Tversky (1972)
  - Order the selection criteria
  - Rank candidates according to first criterion
  - Eliminate those not reaching a threshold
  - Continue with next criterion until one candidate remains
- Thus non-compensatory attributes, not maximal utility

Quite different from maximizing utility

Consumer Buying Behaviour

Extensively studied in marketing

- Screen items to a small "consideration set", Wright & Barbour (1977)
  - Top N items, top N on each attribute, one attribute, $\cdots$
- Make detailed comparison of items in consideration set
- Refine items in consideration set to a "choice set"
- Choose the best item from the choice set, or repeat

Casual users

- No historical user data, no user profile, one time users

Naive users

- Not much domain knowledge, high level information need

Adaptive Decision Maker

Payne, Bettman & Johnson (1993)

- Do not know strategy/preferences/utilities at outset
- Strategy depends on number of choices, complexity
- Trade off accuracy vs effort to choose strategy
- Goal state is refined during decision making process
- Thus constructive, not revealed, preferences

Quiz Question: How did you decide to buy your latest phone/phone plan?

Quiz Question: How many new car models or variants are there?

Information Needs Questions

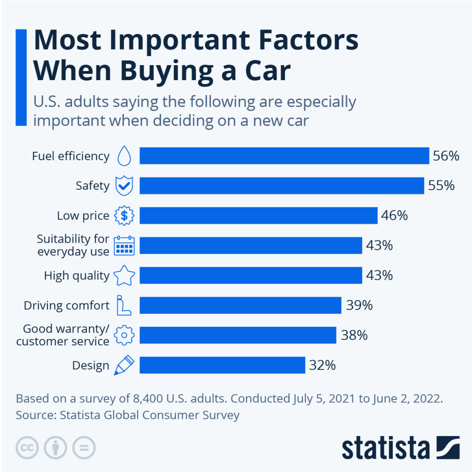
- What sort of information do users need to make a decision?
- When in their system interactions will users want information?
- Do users follow a fixed decision making strategy?
- Do different users have different strategies?
- Can the user's decision strategy be determined from behaviour?
- Does different information affect the user's decision?

Psychological Effects

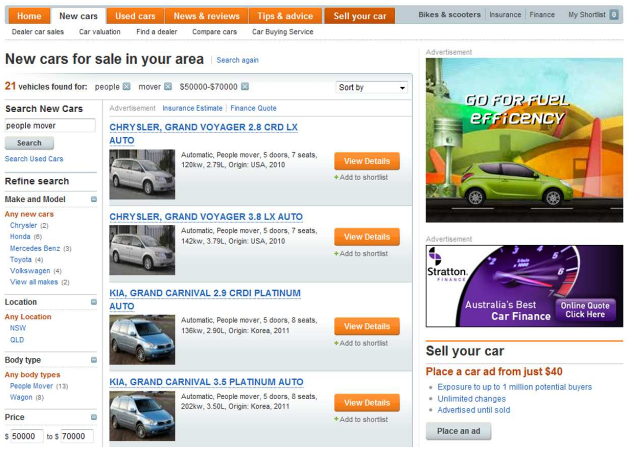
- Context: Additional (decoy) items influence decision
- Anchoring: Item viewed first biases later decisions
- Primacy/recency: Items at beginning and end analysed more
- Framing: Way alternatives are presented influences decision
- Priming: Specific salient properties influence decision
- Defaults: Preset options bias decision making

More work needed on how these apply to recommender systems

Car Attributes Influencing Purchase



Current Car Search



Case-Based Reasoning

- Models of memory organization (scripts) for story “understanding”
  - Departures from the conventional “script” are “interesting”
- Obvious analogy to reasoning about legal cases
  - Start with “similar” precedent(s), ask how current case is different
  - How can precedent be tweaked to be like the current case?

This means “similarity” in the relevant legal sense ...

Critiquing Recommender Systems

- Unit critiques – technical features
- Compound critiques – combinations of features
- Dynamic critiquing – different at different times
- Qualitative critiquing – user-oriented needs-based attributes

Helps user learn tradeoffs in the domain

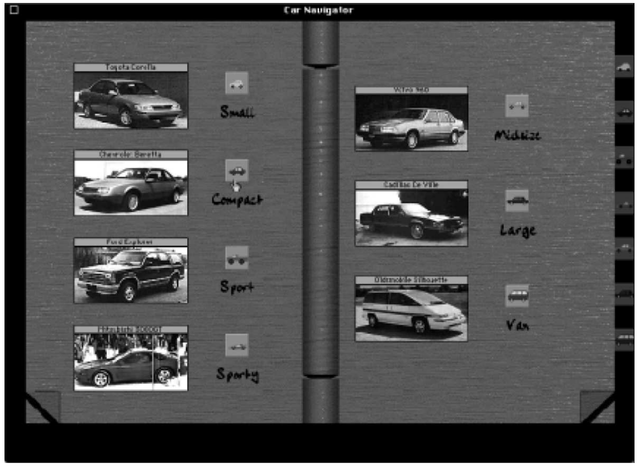
Compound Critiques (Concrete)

Technical features and tradeoffs

Unit Critiques (Qualitative)

Qualitative attributes – but inaccurate

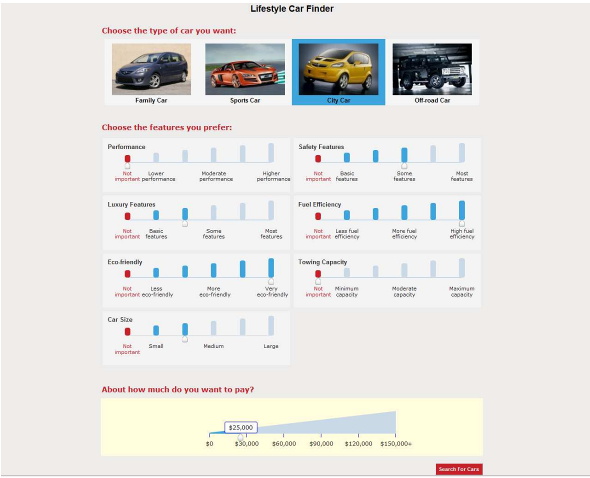
Car Navigator



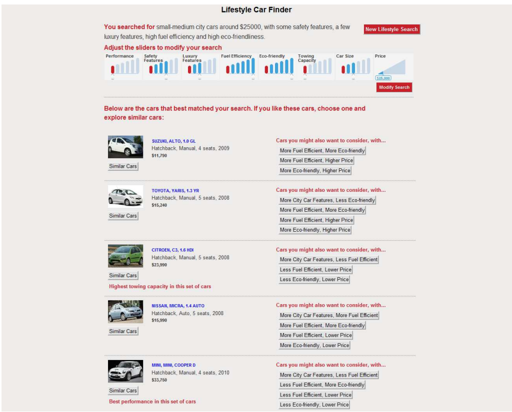
Car Navigator

Qualitative and technical attributes

Lifestyle Search for Cars



Query-Based Qualitative Critiquing



Compound critiques over pairs of qualitative attributes based on query

Association Rule Mining

Given a list of transactions (“market baskets”)

- Itemset: Any set of items
- Frequent Itemset: Itemset occurs in many baskets (fixed threshold)
- Association Rule: If itemset contains I then it also contains J
- Support: Fraction of all transactions containing itemset I
- Confidence: Support($I \cup J$)/Support(I) $\approx P(J|I)$

Note: Rule can have high confidence, but low support

Apriori Algorithm

- If an itemset is not frequent, all its supersets are not frequent
- Start with all itemsets of size 1 ($k = 1$)
- Join all pairs of itemsets with $k-1$ elements in common
  - e.g. $\{1,2,3\}$ join $\{1,2,4\}$ = $\{1,2,3,4\}$
- Prune itemsets that are not frequent (below threshold)
- Repeat until no further itemsets can be generated
- Extract rules with high confidence (meets threshold)

Bottom up: very slow!

Apriori Algorithm

Example: Support = 2/9, Confidence = 0.5

Transactions

T1	A,B
T2	B,D
T3	B,C
T4	A,B,D
T5	A,C
T6	B,C
T7	A,C
T8	A,B,C,E
T9	A,B,C

Step 1

A	6
B	7
C	5
D	2
E	1

Step 2

A,B	4
A,C	4
A,D	1
B,C	4
B,D	2
C,D	0

Step 3

A,B,C	2
A,B,D	1
A,C,D	0
B,C,D	0

Rules

A,B → C	50%
B,C → A	50%
A,C → B	50%

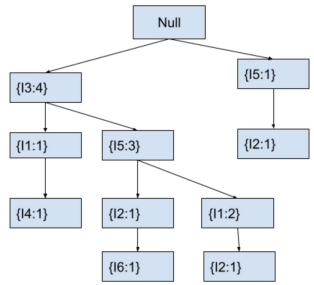
FP-Growth Algorithm

1. Calculate support of each item
2. Sort items in each transaction in descending order of support
3. Create FP-Tree by scanning transactions and adding items in order
4. Create Pattern Base for each item in reverse order of support
   - Paths from the root to the item (excluding both nodes)
5. Create Conditional FP-Tree for each frequent item
   - Consider each pattern base as a distinct set of transactions
6. Generate Frequent Itemsets from Conditional FP-Trees
   - Add original item to each subset of nodes

FP-Growth Algorithm

Transactions

T1	I3, I1, I4
T2	I3, I5, I2, I6
T3	I3, I5, I1, I2
T4	I5, I2
T5	I3, I5, I1



Item	Pattern Base	Conditional FP-Tree	Frequent Itemsets
I2	{I3,I5}:1,{I3,I5,I1}:1,{I5}:1	{I3:2,I5:3}	{I2,I3}:2,{I2,I5}:3,{I2,I3,I5}:2
I1	{I3}:1,{I3,I5}:2	{I3:3,I5:2}	{I1,I3}:3,{I1,I5}:2,{I1,I3,I5}:2
I5	{I3}:3	{I3:3}	{I3,I5}:3

Dynamic Critiquing

- Discovering Compound Critiques
  - Use Apriori on all remaining items to generate compound critiques
  - Works only if entire dataset is small
- Multi-Attribute Utility Theory (MAUT)
  - Suppose utility = $\sum w_i V_i(x_i)$ for weights w_i, value function V_i
  - Preference model = weights, preferred values of item attributes
  - Determine top k items according to user preference model
  - Update user preference model after user selects critique
    - Weights adjusted up or down (factor 2)
    - Values of attributes those of selected critique

Expert Knowledge Acquisition

- Ranking function for each attribute in terms of technical features
  1. Initialize weights of ranking function to random values
  2. Expert: Choose 5 best and 5 worst items
  3. Re-estimate weights – multiple linear regression
  4. Re-order the items according to new ranking function
  5. Sample 10 more items according to distance from median
  6. Expert: Reorder the list in the correct way
  7. Keep going from Step 3 until a satisfactory ranking is achieved

Expert was adamant that linear functions would work!

Interface Design Questions

- How many items should be presented at one time?
- Should items be organized as a list or grouped into clusters?
- How diverse should the set of items be?
- How should items be differentiated from one another?
- What information should be presented with each item?
- How many critiques (per item) should be given to the user?
- Should critiques be different for different items?
- Should the user be allowed to create their own critiques?
- Which critiques better suit the user's requirements at a given time?
- Should critiques concern individual or multiple product attributes?

User Trial

- 9 everyday users in 6 realistic scenarios + free choice
- Assess system usability
  - How easy was the system to use?
  - Can users navigate through the search space?
  - How much use is made of refinement/tradeoffs/similar cars?
  - Are users confident in their decisions?
  - Would they use the system again?

User Study Sample Scenario

You have 1 school-aged child and 1 baby due very soon.

You need to upgrade from the small city car you currently have.

The car you're looking for needs to be:

- safe
- big enough to accommodate family, baby stuff and the school run

Budget: \$20,000–\$40,000, preferably towards the lower end

User Satisfaction

How easy was it for you to figure out how to use ...?	4.2
How easy was it for you to navigate?	4.0
Did the lifestyle car finder work the way you expected it to?	3.6
Was the layout of the information clear and easy to understand?	4.2
Was there enough information about individual cars?	3.0
Was modifying the search using the sliders helpful?	4.0
How helpful was it to refine your search using [critiquing]?	2.9
How relevant and useful were the similar cars?	3.8
How often did you think the [explanations] were useful?	3.7
Were the results the lifestyle car finder gave you relevant and useful?	4.1
Would you be confident enough in the results to further investigate?	3.8
Would you use something like the lifestyle car finder in future?	3.9

Summary

- Suitable for naive and casual users with no prior history
- Useful for exploring complex product spaces
- Suitable for rare, high-risk purchases
- Aim to guide user to preferred subspace of products
- Aim to help user understand domain and construct preferences
- Issue in constructing large-scale knowledge bases
  - Use expert knowledge in well-understood domain
  - Use in other domains (movies), but features limited
  - What if experts disagree?