Recommender Systems: Content-based, Knowledge-based, Hybrid

Radek Pelánek

Today

- lecture, basic principles:
  - **content-based**
  - knowledge-based
  - hybrid, choice of approach, ...
  - critiquing, explanations, ...
- illustrative examples from various domains: videos, recipes, products, finance, restaurants, ...
- highlighting wider context / connections:
  - machine learning, natural language processing, constraint satisfaction problems, automata, ...

Project Discusion

at the end:

- brief presentation of your projects
- application of covered notions to projects
  $\Rightarrow$ *make notes during lecture*

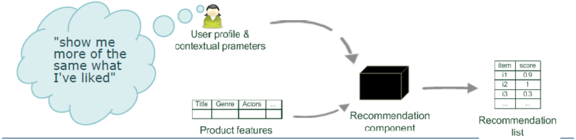
Content-based vs Collaborative Filtering

- collaborative filtering:
  "recommend items that similar users liked"
- content based:
  "recommend items that are similar to those the user liked in the past"

Content-based Recommendations

we need explicit (cf latent factors in CF):

- information about items (e.g., genre, author)
- user profile (preferences)



Recommender Systems: An Introduction (slides)

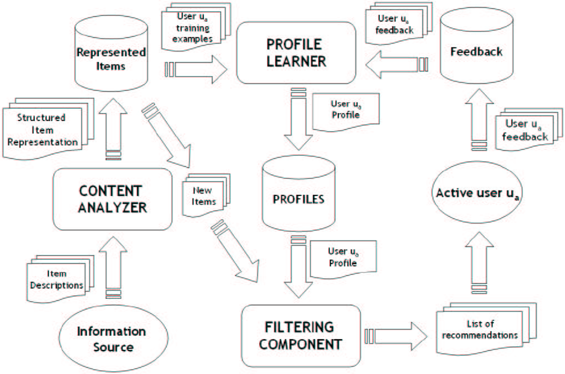
Typical Example: Recipies

- title
- brief description
- list of ingredients
- cooking instructions
- tags (cousine type, dietary restrictions)
- numerical atributes (cooking time, estimated price)

Recipies: Initial Brainstorming

- How to recommend similar items? How to measure similarity of two recipies?
- What could be “user profile”?
- How would we recommend recipies to a given user?

Architecture of a Content-Based Recommender



Handbook of Recommender Systems

Content

Most CB-recommendation techniques were applied to recommending text documents.

- Like web pages or newsgroup messages for example.

Content of items can also be represented as text documents.

- With textual descriptions of their basic characteristics.
- Structured: Each item is described by the same set of attributes

Title	Genre	Author	Type	Price	Keywords
The Night of the Gun	Memoir	David Carr	Paperback	29.90	Press and journalism, drug addiction, personal memoirs, New York
The Lace Reader	Fiction, Mystery	Brunonia Barry	Hardcover	49.90	American contemporary fiction, detective, historical
Into the Fire	Romance, Suspense	Suzanne Brockmann	Hardcover	45.90	American fiction, murder, neo-Nazism

- Unstructured: free-text description.

Recommender Systems: An Introduction (slides)

Content: Multimedia

- manual anotation
  - songs, hundreds of features
  - Pandora, Music Genome Project
  - experts, 20-30 minutes per song
- automatic techniques – signal processing

User Profile

- explicitly specified by user
- automatically learned
  - easier than in CF – features of items are now available

Similarity: Keywords

- general similarity approach based on keywords
- two sets of keywords A, B (description of two items or description of item and user)
- how to measure similarity of A and B?

Similarity: Keywords Example

user preferences: sport, funny, comedy, learning, tricks, skateboard

- video 1: machine learning, education, visualization, math
- video 2: late night, comedy, politics
- video 3: footbal, goal, funny, Messi, trick, fail

Similarity: Keywords

sets of keywords A, B

- **Jaccard coefficient**: $\frac{|A \cap B|}{|A \cup B|}$
- Dice coefficient: $\frac{2 \cdot |A \cap B|}{|A| + |B|}$

many other coefficients available, see e.g. "A Survey of Binary Similarity and Distance Metrics"

Jaccard coefficient/index/score is very simple, but worth knowing under this name; it is used in many different settings

Recommendations by Nearest Neighbors

- k-nearest neighbors (kNN)
- predicting rating for not-yet-seen item i:
  - find k most similar items, already rated
  - predict rating based on these
- good for modeling short-term interest, "follow-up" stories

Similarity: Text Descriptions

Example: similarity of recipes based on the text of instructions

Melt the butter and heat the oil in a skillet over medium-high heat. Season chicken with salt and pepper, and place in the skillet. Brown on both sides. Reduce heat to medium, cover, and continue cooking 15 minutes, or until chicken juices run clear. Set aside and keep warm. Stir cream into the pan, scraping up brown bits. Mix in mustard and tarragon. Cook and stir 5 minutes, or until thickened. Return chicken to skillet to coat with sauce. Drizzle chicken with remaining sauce to serve.

Similarity: Text Descriptions

Examples: product description, recipe instructions, movie plot

basic approach: bag-of-words representation (words + counts of occurrences)

limitations?

Simple Bag-of-words

```
7 and
4 the
4 chicken
4 to
3 heat
3 in
3 skillet
3 with
2 brown
2 minutes
2 or
2 until
2 stir
2 sauce
1 melt
1 butter
```

Term Frequency – Inverse Document Frequency

- disadvantages of simple counts:
  - importance of words ("skillet" vs "with")
  - length of documents
- TF-IDF – standard technique in information retrieval
  - Term Frequency – how often term appears in a particular document (normalized by document length)
  - Inverse Document Frequency – how often term appears in all documents

Term Frequency – Inverse Document Frequency

keyword (term) t, document d

- $TF(t, d) =$ frequency of t in d / maximal frequency of a term in d
- $IDF(t) = \log(N/n_t)$
  - N – number of all documents
  - n_t – number of documents containing t
- $TFIDF(t, d) = TF(t, d) \cdot IDF(t)$

note: there are multiple specific definitions of TF-IDF; they all express the same basic idea, but in different manners

Wikipedia page provides quite good summary: `https://en.wikipedia.org/wiki/Tf-idf`

Similarity

similarity between user and item profiles (or two item profiles):

- vector of keywords and their TF-IDF values
- cosine similarity – angle between vectors
  - $sim(\vec{a}, \vec{b}) = \frac{\vec{a} \cdot \vec{b}}{|\vec{a}||\vec{b}|}$
- (adjusted) cosine similarity
  - normalization by subtracting average values
  - closely related to Pearson correlation coefficient

Improvements

using all words $\rightarrow$ long, sparse vectors

typical processing steps:

- common words, stop words (e.g., "a", "the", "on")
- lemmatization, stemming (e.g., "went" $\rightarrow$ "go", "university" $\rightarrow$ "univers")
- cut-offs (e.g., *n* most informative words)
- phrases (e.g., "United Nations", "New York")

wider context: natural language processing techniques

Limitations of Bag-of-words

- semantic meaning unknown
- example – use of words in negative context

steakhouse description: "there is nothing on the menu that a vegetarian would like..." $\Rightarrow$ *keyword "vegetarian"* $\Rightarrow$ *recommended to vegetarians*

Incorporating Domain Knowledge

user preferences: sport, funny, comedy, learning, tricks, skateboard

- video 1: machine learning, education, visualization, math
- video 2: late night, comedy, politics
- video 3: footbal, goal, funny, Messi, trick, fail

Ontologies, Taxonomies, Folksomies

- ontology – formal definition of entities and their relations
- taxonomy – tree, hierarchy (example: news, sport, football, football world cup)
- folksonomy (folk + taxonomy) – collaborative tagging, tag clouds

Processing Pipeline

input data $\rightarrow$ *[pipeline]* $\rightarrow$ *recommendations*

common pipeline parts:

- data cleaning, tokenization, bag-of-words representation
- lemmatization, stop word removal
- Jaccard, TF-IDF, cosine similarity
- kNN, clustering, dimensionality reduction

Large Language Models

- methods based on neural networks, large language models (LLMs), BERT, GPT, word/sentence embeddings, ...
- potentially powerfull, can capture semantics
- however: “black-box” methods, hard interpretability, pre-trained method may not be suitable for a specific task
- field with intensive developments

my recommendation: gain experience with the interpretable methods and only then start experimenting with LLMs

Recommendation as Classification

- classification problem: features $\rightarrow$ like/dislike (rating)
- use of general machine learning techniques
  - probabilistic methods – Naive Bayes
  - linear classifiers
  - decision trees
  - neural networks
  - ...

wider context: machine learning techniques

Content-Based Recommendations: Advantages

- user independence – does not depend on other users
- new items can be easily incorporated (no cold start)
- transparency – understandable, provides explanations (at least with basic methods)

Content-Based Recommendations: Limitations

- limited content analysis
  - content may not be automatically extractable (multimedia)
  - missing domain knowledge
  - keywords may not be sufficient
- overspecialization – "more of the same", too similar items
- new user – ratings or information about user has to be collected

Content-Based vs Collaborative Filtering

- paper "Recommending new movies: even a few ratings are more valuable than metadata" (context: Netflix)
- our experience in educational domain – difficulty rating (logic puzzle, countries)

Knowledge-based Recommendations

application domains:

- expensive items, not frequently purchased, few ratings (car, house)
- time span important (technological products)
- explicit requirements of user (vacation)

- collaborative filtering unusable – not enought data
- content based – “similarity” not sufficient

Knowledge-based Recommendations

- constraint-based
  - explicitly defined conditions
- case-based
  - similarity to specified requirements

"conversational" recommendations

Constraint-Based Recommmendations – Example

id	price(€)	mpix	opt-zoom	LCD-size	movies	sound	waterproof
P_1	148	8.0	4×	2.5	no	no	yes
P_2	182	8.0	5×	2.7	yes	yes	no
P_3	189	8.0	10×	2.5	yes	yes	no
P_4	196	10.0	12×	2.7	yes	no	yes
P_5	151	7.1	3×	3.0	yes	yes	no
P_6	199	9.0	3×	3.0	yes	yes	no
P_7	259	10.0	3×	3.0	yes	yes	no
P_8	278	9.1	10×	3.0	yes	yes	yes

Recommender Systems: An Introduction (slides)

Constraint Satisfaction Problem

- V is a set of variables
- D is a set of finite domains of these variables
- C is a set of constraints

Typical problems: logic puzzles (Sudoku, N-queen), scheduling

CSP: N-queens

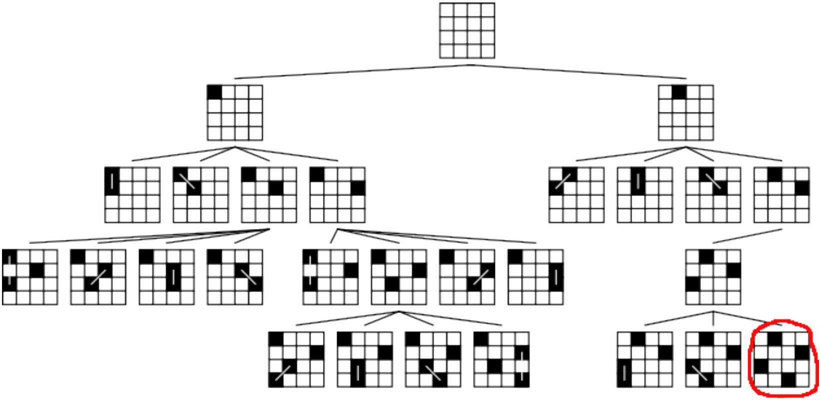
problem: place N queens on an $N \times N$ chess-board, no two queens threaten each other

- V – N variables (locations of queens)
- D – each domain is $\{1, \ldots, N\}$
- C – threatening

CSP Algorithms

- basic algorithm – backtracking
- heuristics
  - preference for some branches
  - pruning
  - … many others

CSP Example: N-queens Problem



Recommender Knowledge Base

- customer properties V_C
- product properties V_{PROD}
- constraints C_R (on customer properties)
- filter conditions C_F – relationship between customer and product
- products C_{PROD} – possible instantiations

$V_C = \{kl_c$: [expert, average, beginner] .... /* level of expertise */
wr_c: [low, medium, high] .... /* willingness to take risks */
id_c: [shortterm, mediumterm, longterm] .... /* duration of investment */
aw_c: [yes, no] .... /* advisory wanted ? */
ds_c: [savings, bonds, stockfunds, singleshares] ...... /* direct product search */
sl_c: [savings, bonds] .... /* type of low-risk investment */
av_c: [yes, no] .... /* availability of funds */
sh_c: [stockfunds, singlshares] .... /* type of high-risk investment */ $\}$

$V_{PROD} = \{name_p$: [text] .... /* name of the product */
er_p: [1..40] .... /* expected return rate */
ri_p: [low, medium, high] .... /* risk level */
$mniv_p$: [1..14] .... /* minimum investment period of product in years */
$inst_p$: [text] .... /* financial institute */ $\}$

$$C_R = \{CR_1: wr_c = high \rightarrow id_c \neq shortterm,$$
$$CR_2: kl_c = beginner \rightarrow wr_c \neq high\}$$

$$C_F = \{CF_1: id_c = shortterm \rightarrow mniv_p < 3,$$
$$CF_2: id_c = mediumterm \rightarrow mniv_p \geq 3 \wedge mniv_p < 6,$$
$$CF_3: id_c = longterm \rightarrow mniv_p \geq 6,$$
$$CF_4: wr_c = low \rightarrow ri_p = low,$$
$$CF_5: wr_c = medium \rightarrow ri_p = low \vee ri_p = medium,$$
$$CF_6: wr_c = high \rightarrow ri_p = low \vee ri_p = medium \vee ri_p = high,$$
$$CF_7: kl_c = beginner \rightarrow ri_p \neq high,$$
$$CF_8: sl_c = savings \rightarrow name_p = savings,$$
$$CF_9: sl_c = bonds \rightarrow name_p = bonds\ \}$$

$$C_{PROD} = \{CPROD_1: name_p = savings \wedge er_p = 3 \wedge ri_p = low \wedge mniv_p = 1 \wedge inst_p = A;$$
$$CPROD_2: name_p = bonds \wedge er_p = 5 \wedge ri_p = medium \wedge mniv_p = 5 \wedge inst_p = B;$$
$$CPROD_3: name_p = equity \wedge er_p = 9 \wedge ri_p = high \wedge mniv_p = 10 \wedge inst_p = B\}$$

Recommender Systems Handbook; Developing Constraint-based Recommenders

Development of Knowledge Bases

- difficult, expensive
- specilized graphical tools
- methodology (rapid prototyping, detection of faulty constraints, ...)

Unsatisfied Requirements

no solution to provided constraints

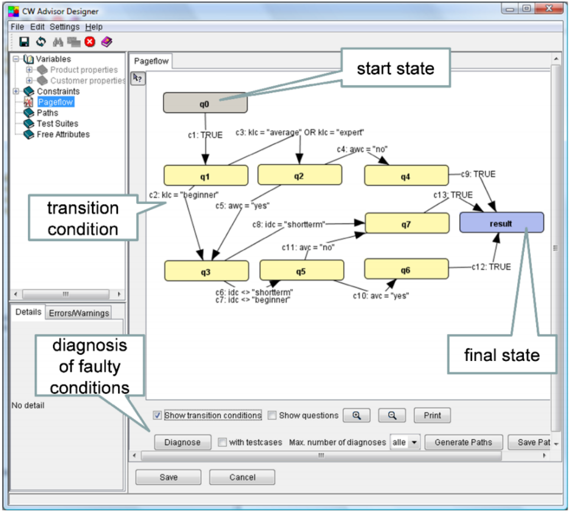
- we want to provide user at least something
- constraint relaxation
- proposing “repairs”
- minimal set of requirements to be changed

User Guidance

requirements elicitation process

- session independent user profile
- static fill-out forms
- conversational dialogs

User Guidance



Recommender Systems Handbook; Developing Constraint-based Recommenders

User Guidance

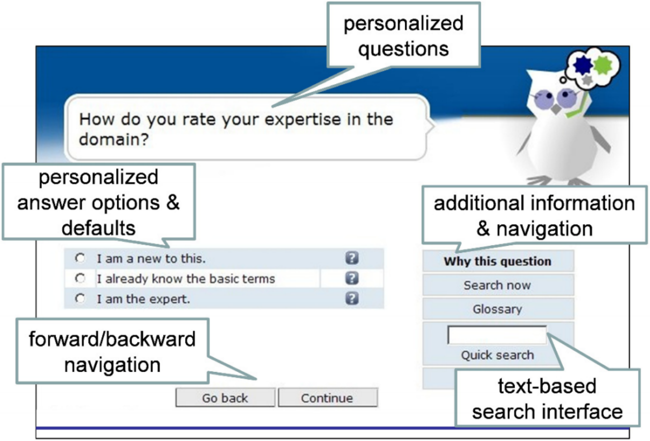


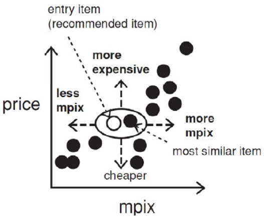
Fig. 6.4: Interactive and personalized preference elicitation example. Customers specify their preferences by answering questions.

Recommender Systems Handbook; Developing Constraint-based Recommenders

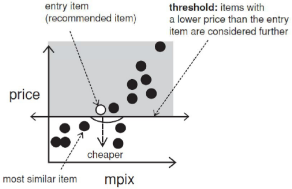
Critiquing

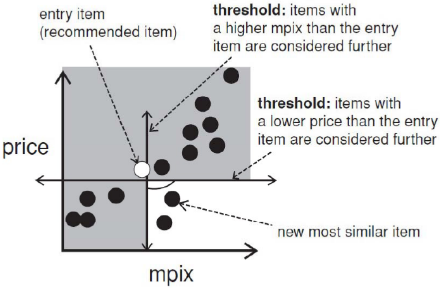
Recommender Systems: An Introduction (slides)

Critiquing



Critique on price





Critiquing: Example

A Visual Interface for Critiquing-based Recommender Systems

Critiquing: Example

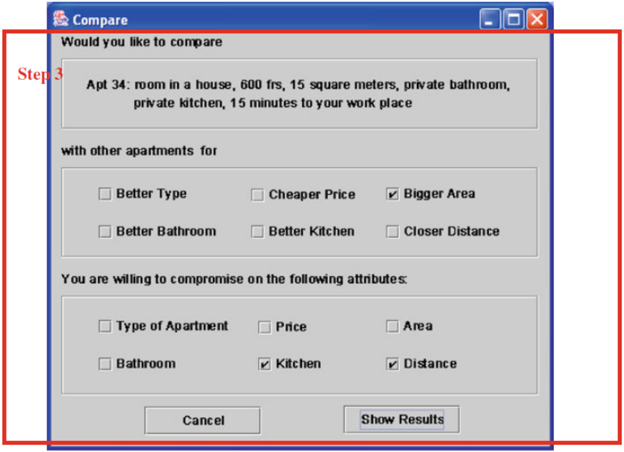


Fig. 3 Critiquing support to guide users to critique the current example product for comparing it with the other tradeoff alternatives

Critiquing: Example

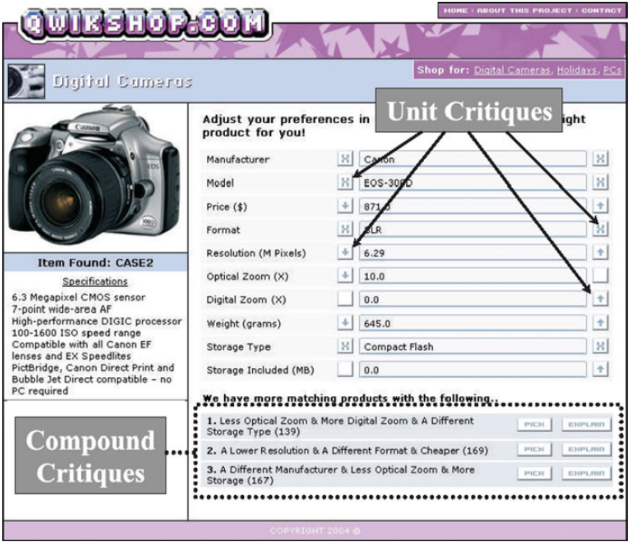


Fig. 5 The Dynamic Critiquing interface with system suggested compound critiques for users to select (McCarthy et al. 2005c)

Knowledge-based Recommendations: Limitations

- cost of knowledge acquisition (consider your project proposals)
- accuracy of models
- independence assumption for preferences

Hybrid Methods

collaborative filtering: *"what is popular among my peers"*
content-based: *"more of the same"*
knowledge-based: *"what fits my needs"*

- each has advantages and disadvantages
- hybridization – combine more techniques, avoid some shortcomings
- simple example: CF with content-based (or simple "popularity recommendation") to overcome "cold start problem"

Hybridization Designs

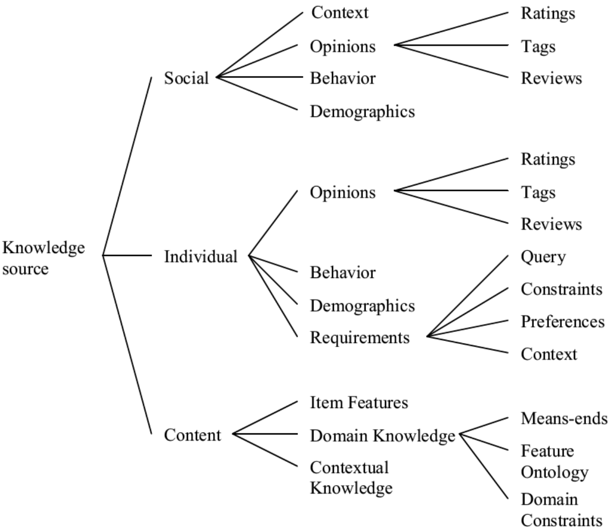
- monolitic desing, combining different features
- parallel use of several systems, weighting/voting
- pipelined invocation of different systems

Types of Recommender Systems

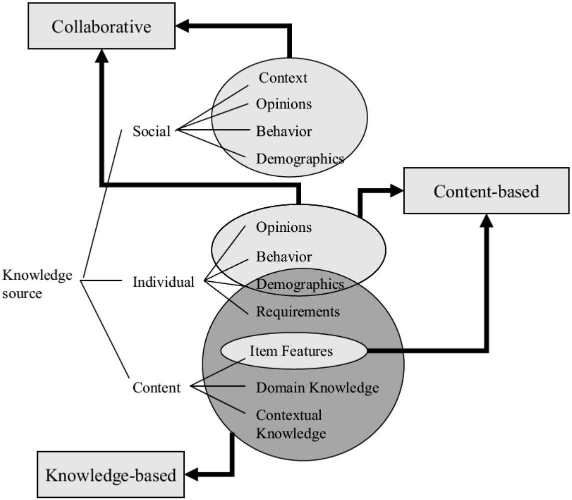
- non-personalized
- demographic
- collaborative filtering
- content based
- knowledge-based
- hybrid

what to apply when?

Taxonomy of Knowledge Sources



Knowledge Sources and Recommendation Types



Sample Domains for Recommendation

Domain	Risk	Churn	Heterog-eneous	Preferences	Interaction Style	Scrutabi-lity	Examples	Technology
News	Low	High	Low	Stable?	Implicit	Not required	Yahoo news[6] ACR news[45] and [38] Google news[16]	Content-based Collaborative-Filtering
E-commerce	Low	High	High	Stable	Implicit	Not required	Amazon.com eBay	Collaborative-Filtering
Web Page Recom-mender	Low	High	High	Unstable	Implicit	Not required	[9, 36, 4]	Collaborative-Filtering Hybrid
Movie	Low	Low	Low	Stable	Implicit	Not required	Netflix[50, 64] Movielens[21]	Collaborative-Filtering
Music	Low	Low	Low	Stable?	Implicit	Not required	Pandora and [24, 28, 14]	Content-based Hybrid
Financial-services Life-insurance	High	Low	Low	Stable	Explicit	Required	Koba4MS[17] FSAdvisor[19] [65]	Knowledge-Based
Software Engineer-ing	Low	Low	Low	Stable	Explicit /Im-plicit	Required	[13] and [29]	Hybrid and Content-based
Tourism	High	Low	Low	Unstable	Explicit	Required	Travel Recom-mender [55] [37]	Content-based Knowledge-based
Job search Recruit-ing	High	Low	Low	Stable	Explicit	Required	CASPER [35] and [39]	Content-based
Real Estate	High	Low	Low	Stable	Explicit	Required	RentMe [10] FlatFinder[67] and [73]	Knowledge-based

Matching Recommendation Technologies and Domains

Explanations of Recommendations

- recommendations: selection (ranked list) of items
- explanations: (some) reasons for the choice

Goals of Providing Explanations

Why explanations?

Goals of Providing Explanations

Why explanations?

- transparency, trustworthiness, validity, satisfaction (users are more likely to use the system)
- persuasiveness (users are more likely to follow recommendations)
- effectiveness, efficiency (users can make better/faster decisions)
- education (users understand better the behaviour of the system, may use it in better ways)

Examples of Explanations

- knowledge-based recommenders
    - "Because you, as a customer, told us that simple handling of car is important to you, we included a special sensor system in our offer that will help you park your car easily."
    - algorithms based on CSP representation

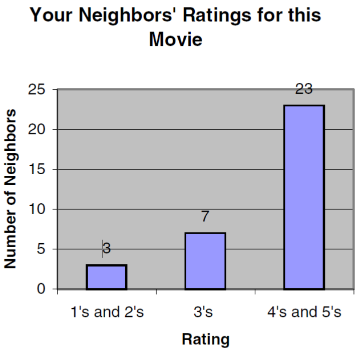
Examples of Explanations

- knowledge-based recommenders
  - "Because you, as a customer, told us that simple handling of car is important to you, we included a special sensor system in our offer that will help you park your car easily."
  - algorithms based on CSP representation
- recommendations based on item-similarity
  - "Because you watched X we recommend Y"

Explanations – Collaborative Filtering

Your Neighbors' Ratings for this Movie

Rating	Number of Neighbors
★	1
★★	2
★★★	7
★★★★	14
★★★★★	9



Explaining Collaborative Filtering Recommendations, Herlocker, Konstan, Riedl

Explanations – Collaborative Filtering

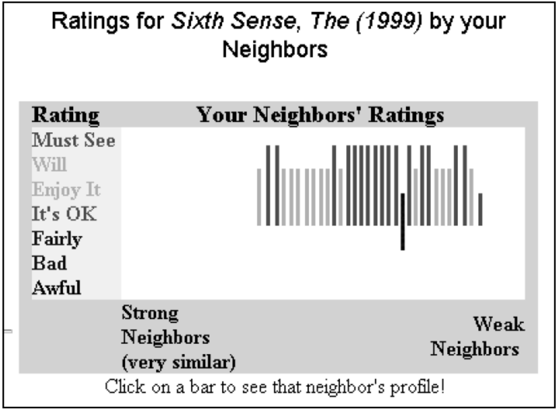


Figure 4. A screen explaining the recommendation for the movie "The Sixth Sense." Each bar represents a rating of a neighbor. Upwardly trending bars are positive ratings, while downward trending ones are negative. The x-axis represents similarity to the user.

Explaining Collaborative Filtering Recommendations, Herlocker, Konstan, Riedl

Explanations – Comparison

#		N	Mean Response	Std Dev
1	Histogram with grouping	76	5.25	1.29
2	Past performance	77	5.19	1.16
3	Neighbor ratings histogram	78	5.09	1.22
4	Table of neighbors ratings	78	4.97	1.29
5	Similarity to other movies rated	77	4.97	1.50
6	Favorite actor or actress	76	4.92	1.73
7	MovieLens percent confidence in prediction	77	4.71	1.02
8	Won awards	76	4.67	1.49
9	Detailed process description	77	4.64	1.40
10	# neighbors	75	4.60	1.29
11	No extra data – focus on system	75	4.53	1.20
12	No extra data – focus on users	78	4.51	1.35
13	MovieLens confidence in prediction	77	4.51	1.20
14	Good profile	77	4.45	1.53
15	Overall percent rated 4+	75	4.37	1.26
16	Complex graph: count, ratings, similarity	74	4.36	1.47
17	Recommended by movie critics	76	4.21	1.47
18	Rating and %agreement of closest neighbor	77	4.21	1.20
19	# neighbors with std. deviation	78	4.19	1.45
20	# neighbors with avg correlation	76	4.08	1.46
21	Overall average rating	77	3.94	1.22

Table 1. Mean response of users to each explanation interface, based on a scale of one to seven. Explanations 11 and 12 represent the base case of no additional information. Shaded rows indicate explanations with a mean response significantly different from the base cases (two-tailed $\alpha = 0.05$).

Moment of Recommendation

- front page, dashboard
- follow-up
- sidebar
- on demand

recommendation placements may differ in their requirements

Your Projects: Questions

- What is the purpose / use case? What is the "business model"?
- What will you recommend? In what situation?
- A new system or extention of an existing one?
- **What data you have?**
  - items
  - user preferences; explicit/implicit ratings?
- Which techniques are relevant/suitable for you project? Collaborative filtering? Content-based? Knowledge-based? Combination?
- Are the following notions relevant: taxonomy, critiquing, explanations?