technische universität
dortmund

ALPEN-ADRIA
UNIVERSITÄT
KLAGENFURT | WIEN GRAZ

Tutorial: Recommender Systems

International Joint Conference on Artificial Intelligence
Barcelona, July 17, 2011

Dietmar Jannach
TU Dortmund

Gerhard Friedrich
Alpen-Adria Universität Klagenfurt


Recommender Systems: An Introduction
by Dietmar Jannach, Markus Zanker, Alexander Felfernig, Gerhard Friedrich
AVERAGE CUSTOMER RATING:
(Be the first to review)
Gefällt mir
Registrieren, um sehen zu können, was deinen Freunden gefällt.
FORMAT:
Hardcover
NOOKbook (eBook) - not available
Tell the publisher you want this in NOOKbook format
NEW FROM BN.COM
~~$65.00~~ List Price
$52.00 Online Price
(You Save 20%)
Add to Cart
NEW & USED FROM OUR
New starting at $56.46(You S
Used starting at $51.98(You
See All Prices
Table of Contents
Customers who bought this also bought
Recommender Systems
An Introduction
Search Patterns
Hello, Android
Disconnect
DEVRA DAVIS
iPhone For Dummies

About the speakers

- **Gerhard Friedrich**
  - Professor at University Klagenfurt
- **Dietmar Jannach**
  - Professor at TU Dortmund, Germany
- **Research background and interests**
  - Application of Intelligent Systems technology in business
    - Recommender systems implementation & evaluation
    - Product configuration systems
    - Web mining
    - Operations research

Agenda

- **What are recommender systems for?**
  - Introduction
- **How do they work (Part I) ?**
  - Collaborative Filtering
- **How to measure their success?**
  - Evaluation techniques
- **How do they work (Part II) ?**
  - Content-based Filtering
  - Knowledge-Based Recommendations
  - Hybridization Strategies
- **Advanced topics**
  - Explanations
  - Human decision making

Introduction

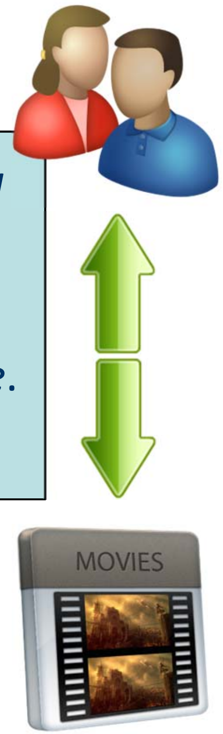
Problem domain

- **Recommendation systems (RS) help to match users with items**
  - Ease information overload
  - Sales assistance (guidance, advisory, persuasion,…)

RS are software agents that elicit the interests and preferences of individual consumers […] and make recommendations accordingly.

They have the potential to support and improve the quality of the decisions consumers make while searching for and selecting products online.

» [Xiao & Benbasat, MISQ, 2007]



- **Different system designs / paradigms**
  - Based on availability of exploitable data
  - Implicit and explicit user feedback
  - Domain characteristics

Purpose and success criteria (1)

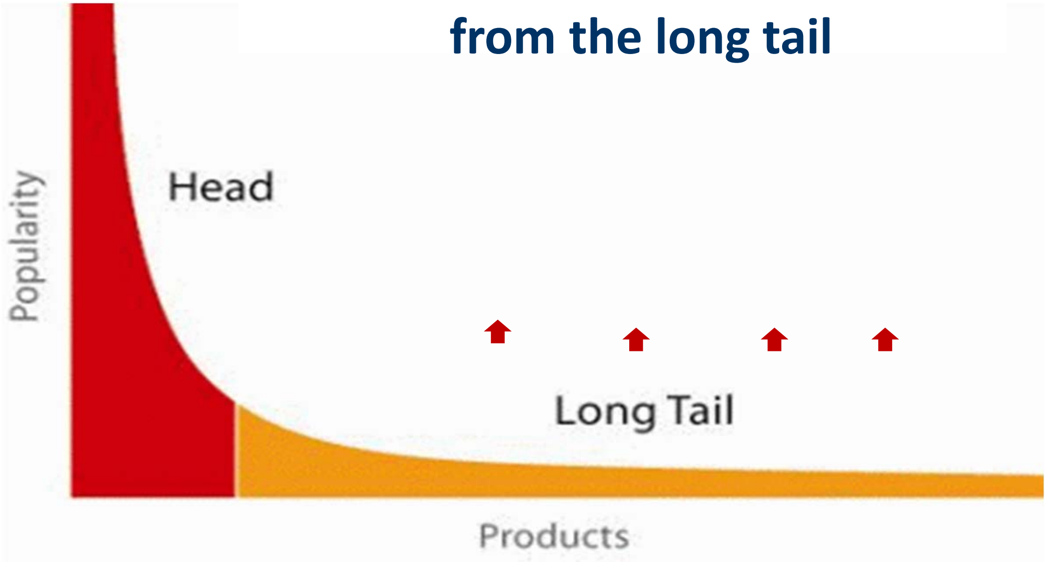
- **Different perspectives/aspects**
  - Depends on domain and purpose
  - No holistic evaluation scenario exists

- **Retrieval perspective**
  - Reduce search costs
  - Provide "correct" proposals
  - Users know in advance what they want

- **Recommendation perspective**
  - Serendipity – identify items from the Long Tail
  - Users did not know about existence

When does a RS do its job well?

Recommend items from the long tail



- "Recommend widely unknown items that users might actually like!"
- 20% of items accumulate 74% of all positive ratings
- Items rated > 3 in MovieLens 100K dataset

Purpose and success criteria (2)

- **Prediction perspective**
  - Predict to what degree users like an item
  - Most popular evaluation scenario in research

- **Interaction perspective**
  - Give users a "good feeling"
  - Educate users about the product domain
  - Convince/persuade users - explain

- **Finally, conversion perspective**
  - Commercial situations
  - Increase "hit", "clickthrough", "lookers to bookers" rates
  - Optimize sales margins and profit

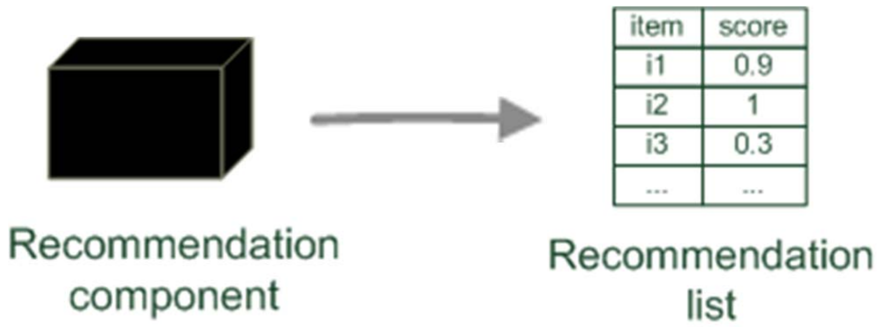
Recommender systems

- **RS seen as a function** [AT05]
- **Given:**
  - User model (e.g. ratings, preferences, demographics, situational context)
  - Items (with or without description of item characteristics)
- **Find:**
  - Relevance score. Used for ranking.

- **Relation to Information Retrieval:**
  - IR is finding material [..] of an unstructured nature [..] that satisfies an information need from within large collections [..].
    - » [Manning et al., CUP, 2008]

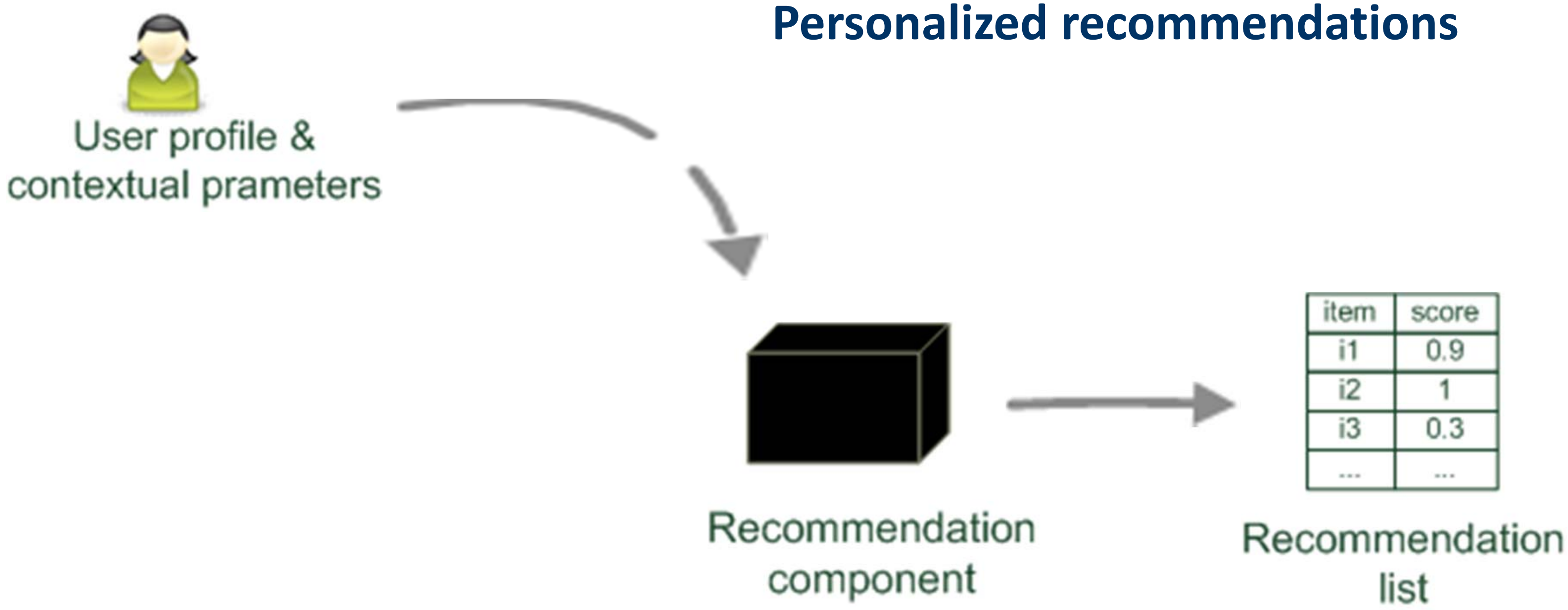
Paradigms of recommender systems

Recommender systems reduce information overload by estimating relevance

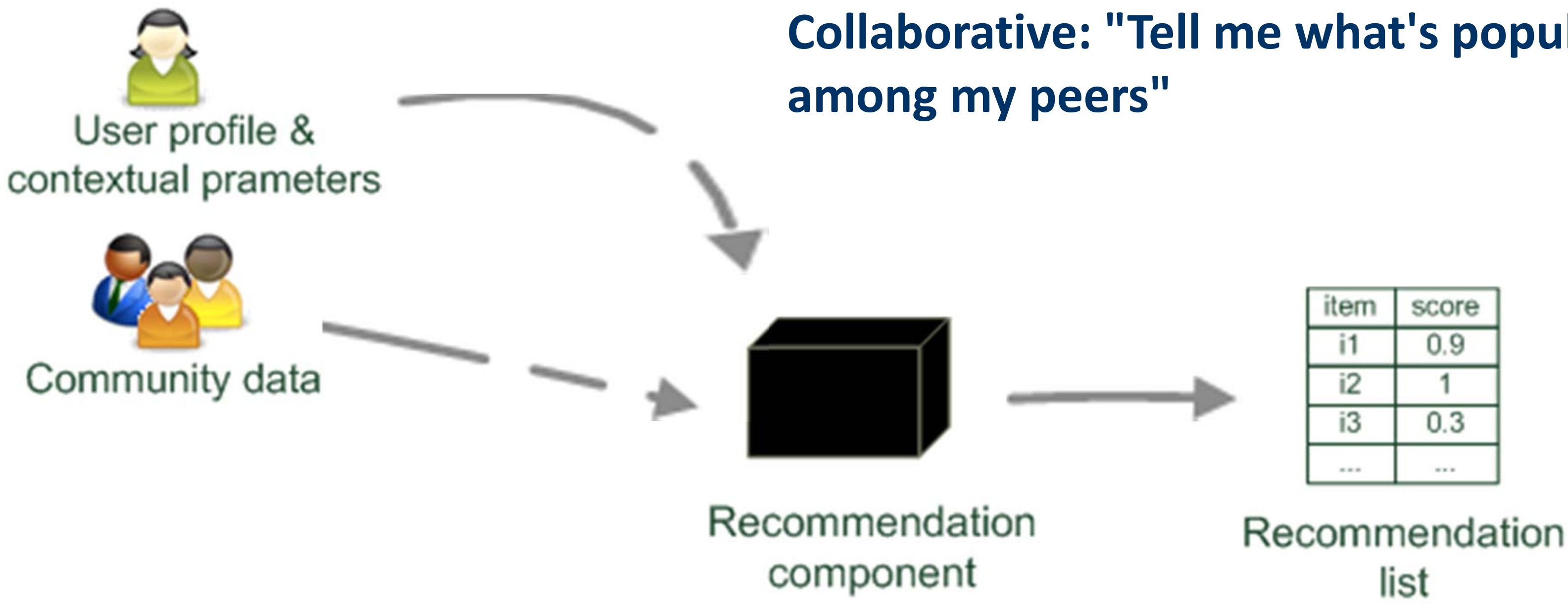


Paradigms of recommender systems

Personalized recommendations



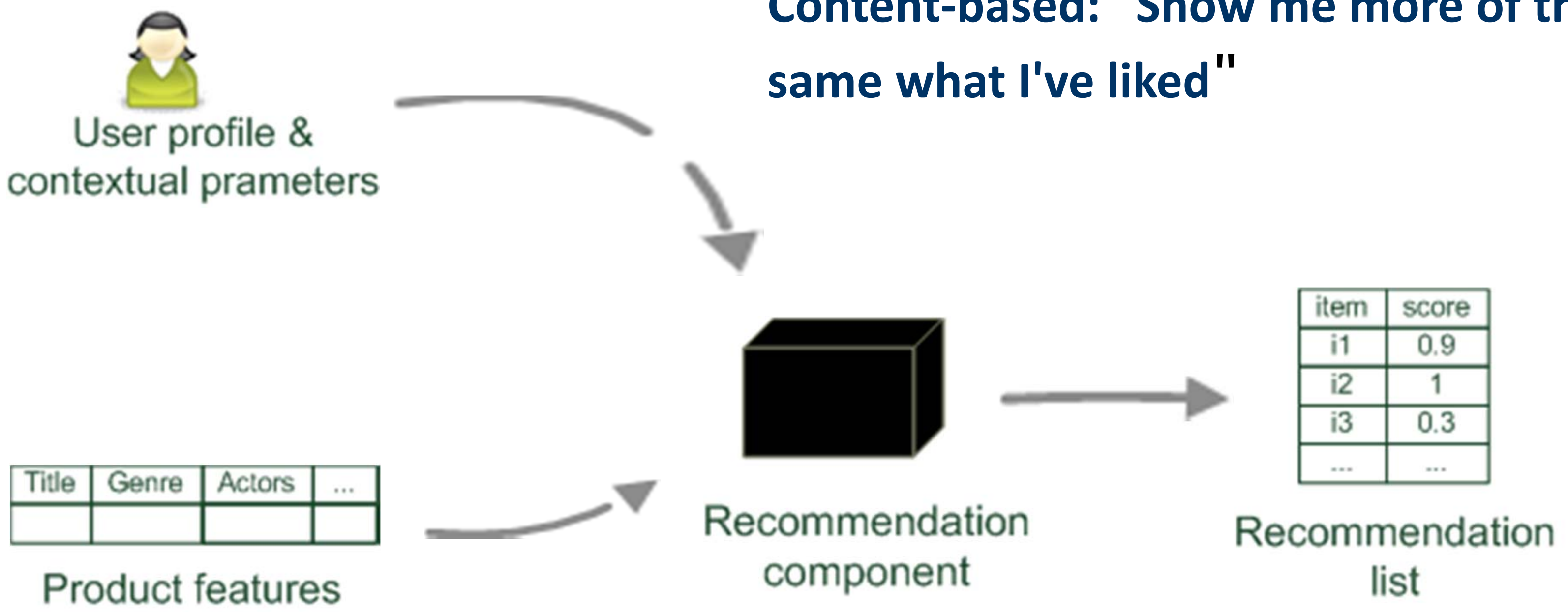
Paradigms of recommender systems



Collaborative: "Tell me what's popular among my peers"

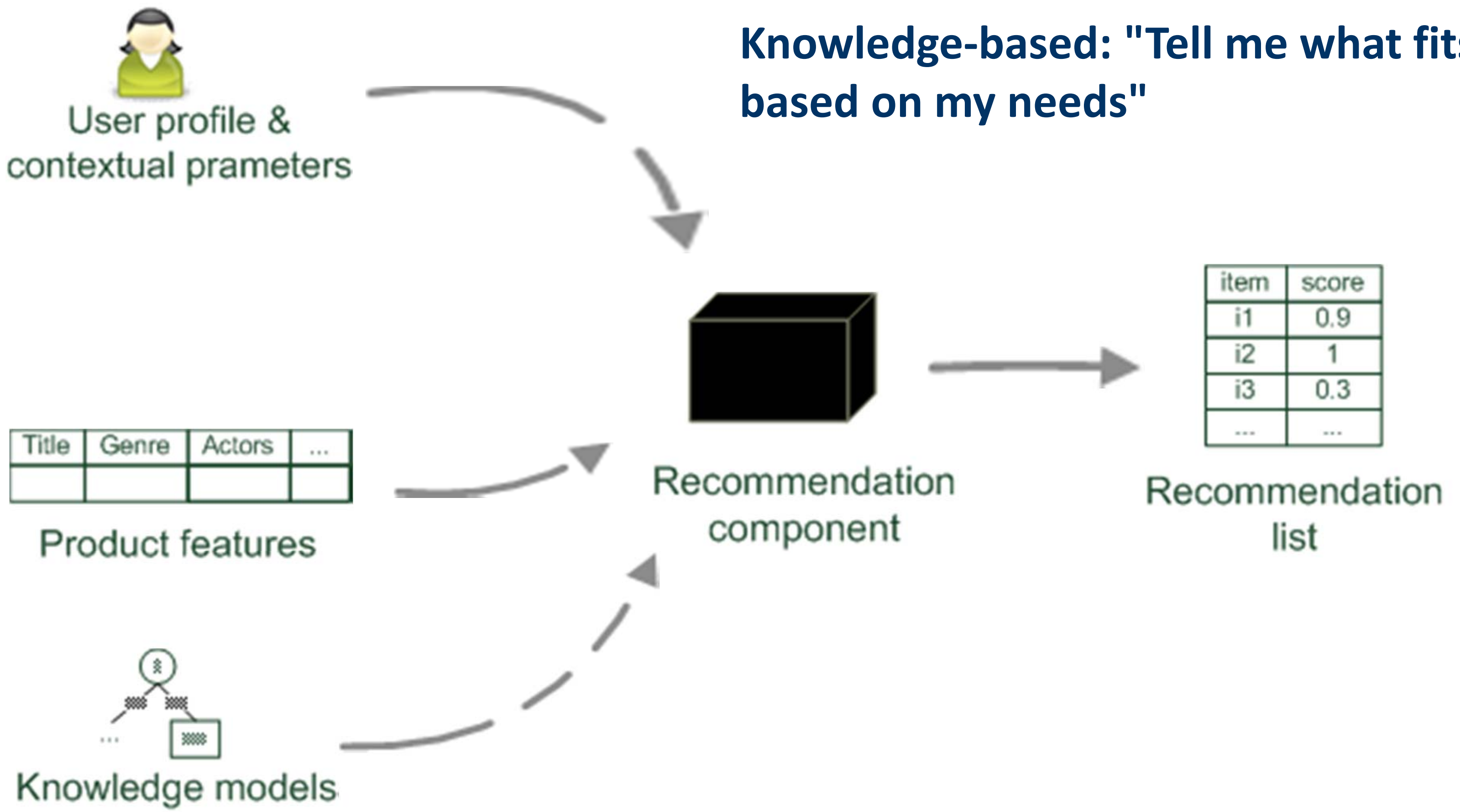
Paradigms of recommender systems

Content-based: "Show me more of the same what I've liked"



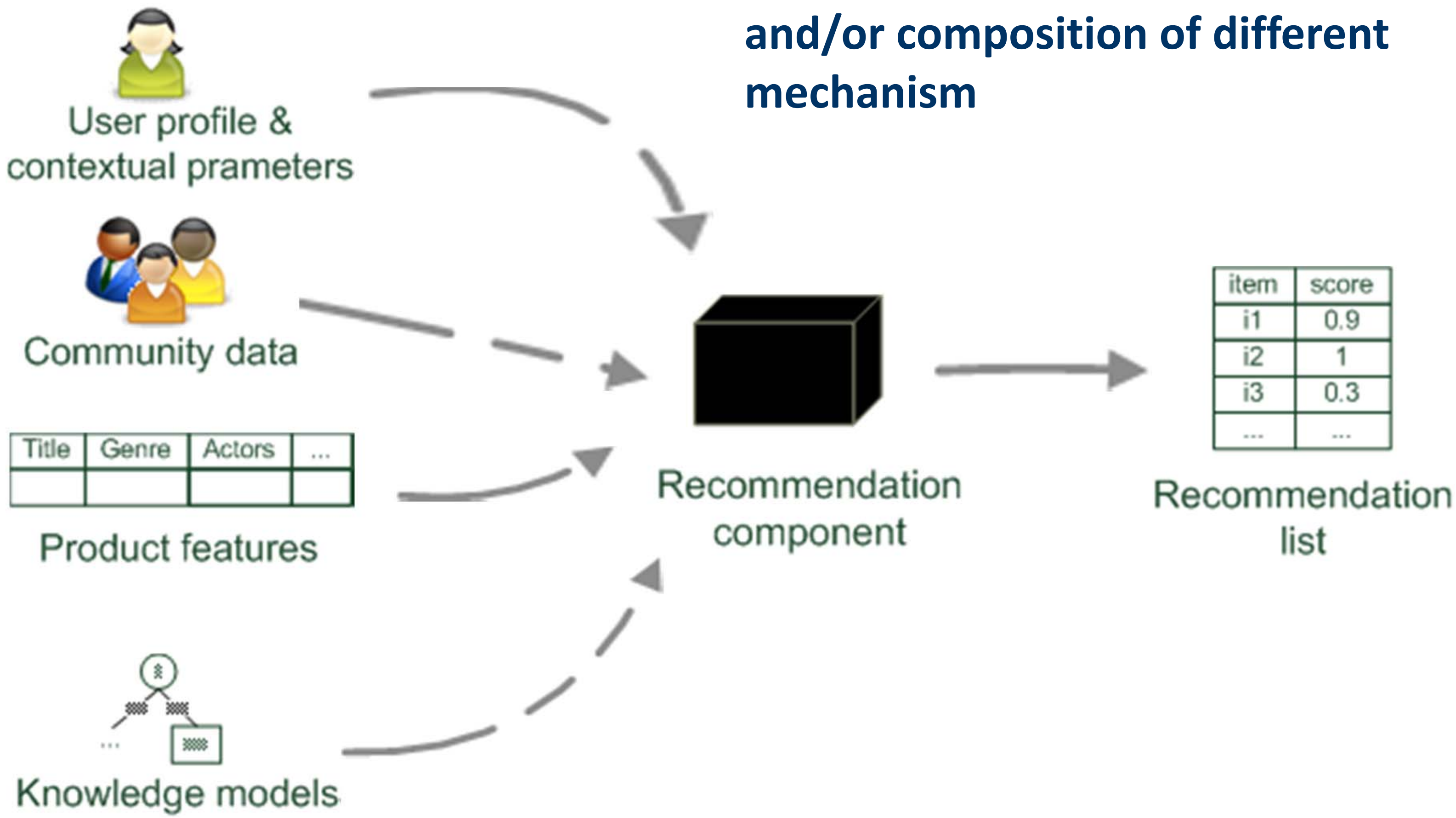
Paradigms of recommender systems

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



Paradigms of recommender systems

Hybrid: combinations of various inputs and/or composition of different mechanism



Recommender systems: basic techniques

	Pros	Cons
Collaborative	Nearly no ramp-up effort, serendipity of results, learns market segments	Requires some form of rating feedback, cold start for new users and new items
Content-based	No community required, comparison between items possible	Content-descriptions necessary, cold start for new users, no surprises
Knowledge-based	Deterministic recommendations, assured quality, no cold-start, can resemble sales dialogue	Knowledge engineering effort to bootstrap, basically static, does not react to short-term trends

Collaborative Filtering

Collaborative Filtering (CF)

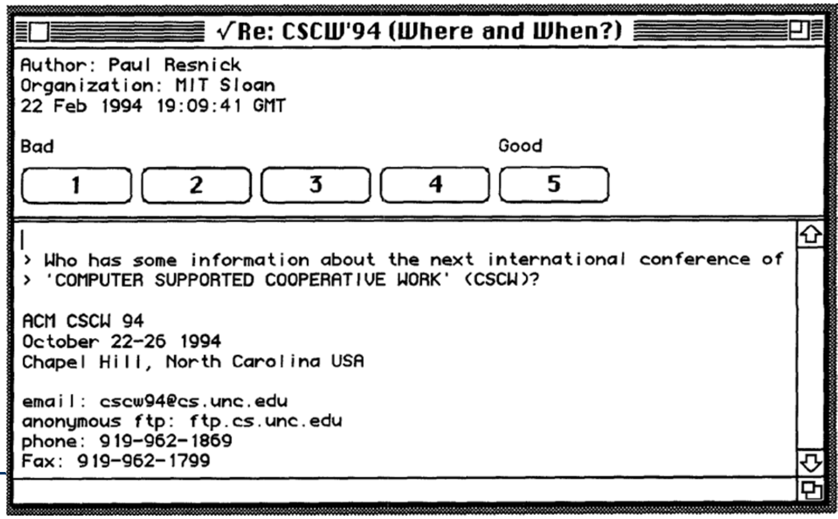
- **The most prominent approach to generate recommendations**
  - used by large, commercial e-commerce sites
  - well-understood, various algorithms and variations exist
  - applicable in many domains (book, movies, DVDs, ..)
- **Approach**
  - use the "wisdom of the crowd" to recommend items
- **Basic assumption and idea**
  - Users give ratings to catalog items (implicitly or explicitly)
  - Customers who had similar tastes in the past, will have similar tastes in the future

1992: *Using collaborative filtering to weave an information tapestry*, D. Goldberg et al., Communications of the ACM

- **Basic idea: "Eager readers read all docs immediately, casual readers wait for the eager readers to annotate"**
- **Experimental mail system at Xerox Parc that records reactions of users when reading a mail**
- **Users are provided with personalized mailing list filters instead of being forced to subscribe**
  - Content-based filters (topics, from/to/subject…)
  - Collaborative filters
- **E.g. Mails to [all] which were replied by [John Doe] and which received positive ratings from [X] and [Y].**

1994: *GroupLens: an open architecture for collaborative filtering of netnews*, P. Resnick et al., ACM CSCW

- **Tapestry system does not aggregate ratings and requires knowing each other**
- **Basic idea: "People who agreed in their subjective evaluations in the past are likely to agree again in the future"**
- **Builds on newsgroup browsers with rating functionality**



User-based nearest-neighbor collaborative filtering (1)

- **The basic technique:**
  - Given an "active user" (Alice) and an item I not yet seen by Alice
  - The *goal is to estimate Alice's rating for this item*, e.g., by
    - find a set of users (peers) who liked the same items as Alice in the past **and** who have rated item I
    - use, e.g. the average of their ratings to predict, if Alice will like item I
    - do this for all items Alice has not seen and recommend the best-rated

	Item1	Item2	Item3	Item4	Item5
Alice	5	3	4	4	?
User1	3	1	2	3	3
User2	4	3	4	3	5
User3	3	3	1	5	4
User4	1	5	5	2	1

User-based nearest-neighbor collaborative filtering (2)

- **Some first questions**
  - How do we measure similarity?
  - How many neighbors should we consider?
  - How do we generate a prediction from the neighbors' ratings?

	Item1	Item2	Item3	Item4	Item5
Alice	5	3	4	4	?
User1	3	1	2	3	3
User2	4	3	4	3	5
User3	3	3	1	5	4
User4	1	5	5	2	1

Measuring user similarity

- **A popular similarity measure in user-based CF: Pearson correlation**

$$sim(a,b) = \frac{\sum_{p \in P}(r_{a,p} - \bar{r}_a)(r_{b,p} - \bar{r}_b)}{\sqrt{\sum_{p \in P}(r_{a,p} - \bar{r}_a)^2}\sqrt{\sum_{p \in P}(r_{b,p} - \bar{r}_b)^2}}$$

a, b : users

$r_{a,p}$: rating of user a for item p

P : set of items, rated both by a and b

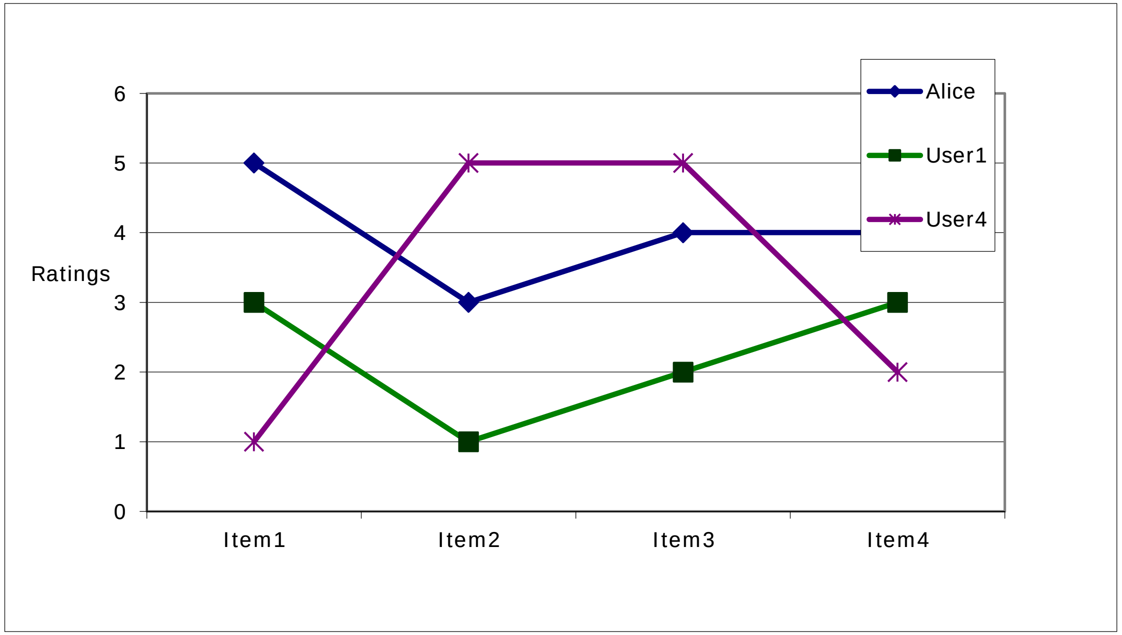
Possible similarity values between -1 and 1; $\bar{r}_a, \bar{r}_b$ = user's average ratings

	Item1	Item2	Item3	Item4	Item5
Alice	5	3	4	4	?
User1	3	1	2	3	3
User2	4	3	4	3	5
User3	3	3	1	5	4
User4	1	5	5	2	1

sim = 0,85
sim = 0,70
sim = -0,79

Pearson correlation

- **Takes differences in rating behavior into account**



- **Works well in usual domains, compared with alternative measures**
  - such as cosine similarity

Making predictions

- A common prediction function:

$$pred(a,p) = \overline{r_a} + \frac{\sum_{b \in N} sim(a,b) * (r_{b,p} - \overline{r_b})}{\sum_{b \in N} sim(a,b)}$$

- Calculate, whether the neighbors' ratings for the unseen item ***i*** are higher or lower than their average
- Combine the rating differences – use the similarity with as a weight
- Add/subtract the neighbors' bias from the active user's average and use this as a prediction

Improving the metrics / prediction function

- **Not all neighbor ratings might be equally "valuable"**
  - Agreement on commonly liked items is not so informative as agreement on controversial items
  - **Possible solution**: Give more weight to items that have a higher variance
- **Value of number of co-rated items**
  - Use "significance weighting", by e.g., linearly reducing the weight when the number of co-rated items is low
- **Case amplification**
  - Intuition: Give more weight to "very similar" neighbors, i.e., where the similarity value is close to 1.
- **Neighborhood selection**
  - Use similarity threshold or fixed number of neighbors

Memory-based and model-based approaches

- **User-based CF is said to be "memory-based"**
  - the rating matrix is directly used to find neighbors / make predictions
  - does not scale for most real-world scenarios
  - large e-commerce sites have tens of millions of customers and millions of items
- **Model-based approaches**
  - based on an offline pre-processing or "model-learning" phase
  - at run-time, only the learned model is used to make predictions
  - models are updated / re-trained periodically
  - large variety of techniques used
  - model-building and updating can be computationally expensive

2001: *Item-based collaborative filtering recommendation algorithms*, B. Sarwar et al., WWW 2001

- **Scalability issues arise with U2U if many more users than items ($m >> n$, $m = |users|$, $n = |items|$)**
  - e.g. amazon.com
  - Space complexity $O(m^2)$ when pre-computed
  - Time complexity for computing Pearson $O(m^2n)$

- **High sparsity leads to few common ratings between two users**

- **Basic idea: "Item-based CF exploits relationships between items first, instead of relationships between users"**

Item-based collaborative filtering

- **Basic idea:**
  - Use the similarity between items (and not users) to make predictions
- **Example:**
  - Look for items that are similar to Item5
  - Take Alice's ratings for these items to predict the rating for Item5

	Item1	Item2	Item3	Item4	Item5
Alice	5	3	4	4	?
User1	3	1	2	3	3
User2	4	3	4	3	5
User3	3	3	1	5	4
User4	1	5	5	2	1

The cosine similarity measure

- **Produces better results in item-to-item filtering**
  - for some datasets, no consistent picture in literature
- **Ratings are seen as vector in n-dimensional space**
- **Similarity is calculated based on the angle between the vectors**

$$sim(\vec{a}, \vec{b}) = \frac{\vec{a} \cdot \vec{b}}{|\,\vec{a}\,| * |\,\vec{b}\,|}$$

- **Adjusted cosine similarity**
  - take average user ratings into account, transform the original ratings

$$sim(a, b) = \frac{\sum_{u \in U}(r_{u,a} - \overline{r_u})(r_{u,b} - \overline{r_u})}{\sqrt{\sum_{u \in U}(r_{u,a} - \overline{r_u})^2}\sqrt{\sum_{u \in U}(r_{u,b} - \overline{r_u})^2}}$$

Pre-processing for item-based filtering

- **Item-based filtering does not solve the scalability problem itself**
- **Pre-processing approach by Amazon.com (in 2003)**
  - Calculate all pair-wise item similarities in advance
  - The neighborhood to be used at run-time is typically rather small, because only items are taken into account which the user has rated
  - Item similarities are supposed to be more stable than user similarities
- **Memory requirements**
  - Up to N^2 pair-wise similarities to be memorized (N = number of items) in theory
  - In practice, this is significantly lower (items with no co-ratings)
  - Further reductions possible
    - Minimum threshold for co-ratings (items, which are rated at least by *n* users)
    - Limit the size of the neighborhood (might affect recommendation accuracy)

More on ratings

- **Pure CF-based systems only rely on the rating matrix**
- **Explicit ratings**
  - Most commonly used (1 to 5, 1 to 7 Likert response scales)
  - Research topics
    - "Optimal" granularity of scale; indication that 10-point scale is better accepted in movie domain
    - Multidimensional ratings (multiple ratings per movie)
  - Challenge
    - Users not always willing to rate many items; sparse rating matrices
    - How to stimulate users to rate more items?
- **Implicit ratings**
  - clicks, page views, time spent on some page, demo downloads …
  - Can be used in addition to explicit ones; question of correctness of interpretation

Data sparsity problems

- **Cold start problem**
  - How to recommend new items? What to recommend to new users?
- **Straightforward approaches**
  - Ask/force users to rate a set of items
  - Use another method (e.g., content-based, demographic or simply non-personalized) in the initial phase
- **Alternatives**
  - Use better algorithms (beyond nearest-neighbor approaches)
  - Example:
    - In nearest-neighbor approaches, the set of sufficiently similar neighbors might be to small to make good predictions
    - Assume "transitivity" of neighborhoods

Example algorithms for sparse datasets

- **Recursive CF**
  - Assume there is a very close neighbor ***n*** of u who however has not rated the target item ***i*** yet.
  - Idea:
    - Apply CF-method recursively and predict a rating for item ***i*** for the neighbor
    - Use this predicted rating instead of the rating of a more distant direct neighbor

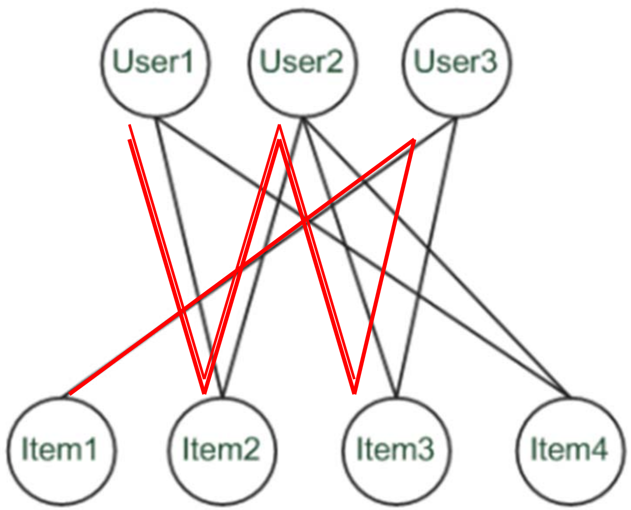
	Item1	Item2	Item3	Item4	Item5
Alice	5	3	4	4	?
User1	3	1	2	3	?
User2	4	3	4	3	5
User3	3	3	1	5	4
User4	1	5	5	2	1

sim = 0,85

Predict rating for User1

Graph-based methods

- **"Spreading activation" (sketch)**
  - Idea: Use paths of lengths > 3 to recommend items
  - Length 3: Recommend Item3 to User1
  - Length 5: Item1 also recommendable



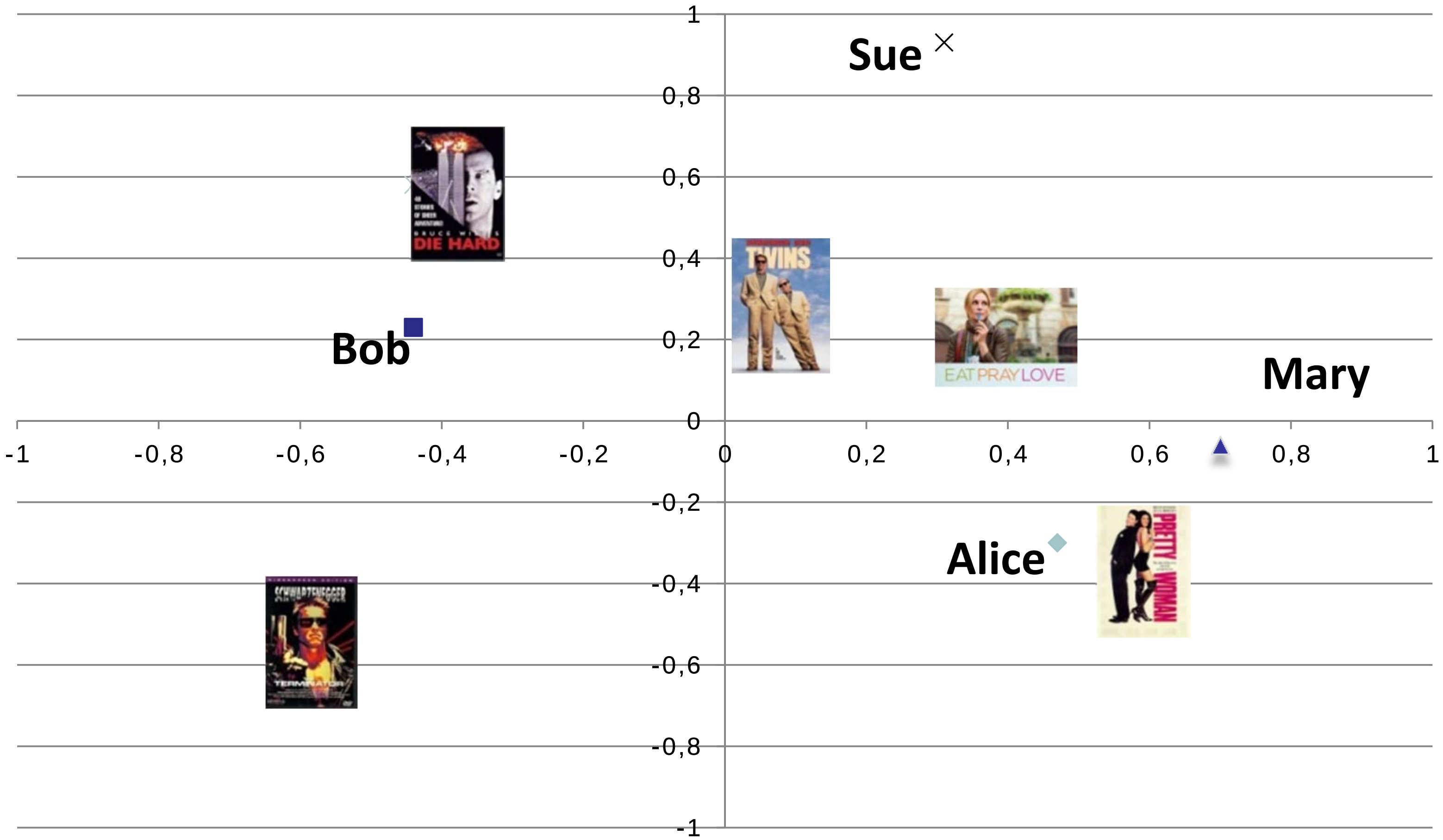
More model-based approaches

- **Plethora of different techniques proposed in the last years, e.g.,**
  - Matrix factorization techniques, statistics
    - singular value decomposition, principal component analysis
  - Association rule mining
    - compare: shopping basket analysis
  - Probabilistic models
    - clustering models, Bayesian networks, probabilistic Latent Semantic Analysis
  - Various other machine learning approaches
- **Costs of pre-processing**
  - Usually not discussed
  - Incremental updates possible?

2000: *Application of Dimensionality Reduction in Recommender System*, B. Sarwar et al., WebKDD Workshop

- **Basic idea: Trade more complex offline model building for faster online prediction generation**
- **Singular Value Decomposition for dimensionality reduction of rating matrices**
  - Captures important factors/aspects and their weights in the data
  - factors can be genre, actors but also non-understandable ones
  - Assumption that k dimensions capture the signals and filter out noise (K = 20 to 100)
- **Constant time to make recommendations**
- **Approach also popular in IR (Latent Semantic Indexing), data compression,…**

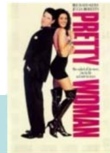
A picture says …



Matrix factorization

- **SVD:** $$M_k = U_k \times \Sigma_k \times V_k^{\ T}$$

U_k	Dim1	Dim2
Alice	0.47	-0.30
Bob	-0.44	0.23
Mary	0.70	-0.06
Sue	0.31	0.93

$V_k^{\ T}$					
Dim1	-0.44	-0.57	0.06	0.38	0.57
Dim2	0.58	-0.66	0.26	0.18	-0.36

Σ_k	Dim1	Dim2
Dim1	5.63	0
Dim2	0	3.23

- **Prediction:** $$\hat{r}_{ui} = \bar{r}_u + U_k(Alice) \times \Sigma_k \times V_k^T(EPL)$$

  = 3 + 0.84 = 3.84

Association rule mining

- **Commonly used for shopping behavior analysis**
  - aims at detection of rules such as
    "If a customer purchases baby-food then he also buys diapers in 70% of the cases"
- **Association rule mining algorithms**
  - can detect rules of the form X => Y (e.g., baby-food => diapers) from a set of sales transactions D = $\{t_1, t_2, \ldots t_n\}$
  - measure of quality: support, confidence
    - used e.g. as a threshold to cut off unimportant rules
  - let $\sigma(X) = \frac{|\{x|x \subseteq ti, ti \in D\}|}{|D|}$
  - support = $\frac{\sigma(X \cup Y)}{|D|}$, confidence = $\frac{\sigma(X \cup Y)}{\sigma(X)}$

$$support = \frac{|X \cup Y|}{|Transactions|}$$

$$confidence = \frac{|X \cup Y|}{|X|}$$

Recommendation based on Association Rule Mining

	Item1	Item2	Item3	Item4	Item5
Alice	1	0	0	0	?
User1	1	0	1	0	1
User2	1	0	1	0	1
User3	0	0	0	1	1
User4	0	1	1	0	0

- **Simplest approach**
  - transform 5-point ratings into binary ratings (1 = above user average)
- **Mine rules such as**
  - Item1 => Item5
    - support (2/4), confidence (2/2) (without Alice)
- **Make recommendations for Alice (basic method)**
  - Determine "relevant" rules based on Alice's transactions (the above rule will be relevant as Alice bought Item1)
  - Determine items not already bought by Alice
  - Sort the items based on the rules' confidence values
- **Different variations possible**
  - dislike statements, user associations ..

Probabilistic methods

- **Basic idea (simplistic version for illustration):**
  - given the user/item rating matrix
  - determine the probability that user Alice will like an item i
  - base the recommendation on such these probabilities
- **Calculation of rating probabilities based on Bayes Theorem**
  - How probable is rating value "1" for Item5 given Alice's previous ratings?
  - Corresponds to conditional probability P(Item5=1 | X), where
    - X = Alice's previous ratings = (Item1 =1, Item2=3, Item3= …)
  - Can be estimated based on Bayes' Theorem

$$P(Y|X) = \frac{P(X|Y) \times P(Y)}{P(X)} \qquad P(Y|X) = \frac{\prod_{i=1}^{d} P(X_i|Y) \times P(Y)}{P(X)}$$

  - Assumption: Ratings are independent (?), d = number of attributes in X

Calculation of probabilities in simplistic approach

	Item1	Item2	Item3	Item4	Item5
Alice	1	3	3	2	?
User1	2	4	2	2	4
User2	1	3	3	5	1
User3	4	5	2	3	3
User4	1	1	5	2	1

X = (Item1 =1, Item2=3, Item3= …)

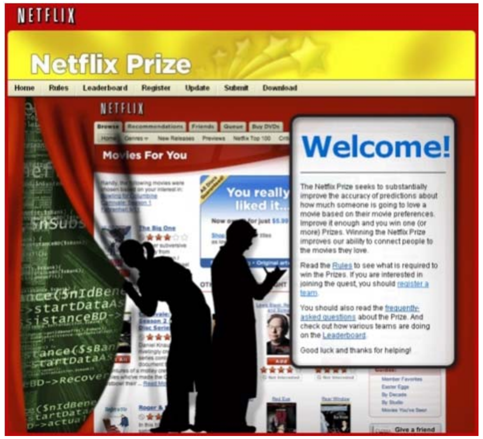
$$
\begin{aligned}
P(X|Item5{=}1) &= P(Item1{=}1|Item5{=}1) \times P(Item2{=}3|Item5{=}1) \times \\
&\quad P(Item3{=}3|Item5{=}1) \times P(Item4{=}2|Item5{=}1) \\
&= 2/4 \times 1/4 \times 1/4 \times 1/4 \\
&\approx 0.0078125 \\
P(X|Item5{=}2) &= P(Item1{=}1|Item5{=}2) \times P(Item2{=}3|Item5{=}2) \times \\
&\quad P(Item3{=}3|Item5{=}2) \times P(Item4{=}2|Item5{=}2) \\
&= 0/4 \times \ldots \times \ldots \times \ldots. \\
&= 0
\end{aligned}
$$

- **More to consider**
  - Zeros (smoothing required)
  - like/dislike simplification possible

Practical probabilistic approaches

- **Use a cluster-based approach**
  - assume users fall in a small number of subgroups (clusters)
  - Make predictions based on estimates
    - probability of Alice falling into cluster c
    - probability of Alice liking item i given a certain cluster and her previous ratings
  - Based on model-based clustering (mixture model)
    - Number of classes and model parameters have to be learned from data in advance (EM algorithm)
- **Others:**
  - Bayesian Networks, Probabilistic Latent Semantic Analysis, ….
- **Empirical analysis shows:**
  - Probabilistic methods lead to relatively good results (movie domain)
  - No consistent winner; small memory-footprint of network model

2008: ***Factorization meets the neighborhood: a multifaceted collaborative filtering model*, Y. Koren, ACM SIGKDD**



- **Stimulated by work on Netflix competition**
  - **Prize of $1,000,000 for accuracy improvement of 10% RMSE compared to own Cinematch system**
  - **Very large dataset (~100M ratings, ~480K users , ~18K movies)**
  - **Last ratings/user withheld (set K)**
- **Root mean squared error metric optimized to 0.8567**

$$RMSE = \sqrt{\frac{\sum_{(u,i)\in K} (\hat{r}_{ui} - r_{ui})^2}{|K|}}$$

2008: ***Factorization meets the neighborhood: a multifaceted collaborative filtering model*, Y. Koren, ACM SIGKDD**

- **Merges neighborhood models with latent factor models**
- **Latent factor models**
  - **good to capture weak signals in the overall data**
- **Neighborhood models**
  - **good at detecting strong relationships between close items**
- **Combination in one prediction single function**
  - Local search method such as stochastic gradient descent to determine parameters
  - Add penalty for high values to avoid over-fitting

$$\hat{r}_{ui} = \mu + b_u + b_i + p_u^T q_i$$

$$\min_{p_*, q_*, b_*} \sum_{(u,i) \in K} (r_{ui} - \mu - b_u - b_i - p_u^T q_i)^2 + \lambda (\|p_u\|^2 + \|q_i\|^2 + b_u^2 + b_i^2)$$

Summarizing recent methods

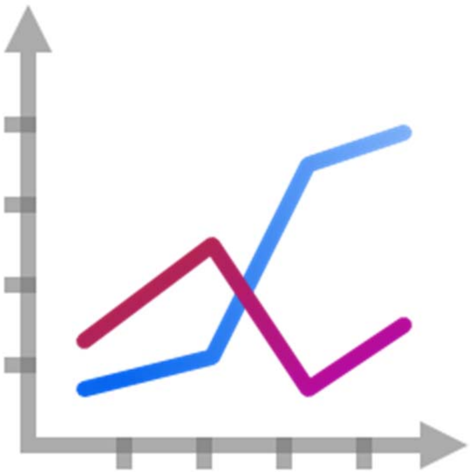
- **Recommendation is concerned with learning from noisy observations (x,y), where $f(x) = \hat{y}$ has to be determined such that $\sum_{\hat{y}} (\hat{y} - y)^2$ is minimal.**

- **A huge variety of different learning strategies have been applied trying to estimate f(x)**
  - Non parametric neighborhood models
  - MF models, SVMs, Neural Networks, Bayesian Networks,…

Collaborative Filtering Issues

- **Pros:** 
  - well-understood, works well in some domains, no knowledge engineering required
- **Cons:** 
  - requires user community, sparsity problems, no integration of other knowledge sources, no explanation of results
- **What is the best CF method?**
  - In which situation and which domain? Inconsistent findings; always the same domains and data sets; differences between methods are often very small (1/100)
- **How to evaluate the prediction quality?**
  - MAE / RMSE: What does an MAE of 0.7 actually mean?
  - Serendipity: Not yet fully understood
- **What about multi-dimensional ratings?**

Evaluation of Recommender Systems



Evaluating Recommender Systems

- **A myriad of techniques has been proposed, but**
  - Which one is the best in a given application domain?
  - What are the success factors of different techniques?
  - Comparative analysis based on an optimality criterion?

- **Research questions are:**
  - Is a RS efficient with respect to a specific criteria like accuracy, user satisfaction, response time, serendipity, online conversion, ramp-up efforts, ….
  - Do customers like/buy recommended items?
  - Do customers buy items they otherwise would have not?
  - Are they satisfied with a recommendation after purchase?

Empirical research

- **Characterizing dimensions:**
  - Who is the **subject** that is in the focus of research?
  - What **research methods** are applied?
  - In which **setting** does the research take place?

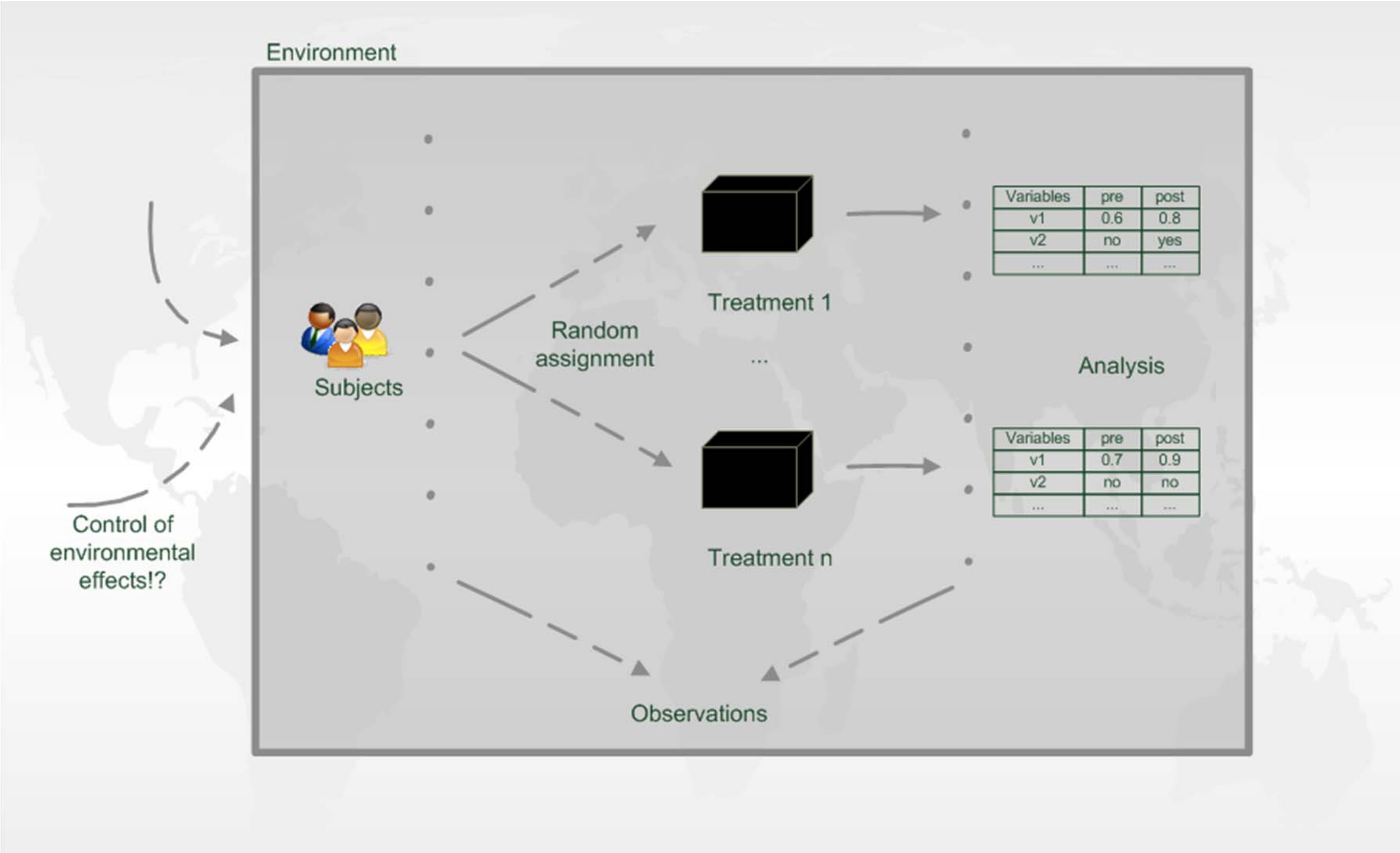
Subject	Online customers, students, historical online sessions, computers, …
Research method	Experiments, quasi-experiments, non-experimental research
Setting	Lab, real-world scenarios

Research methods

- **Experimental vs. non-experimental (observational) research methods**
  - Experiment (test, trial):
    - *"An experiment is a study in which at least one variable is manipulated and units are randomly assigned to different levels or categories of manipulated variable(s)."*

    - Units: users, historic sessions, …
    - Manipulated variable: type of RS, groups of recommended items, explanation strategies …
    - Categories of manipulated variable(s): content-based RS, collaborative RS

Experiment designs



Evaluation in information retrieval (IR)

- **Recommendation is viewed as information retrieval task:**
  - Retrieve (recommend) all items which are predicted to be "good".

- **Ground truth established by human domain experts**

		Reality	
		Actually Good	Actually Bad
Prediction	Rated Good	True Positive (tp)	False Positive (fp)
	Rated Bad	False Negative (fn)	True Negative (tn)

All recommend items

All good items

Metrics: Precision and Recall

- **Precision: a measure of exactness, determines the fraction of relevant items retrieved out of all items retrieved**
  – E.g. the proportion of recommended movies that are actually good

$$Precision = \frac{tp}{tp + fp} = \frac{|good\ movies\ recommended|}{|\text{all recommendations}|}$$

- **Recall: a measure of completeness, determines the fraction of relevant items retrieved out of all relevant items**
  – E.g. the proportion of all good movies recommended

$$Recall = \frac{tp}{tp + fn} = \frac{|good\ movies\ recommended|}{|all\ good\ movies|}$$

F_1 Metric

- **The F_1 Metric attempts to combine Precision and Recall into a single value for comparison purposes.**
  – May be used to gain a more balanced view of performance

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

- **The F_1 Metric gives equal weight to precision and recall**
  – Other F_β metrics weight recall with a factor of β.

Metrics: Rank Score – position matters

For a user:

Actually good
Item 237
Item 899

hit

Recommended (predicted as good)
Item 345
Item 237
Item 187

- **Rank Score extends recall and precision to take the positions of correct items in a ranked list into account**
  - Particularly important in recommender systems as lower ranked items may be overlooked by users

Evaluation in RS

- **Datasets with items rated by users**
  - MovieLens datasets 100K-10M ratings
  - Netflix 100M ratings
- **Historic user ratings constitute ground truth**
- **Metrics measure error rate**
  - Mean Absolute Error (*MAE*) computes the deviation between predicted ratings and actual ratings

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|p_i - r_i|$$

  - Root Mean Square Error (*RMSE*) is similar to *MAE*, but places more emphasis on larger deviation

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(p_i - r_i)^2}$$

Dilemma of establishing ground truth

- **IR measures are frequently applied, however**:

Offline experimentation	Online experimentation
Ratings, transactions	Ratings, feedback
Historic session (not all recommended items are rated)	Live interaction (all recommended items are rated)
Ratings of unrated items unknown, but interpreted as “bad” (default assumption, user tend to rate only good items)	“Good/bad” ratings of not recommended items are unknown
If default assumption does not hold: True positives may be too small False negatives may be too small	False/true negatives cannot be determined
Precision may increase Recall may vary	Precision ok Recall questionable

Results from offline experimentation have limited predictive power for online user behavior.

Offline experimentation

- **Netflix competition**
  - Web-based movie rental
  - Prize of $1,000,000 for accuracy improvement (RMSE) of 10% compared to own Cinematch system.

- **Historical dataset**
  - ~480K users rated ~18K movies on a scale of 1 to 5
  - ~100M ratings
  - Last 9 ratings/user withheld
    - Probe set – for teams for evaluation
    - Quiz set – evaluates teams’ submissions for leaderboard
    - Test set – used by Netflix to determine winner

Online experimentation

- **Effectiveness of different algorithms for recommending cell phone games**
  [Jannach, Hegelich 09]

- **Involved 150,000 users on a commercial mobile internet portal**

- **Comparison of recommender methods**

Details and results

- **Recommender variants included:**
  - Item-based collaborative filtering
  - SlopeOne (also collaborative filtering)
  - Content-based recommendation
  - Hybrid recommendation
  - Top-rated items
  - Top-sellers

  non-personalized

- **Findings:**
  - Personalized methods increased sales up to 3.6% compared to non-personalized
  - Choice of recommendation algorithm depends on user situation (e.g. avoid content-based RS in post-sales situation)

Non-experimental research

- **Quasi-experiments**
  - Lack random assignments of units to different treatments

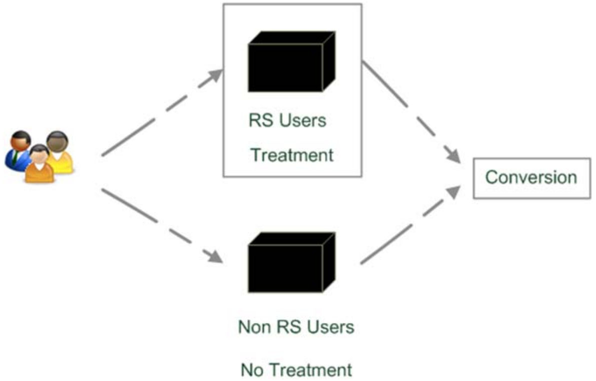
- **Non-experimental / observational research**
  - Surveys / Questionnaires
  - Longitudinal research
    - Observations over long period of time
    - E.g. customer life-time value, returning customers
  - Case studies
  - Focus group
    - Interviews
    - Think aloud protocols

Quasi-experimental

- **SkiMatcher Resort Finder introduced by Ski-Europe.com to provide users with recommendations based on their preferences**

- **Conversational RS**
  - question and answer dialog
  - matching of user preferences with knowledge base

- **Delgado and Davidson evaluated the effectiveness of the recommender over a 4 month period in 2001**
  - Classified as a quasi-experiment as users decide for themselves if they want to use the recommender or not



SkiMatcher Results

	July	August	September	October
Unique Visitors	10,714	15,560	18,317	24,416
• SkiMatcher Users	1,027	1,673	1,878	2,558
• Non-SkiMatcher Users	9,687	13,887	16,439	21,858
Requests for Proposals	272	506	445	641
• SkiMatcher Users	75	143	161	229
• Non-SkiMatcher Users	197	363	284	412
Conversion	2.54%	3.25%	2.43%	2.63%
• SkiMatcher Users	7.30%	8.55%	8.57%	8.95%
• Non-SkiMatcher Users	2.03%	2.61%	1.73%	1.88%
Increase in Conversion	359%	327%	496%	475%

[Delgado and Davidson, ENTER 2002]

Interpreting the Results

- **The nature of this research design means that questions of causality cannot be answered (lack of random assignments), such as**
  - Are users of the recommender systems more likely convert?
  - Does the recommender system itself cause users to convert?

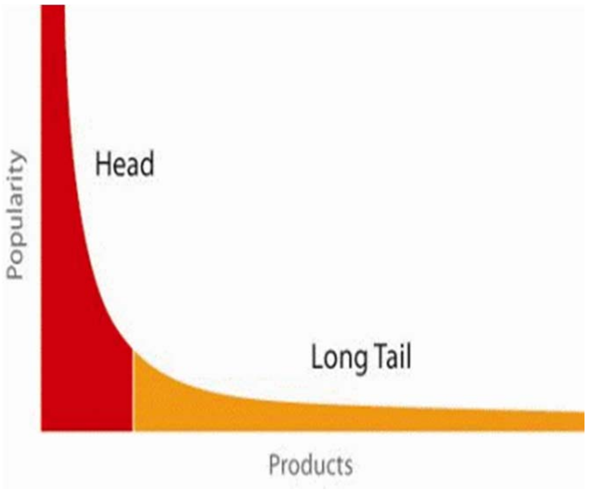
  Some hidden exogenous variable might influence the choice of using RS as well as conversion.

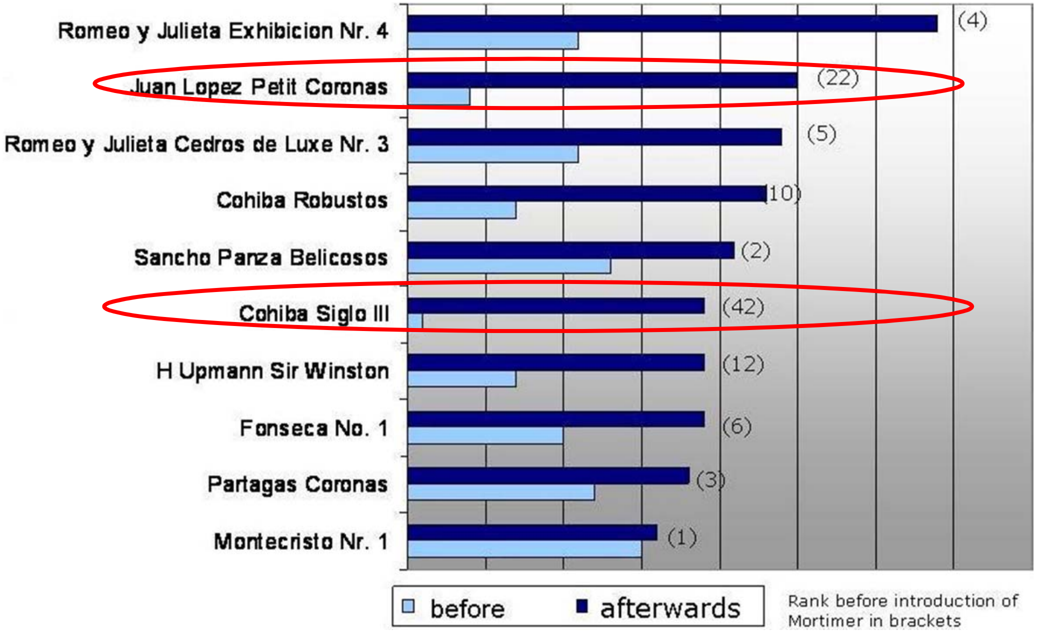
- **However, significant correlation between using the recommender system and making a request for a proposal**

- **Size of effect has been replicated in other domains**
  - Tourism [Jannach et al., JITT 2009]
  - Electronic consumer products

Observational research

- **Increased demand in niches/long tail products**
  - Books ranked above 250.000 represent >29% of sales at Amazon, approx. 2.3 million books [Brynjolfsson et al., Mgt. Science, 2003]
  - Ex post from webshop data [Zanker et al., EC-Web, 2006]





What is popular?

- **Evaluations on historical datasets measuring accuracy**
- **Most popular datasets**
  - Movies (MovieLens, EachMovie, Netflix)
  - Web 2.0 platforms (tags, music, papers, …)
- **Most popular measures for accuracy**
  - Precision/Recall
    - Items are classified as good or bad
  - MAE (Mean Absolute Error), RMSE (Root Mean Squared Error)
    - Items are rated on a given scale
- **Availability of data heavily biases what is done**
  - Tenor at RecSys'09 to foster live experiments
  - Public infrastructures to enable A/B tests

Content-based recommendation

Content-based recommendation

- **While CF – methods do not require any information about the items,**
  - it might be reasonable to exploit such information; and
  - recommend fantasy novels to people who liked fantasy novels in the past
- **What do we need:**
  - some information about the available items such as the genre ("content")
  - some sort of *user profile* describing what the user likes (the preferences)
- **The task:**
  - learn user preferences
  - locate/recommend items that are "similar" to the user preferences

What is the "content"?

- **The genre is actually not part of the content of a book**
- **Most CB-recommendation methods originate from Information Retrieval (IR) field:**
  – goal is to find and rank interesting text documents (news articles, web pages)
  – the item descriptions are usually automatically extracted (important words)
- **Fuzzy border between content-based and "knowledge-based" RS**
- **Here:**
  - classical IR-based methods based on keywords
  - no expert recommendation knowledge involved
  - User profile (preferences) are rather learned than explicitly elicited

Content representation and item similarities

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
...					

Title	Genre	Author	Type	Price	Keywords
...	Fiction, Suspense	Brunonia Barry, Ken Follet, ..	Paperback	25.65	detective, murder, New York

- **Simple approach**
  - Compute the similarity of an unseen item with the user profile based on the keyword overlap (e.g. using the Dice coefficient)
  - sim(b_i, b_j) = $\frac{2 * |keywords(b_i) \cap keywords(b_j)|}{|keywords(b_i)| + |keywords(b_j)|}$
  - Or combine multiple metrics in a weighted approach

Term-Frequency - Inverse Document Frequency (TF-IDF)

- **Simple keyword representation has its problems**
  - in particular when automatically extracted as
    - not every word has similar importance
    - longer documents have a higher chance to have an overlap with the user profile
- **Standard measure: TF-IDF**
  - Encodes text documents in multi-dimensional Euclidian space
    - weighted term vector
  - TF: Measures, how often a term appears (density in a document)
    - assuming that important terms appear more often
    - normalization has to be done in order to take document length into account
  - IDF: Aims to reduce the weight of terms that appear in all documents

TF-IDF

- **Compute the overall importance of keywords**
  - Given a keyword i and a document j

    $TF\text{-}IDF(i,j) = TF(i,j) * IDF(i)$

- **Term frequency (TF)**
  - Let $freq(i,j)$ number of occurrences of keyword *i* in document *j*
  - Let $maxOthers(i,j)$ denote the highest number of occurrences of another keyword of *j*
  - $TF(i,j) = \frac{freq(i,j)}{maxOthers(i,j)}$

- **Inverse Document Frequency (IDF)**
  - N: number of all recommendable documents
  - n(i): number of documents in which keyword *i* appears
  - $IDF(i) = log \frac{N}{n(i)}$

Example TF-IDF representation

	Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
Antony	5.25	3.18	0	0	0	0.35
Brutus	1.21	6.1	0	1	0	0
Caesar	8.59	2.54	0	1.51	0.25	0
Calpurnia	0	1.54	0	0	0	0
Cleopatra	2.85	0	0	0	0	0
mercy	1.51	0	1.9	0.12	5.25	0.88
worser	1.37	0	0.11	4.15	0.25	1.95

Figure taken from http://informationretrieval.org

More on the vector space model

- **Vectors are usually long and sparse**
- **Improvements**
  - remove stop words ("a", "the", ..)
  - use stemming
  - size cut-offs (only use top n most representative words, e.g. around 100)
  - use additional knowledge, use more elaborate methods for feature selection
  - detection of phrases as terms (such as United Nations)
- **Limitations**
  - semantic meaning remains unknown
  - example: usage of a word in a negative context
    - "there is nothing on the menu that a vegetarian would like.."
- **Usual similarity metric to compare vectors: Cosine similarity (angle)**

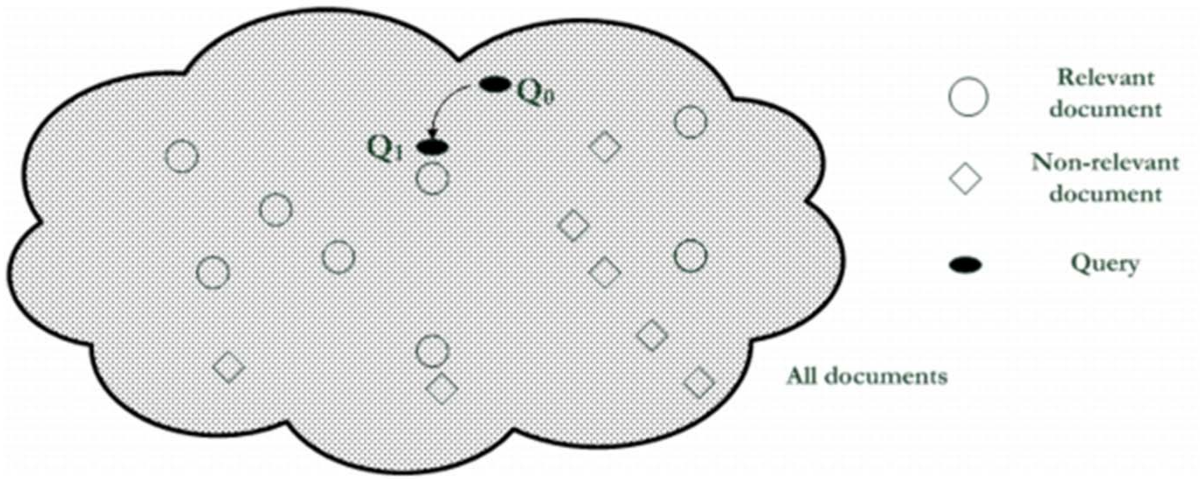
Recommending items

- **Simple method: nearest neighbors**
  - Given a set of documents D already rated by the user (like/dislike)
    - Find the n nearest neighbors of a not-yet-seen item *i* in D
    - Take these ratings to predict a rating/vote for *i*
    - (Variations: neighborhood size, lower/upper similarity thresholds..)
  - Good to model short-term interests / follow-up stories
  - Used in combination with method to model long-term preferences
- **Query-based retrieval: Rocchio's method**
  - The SMART System: Users are allowed to rate (relevant/irrelevant) retrieved documents (feedback)
  - The system then learns a prototype of relevant/irrelevant documents
  - Queries are then automatically extended with additional terms/weight of relevant documents

Rocchio details

- Document collections D^+ and D^-
- α, β, γ used to fine-tune the feedback
- often only positive feedback is used

$$Q_{i+1} = \alpha * Q_i + \beta(\frac{1}{|D^+|}\sum_{d^+ \in D^+} d^+) - \gamma(\frac{1}{|D^-|}\sum_{d^- \in D^-} d^-)$$



Probabilistic methods

- **Recommendation as classical text classification problem**
  - long history of using probabilistic methods
- **Simple approach:**
  - 2 classes: hot/cold
  - simple Boolean document representation
  - calculate probability that document is hot/cold based on Bayes theorem

Doc-ID	recommender	intelligent	learning	school	**Label**
1	1	1	1	0	**1**
2	0	0	1	1	**0**
3	1	1	0	0	**1**
4	1	0	1	1	**1**
5	0	0	0	1	**0**
6	1	1	0	0	**?**

$$\begin{aligned} P(X|Label=1) &= P(recommender=1|Label=1) \times \\ &\quad P(intelligent=1|Label=1) \times \\ &\quad P(learning=0|Label=1) \times P(school=0|Label=1) \\ &= 3/3 \times 2/3 \times 1/3 \times 2/3 \\ &\approx 0.149 \end{aligned}$$

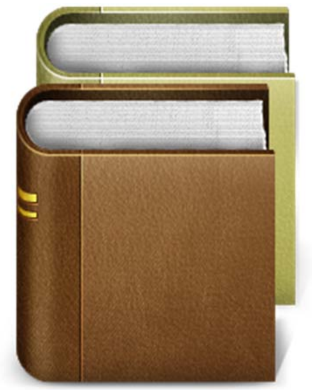
Improvements

- **Side note: Conditional independence of events does in fact not hold**
  - "New York", "Hong Kong"
  - Still, good accuracy can be achieved
- **Boolean representation simplistic**
  - positional independence assumed
  - keyword counts lost
- **More elaborate probabilistic methods**
  - e.g., estimate probability of term v occurring in a document of class C by relative frequency of v in all documents of the class
- **Other linear classification algorithms (machine learning) can be used**
  - Support Vector Machines, ..
- **Use other information retrieval methods (used by search engines..)**

Limitations of content-based recommendation methods

- **Keywords alone may not be sufficient to judge quality/relevance of a document or web page**
  - up-to-dateness, usability, aesthetics, writing style
  - content may also be limited / too short
  - content may not be automatically extractable (multimedia)
- **Ramp-up phase required**
  - Some training data is still required
  - Web 2.0: Use other sources to learn the user preferences
- **Overspecialization**
  - Algorithms tend to propose "more of the same"
  - Or: too similar news items

Knowledge-Based Recommender Systems



Knowledge-Based Recommendation I

- **Explicit domain knowledge**
  - Sales knowledge elicitation from domain experts
  - System mimics the behavior of experienced sales assistant
  - Best-practice sales interactions
  - Can guarantee “correct” recommendations (determinism) with respect to expert knowledge

- **Conversational interaction strategy**
  - Opposed to one-shot interaction
  - Elicitation of user requirements
  - Transfer of product knowledge (“educating users”)

Knowledge-Based Recommendation II

- **Different views on "knowledge"**
  - Similarity functions
    - Determine matching degree between query and item (case-based RS)
  - Utility-based RS
    - E.g. MAUT – Multi-attribute utility theory
  - Logic-based knowledge descriptions (from domain expert)
    - E.g. Hard and soft constraints

- **Hybridization**
  - E.g. merging explicit knowledge with community data
  - Can ensure some policies based on e.g. availability, user context or profit margin

Constraint-based recommendation I

- **A knowledge-based RS formulated as constraint satisfaction problem**

$$CSP\,(X_I \cup X_U, D, SRS \cup KB \cup I)$$

- **Def.**
  - X_I, X_U: Variables describing items and user model with domain D (e.g. lower focal length, purpose)
  - KB: Knowledge base comprising constraints and domain restrictions (e.g. **IF** purpose=“*on travel*” **THEN** lower focal length < *28mm*)
  - SRS: Specific requirements of a user (e.g. purpose = “*on travel*”)
  - I: Product catalog (e.g. (id=1 ∧ lfl = *28mm)* ∨ (id=2 ∧ lfl= *35mm*) ∨ …)

- **Solution: Assignment tuple θ assigning values to all variables X_I**

$s.t.\ SRS \cup KB \cup I \cup \theta$ **is satisfiable.**

Constraint-based recommendation II

- **BUT: What if no solution exists?**
  - $KB \cup I$ not satisfiable → debugging of knowledge base
  - $SRS \cup KB \cup I$ not satisfiable but $KB \cup I$ satisfiable → debugging of user requirements

- **Application of model-based diagnosis for debugging user requirements**
  - Diagnoses: $(SRS \setminus \Delta) \cup KB \cup I$ is satisfiable
  - Repairs: $(SRS \setminus \Delta) \cup \Delta_{repair} \cup KB \cup I$ is satisfiable
  - Conflict sets: $CS \subseteq SRS : CS \cup KB \cup I$ not satisfiable

Example: find minimal relaxations (minimal diagnoses)

Knowledge Base:

	LHS	RHS
C1	TRUE	Brand = Brand pref.
C2	Motives = *Landscape*	Low. foc. Length =< *28*
C3	TRUE	Price =< Max. cost

Product catalogue:

Powershot XY	
Brand	*Canon*
Lower focal length	*35*
Upper focal length	*140*
Price	*420 EUR*

Lumix	
Brand	*Panasonic*
Lower focal length	*28*
Upper focal length	*112*
Price	*319 EUR*

Current user:

		User model (SRS)	
CS1	R1	Motives	*Landscape*
	R2	Brand preference	*Canon*
CS2	R3	Max. cost	*350 EUR*

Diagnoses: $\Delta_1 = \{R2\}, \Delta_2 = \{R1, R3\}$

Ask user

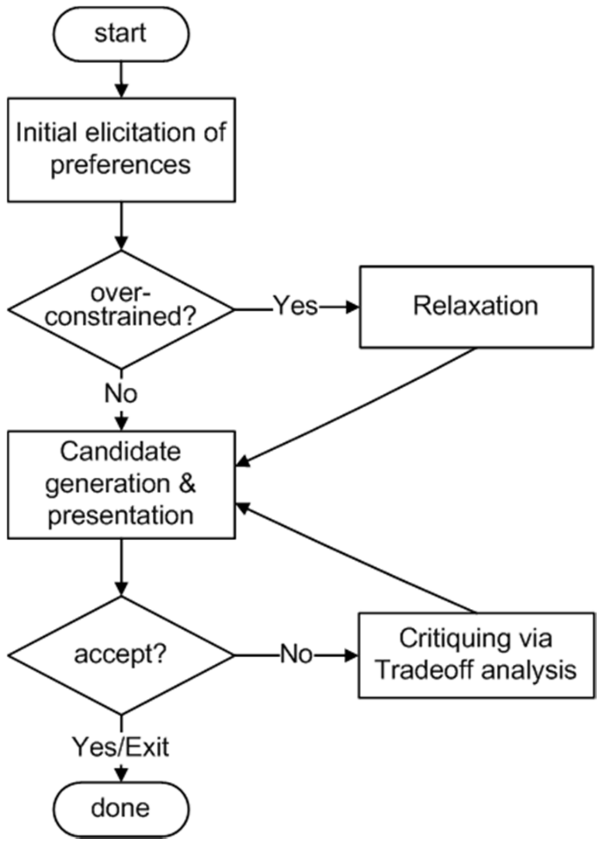
- **Computation of minimal revisions of requirements**
  - Optionally guided by some predefined weights or past community behavior
  - Do you want to relax your brand preference?
    - Accept *Panasonic* instead of *Canon* brand
  - Or is photographing landscapes with a wide-angle lens and maximum cost less important?
    - Lower focal length > 28mm and Price > 350 EUR
- **Be aware of possible revisions** (e.g. age, family status, …)

Constraint-based recommendation III

- **More variants of recommendation task**
  - Find "diverse" sets of items
    - Notion of similarity/dissimilarity
    - Idea that users navigate a product space
    - If recommendations are more diverse than users can navigate via critiques on recommended "entry points" more efficiently (less steps of interaction)
  - Bundling of recommendations
    - Find item bundles that match together according to some knowledge
      - E.g. travel packages, skin care treatments or financial portfolios
      - RS for different item categories, CSP restricts configuring of bundles

Conversational strategies

- **Process consisting of multiple conversational moves**
  - Resembles natural sales interactions
  - Not all user requirements known beforehand
  - Customers are rarely satisfied with the initial recommendations
- **Different styles of preference elicitation:**
  - Free text query interface
  - Asking technical/generic properties
  - Images / inspiration
  - Proposing and Critiquing



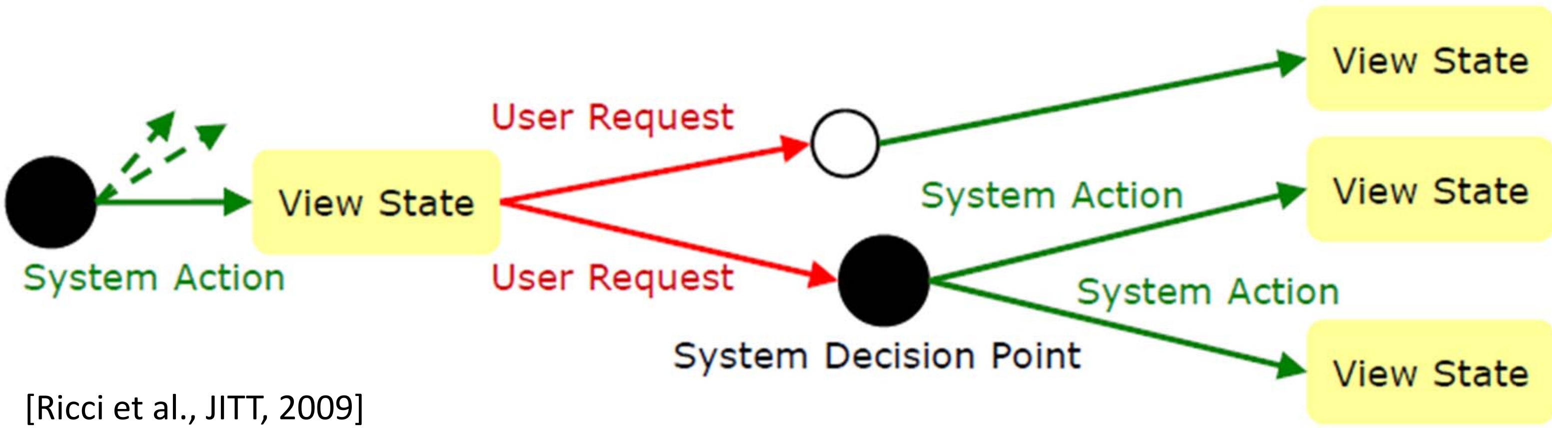
Example: critiquing

- **Similarity-based navigation in item space**

- **Compound critiques**
  - More efficient navigation than with unit critiques

Example: adaptive strategy selection

- **State model, different actions possible**
  - Propose item, ask user, relax/tighten result set,…



[Ricci et al., JITT, 2009]

Limitations of knowledge-based recommendation methods

- **Cost of knowledge acquisition**
  - From domain experts
  - From users
  - From web resources

- **Accuracy of preference models**
  - Very fine granular preference models require many interaction cycles with the user or sufficient detailed data about the user
  - Preferences may depend on each other
  - Collaborative filtering models the preference of a user implicitly

- **Instability of preference models**
  - E.g. asymmetric dominance effects and decoy items

Hybridization Strategies

Hybrid recommender systems

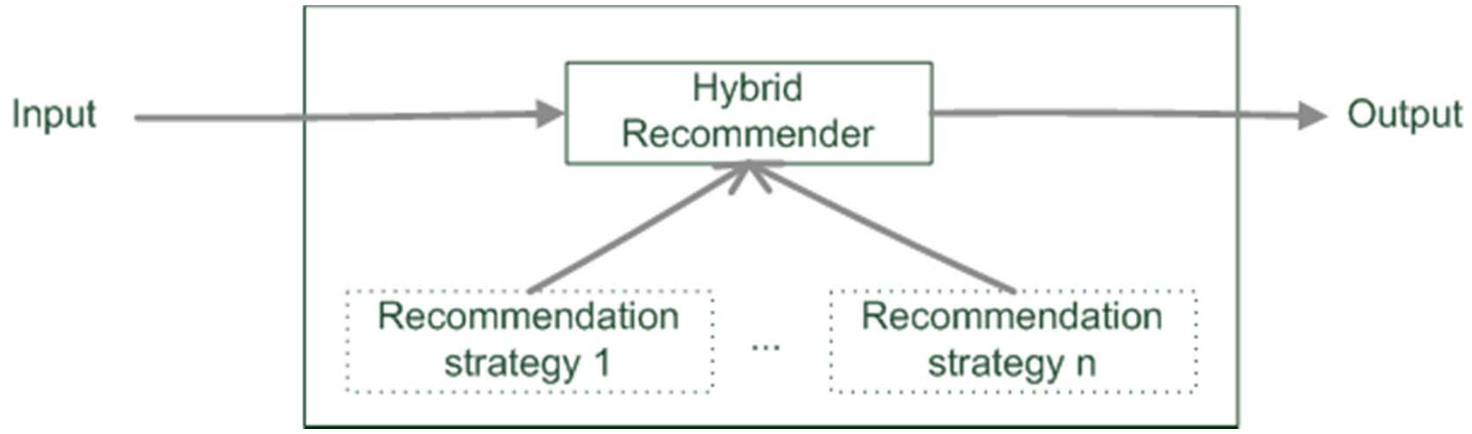
- **All three base techniques are naturally incorporated by a good sales assistance (at different stages of the sales act) but have their shortcomings**

- **Idea of crossing two (or more) species/implementations**
  - *hybrida* [lat.]: denotes an object made by combining two different elements
  - Avoid some of the shortcomings
  - Reach desirable properties not (or only inconsistently) present in parent individuals

- **Different hybridization designs**
  - Monolithic exploiting different features
  - Parallel use of several systems
  - Pipelined invocation of different systems

Monolithic hybridization design

- **Only a single recommendation component**



- **Hybridization is "virtual" in the sense that**
  – Features/knowledge sources of different paradigms are combined

Monolithic hybridization designs: Feature combination

- **"Hybrid" user features:**
  - Social features: Movies liked by user
  - Content features: Comedies liked by user, dramas liked by user
  - Hybrid features: users who like many movies that are comedies, …

  - *"the common knowledge engineering effort that involves inventing good features to enable successful learning"* [BHC98]

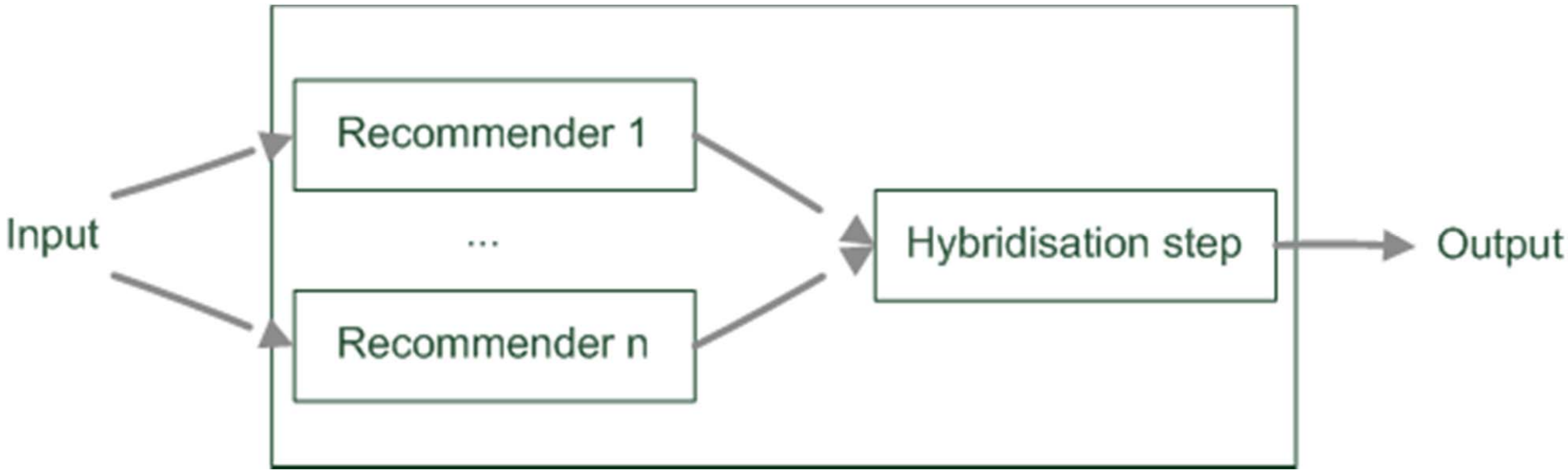
Monolithic hybridization designs: Feature augmentation

- **Content-boosted collaborative filtering [MMN02]**
  - Based on content features additional ratings are created
  - E.g. Alice likes Items 1 and 3 (unary ratings)
    - Item7 is similar to 1 and 3 by a degree of 0,75
    - Thus Alice likes Item7 by 0,75
  - Item matrices become less sparse

- **Recommendation of research papers [TMA+04]**
  - Citations interpreted as collaborative recommendations
  - Integrated in content-based recommendation method

Parallelized hybridization design

- **Output of several existing implementations combined**
- **Least invasive design**
- **Weighting or voting scheme applied**
  - Weights can be learned dynamically



Parallelized hybridization design: Switching

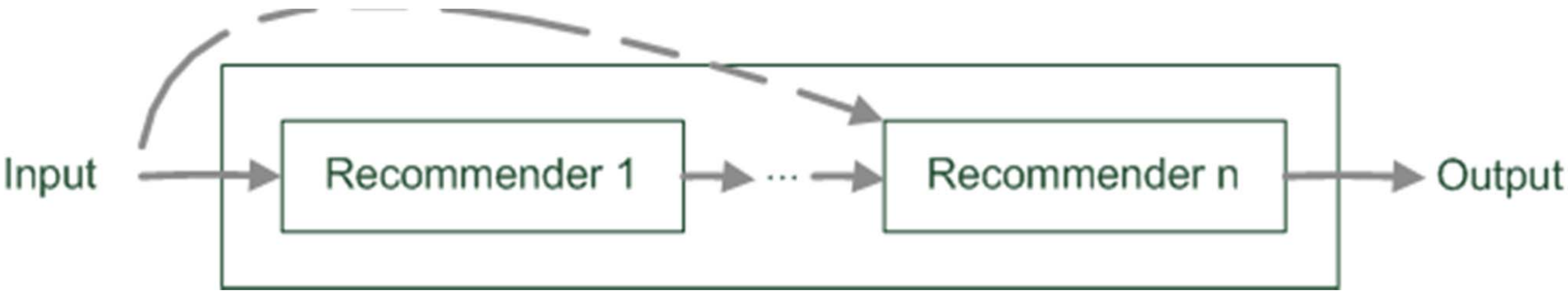
- **Special case of dynamic weights (all weights except one are 0)**

- **Requires an oracle that decides which recommender is used**

- **Example:**
  - Ordering on recommenders and switch based on some quality criteria:
    E.g. if too few ratings in the system, use knowledge-based, else collaborative
  - More complex conditions based on contextual parameters, apply classification techniques

Pipelined hybridization designs

- **One recommender system pre-processes some input for the subsequent one**
  - Cascade
  - Meta-level
- **Refinement of recommendation lists (cascade)**
- **Learning of model (e.g. collaborative knowledge-based meta-level)**



Pipelined hybridization designs: Cascade

Recommender 1		
Item1	0.5	1
Item2	0	
Item3	0.3	2
Item4	0.1	3
Item5	0	

Recommender 2		
Item1	0.8	2
Item2	0.9	1
Item3	0.4	3
Item4	0	
Item5	0	

Recommender cascaded (rec1, rec2)		
Item1	0,80	1
Item2	0,00	
Item3	0,40	2
Item4	0,00	
Item5	0,00	

- **Recommendation list is continually reduced**
- **First recommender excludes items**
  - Remove absolute no-go items (e.g. knowledge-based)
- **Second recommender assigns score**
  - Ordering and refinement (e.g. collaborative)

Pipelined hybridization designs: Meta-level

- **Successor exploits a model Δ built by predecessor**

$$rec_{meta-level}(u,i) = rec_n(u,i,\Delta_{rec_{n-1}})$$

- $\boldsymbol{\Delta_{rec_{n-1}}}$ **is model built by RS_{n-1} exploited by RS_n**
- **Examples:**
  - Fab: content-based, collaborative recommendation [BS97]
    - Online news domain
    - Contend based recommender builds user models based on weighted term vectors
    - Collaborative filtering identifies similar peers based on weighted term vectors but makes recommendations based on ratings
  - Collaborative, constraint-based meta-level RS
    - Collaborative filtering identifies similar peers
    - A constraint base is learned by exploiting the behavior of similar peers
    - Learned constraints are employed to compute recommendations

Limitations and success of hybridization strategies

- **Only few works that compare strategies from the meta-perspective**
  - For instance, [Burke02]
  - Most datasets do not allow to compare different recommendation paradigms
    - i.e. ratings, requirements, item features, domain knowledge, critiques rarely available in a single dataset
  - Thus few conclusions that are supported by empirical findings
    - Monolithic: some preprocessing effort traded-in for more knowledge included
    - Parallel: requires careful matching of scores from different predictors
    - Pipelined: works well for two antithetic approaches
- **Netflix competition – "stacking" recommender systems**
  - Weighted design based on >100 predictors – recommendation functions
  - Adaptive switching of weights based on user model, parameters (e.g. number of ratings in one session)

Explanations in recommender systems

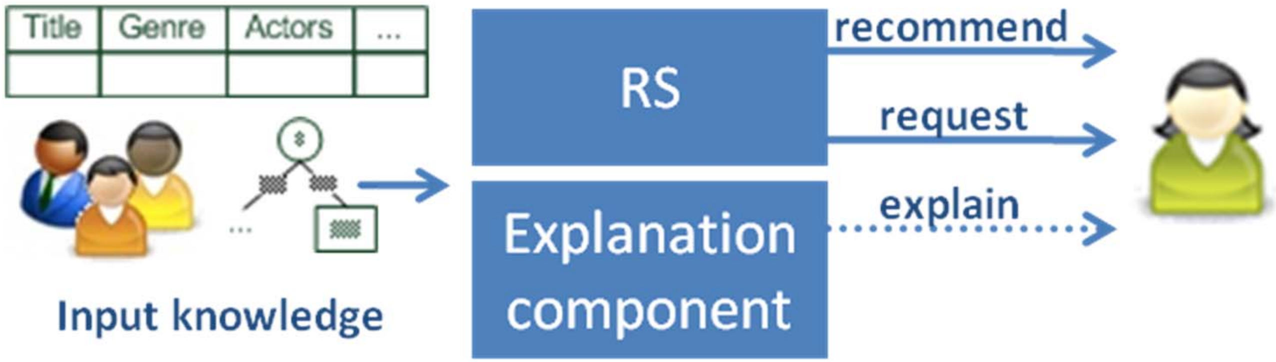
Explanations in recommender systems

Motivation

- “The digital camera *Profishot* is a must-buy for you because . . . .”
- Why should recommender systems deal with explanations at all?
- The answer is related to the two parties providing and receiving recommendations:
  - A selling agent may be interested in promoting particular products
  - A buying agent is concerned about making the right buying decision

Explanations in recommender systems

Additional information to explain the system's output following some objectives



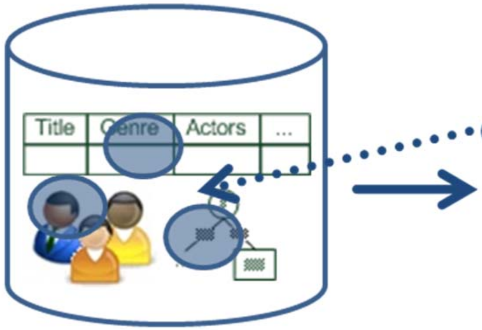
Objectives of explanations

- Transparency
- Validity
- Trustworthiness
- Persuasiveness
- Effectiveness
- Efficiency
- Satisfaction
- Relevance
- Comprehensibility
- Education

Explanations in general

- ***How?*** **and** ***Why?*** **explanations in expert systems**
- **Form of abductive reasoning**
  - Given: $KB \vDash_{RS} i$ (item i is recommended by method RS)
  - Find $KB' \subseteq KB$ s.t. $KB' \vDash_{RS} i$
- **Principle of succinctness**
  - Find smallest subset of $KB' \subseteq KB$ s.t. $KB' \vDash_{RS} i$

    i.e. for all $KB'' \subset KB'$ holds $KB'' \nvDash_{RS} i$
- **But additional filtering**
  - Some parts relevant for deduction, might be obvious for humans



Knowledge base

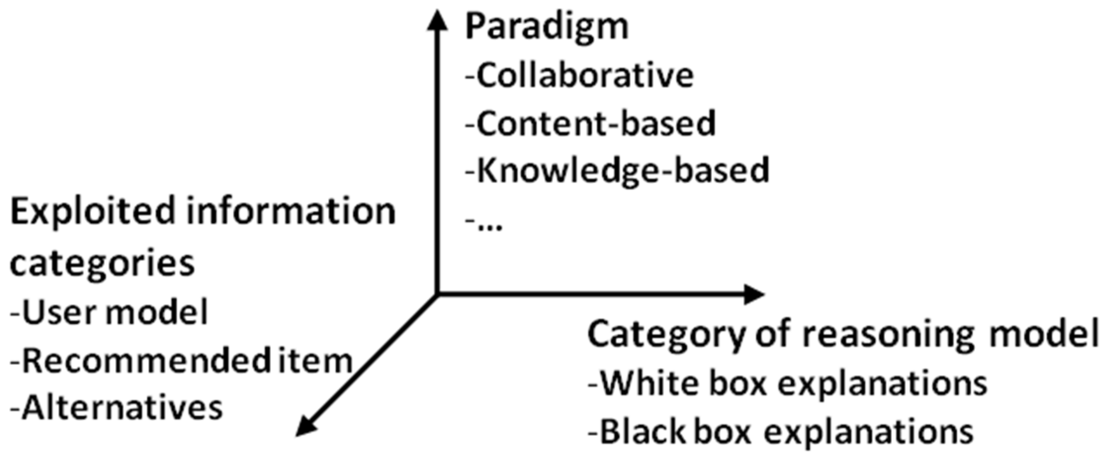
item	score
i7	0.9
i2	0.7
i3	0.3
...	...

[Friedrich & Zanker, AI Magazine, 2011]

Taxonomy for generating explanations in RS

Major design dimensions of current explanation components:

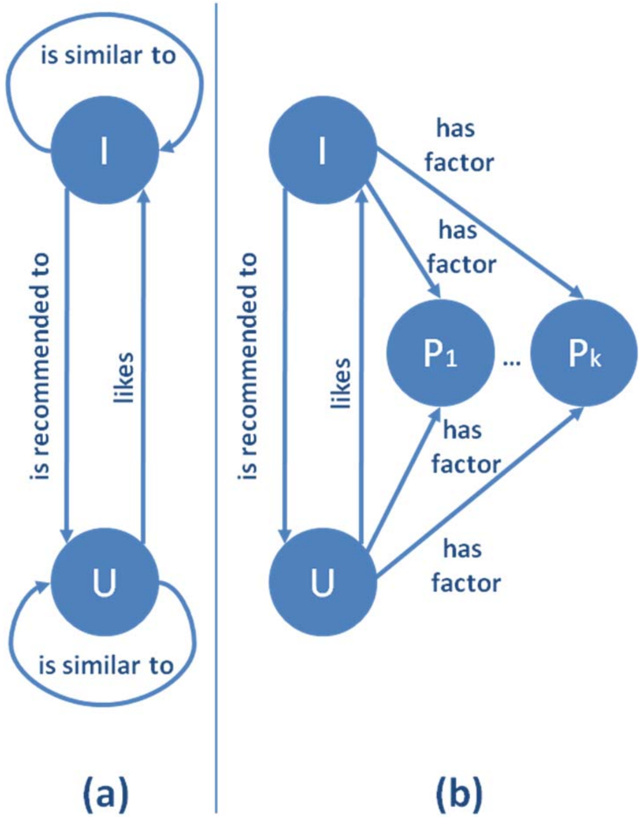
- **Category of reasoning model for generating explanations**
  - White box
  - Black box
- **RS paradigm for generating explanations**
  - Determines the exploitable semantic relations
- **Information categories**



Archetypes of KB

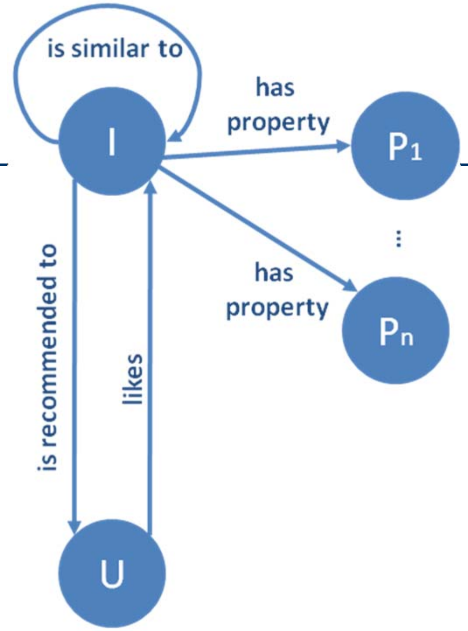
- **Classes of objects**
  - Users
  - Items
  - Properties
- **N-ary relations between them**

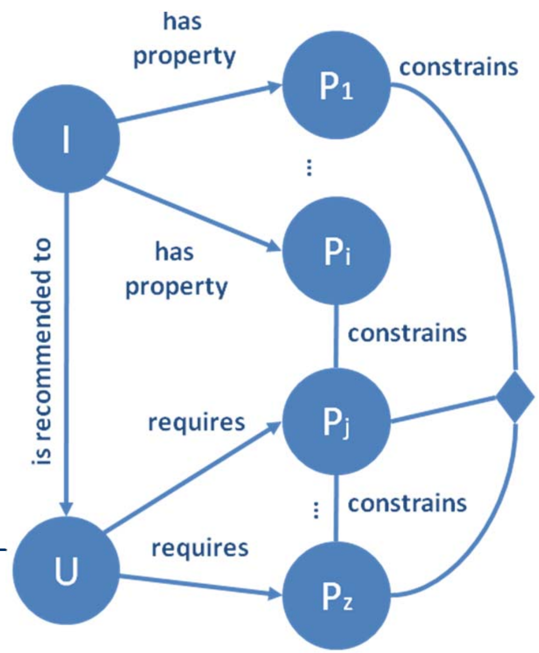
- **Collaborative filtering**
  - Neighborhood based CF (a)
  - Matrix factorization (b)
    - Introduces additional factors as proxies for determining similarities



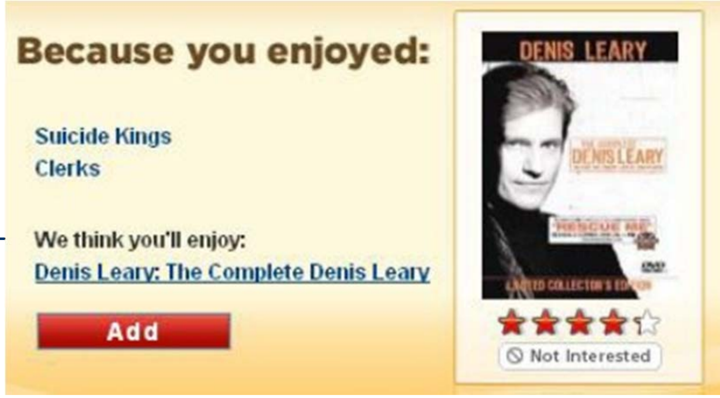
Archetypes of KB

- **Content-based**
  - Properties characterizing items
  - TF*IDF model
- **Knowledge based**
  - Properties of items
  - Properties of user model
  - Additional mediating domain concepts

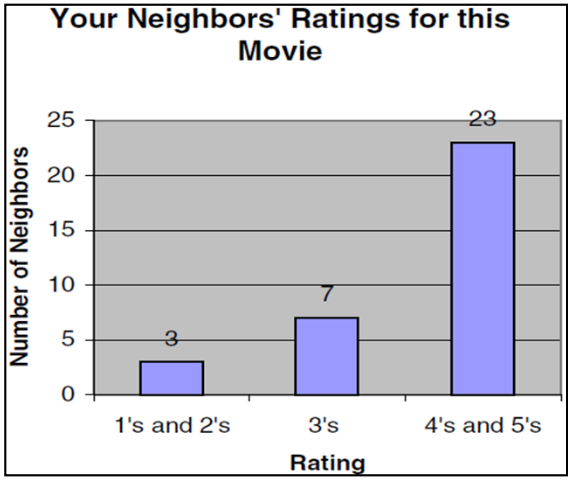




- **Similarity between items**



- **Similarity between users**



- **Tags**
  - Tag relevance (for item)
  - Tag preference (of user)

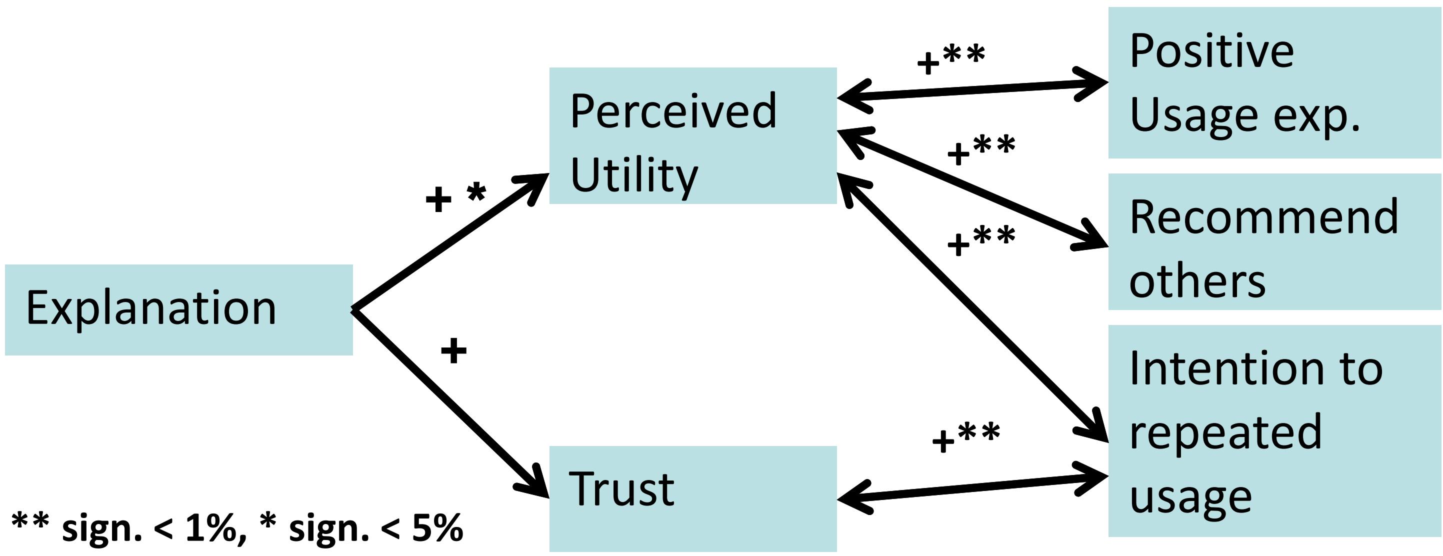
It offers services for families with small children, such as X, Y and Z.

It is a spa resort of medium size offering around 1000 beds.

The water has favorable properties for X, but it is unknown if it also cures Y.

It offers organic food, but no kosher food.

Results from testing the explanation feature

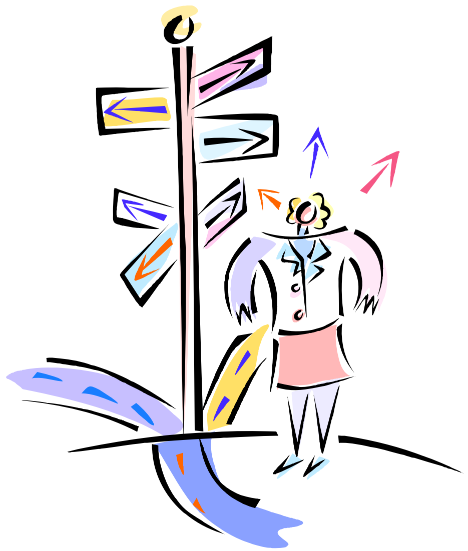


- **Knowledgeable explanations significantly increase the users' perceived utility**
- **Perceived utility strongly correlates with usage intention etc.**

Explanations in recommender systems: Summary

- **There are many types of explanations and various goals that an explanation can achieve**
- **Which type of explanation can be generated depends greatly on the recommender approach applied**
- **Explanations may be used to shape the wishes and desires of customers but are a double-edged sword**
  - On the one hand, explanations can help the customer to make wise buying decisions;
  - On the other hand, explanations can be abused to push a customer in a direction which is advantageous solely for the seller
- **As a result a deep understanding of explanations and their effects on customers is of great interest.**

RS are about Human decision making



Reality check regarding F_1 and accuracy measures for RS

- **Real value lies in increasing conversions**
  - ...and satisfaction with bought items, low churn rate
- **Some reasons why it might be a fallacy to think F_1 on historical data is a good estimate for real conversion:**
  - Recommendation can be self-fulfilling prophecy
    - Users' preferences are not invariant, but can be constructed [ALP03]
  - Position/Rank is what counts (e.g. serial position effects)
    - Actual choices are heavily biased by the item's position [FFG+07]
  - Smaller recommendation sets increase users' confidence in decision making
    - Effect of choice overload - large sets at the same time increase choice difficulty and reduce choice satisfaction [BKW+10]
  - Inclusion of weak (dominated) items increases users' confidence
    - Replacing some recommended items by *decoy* items fosters choice towards the remaining options [TF09]

Bounded rationality

- **Framing and reference dependence, e.g.**
  - Presentation of the decision problem and its recommendations to the user (e.g. gains and losses)
  - Bias towards initial anchor point (in conversational RS)
- **Cognitive consistency theory**
  - Preferences are re-constructed in the course of decision making in order to avoid conflicts
- **Serial position effects**
  - Primacy and recency
- **Decoy effects**
  - Items below pareto frontier
  - Dominance relationships

Implications for RS

- **Preferences cannot be assumed to be stable during RS interaction**
  - Sequence, content and wording matter
- **Preference reasoning**
  - Independence of preferences may not be assumed, impacts the computation of the preference score
- **Presentation/recommendation of items**
  - Serial positions significantly influence the decision
  - Inclusion of decoy items

Which one will the majority select?

Primacy and Decoy effect

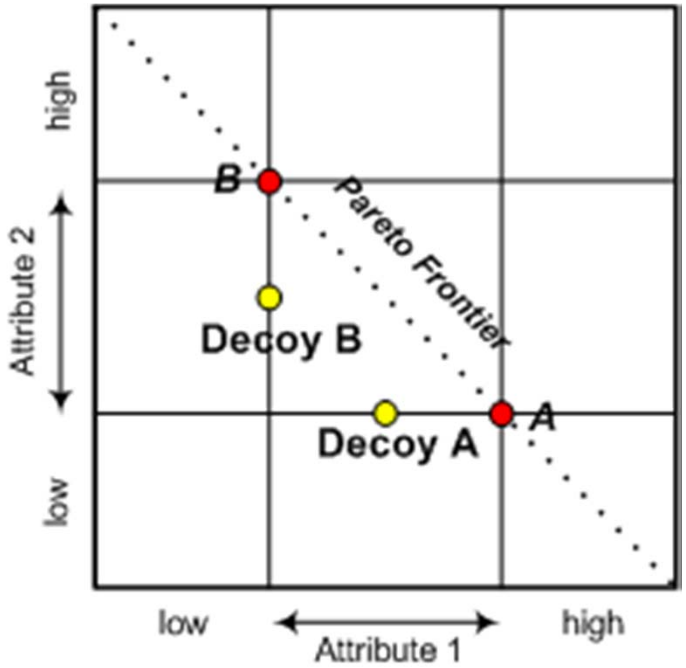
	Item A	Item B	Decoy A
Attribute 1	High	Low	Medium
Attribute 2	Low	High	Low

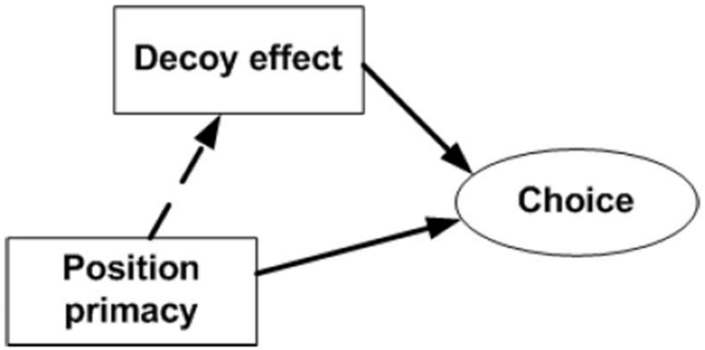
Primacy effect and opposite Decoy effect

	Item B	Item A	Decoy B
Attribute 1	Low	High	Low
Attribute 2	High	Low	Medium

Primacy and Decoy effect

	Item B	Item A	Decoy B
Attribute 2	High	Low	Medium
Attribute 1	Low	High	Low





Outlook

- **Additional topics covered by the book "Recommender Systems - An Introduction"**
  - Case study on the Mobile Internet
  - Attacks on CF Recommender Systems
  - Recommender Systems in the next generation Web (Social Web, Semantic Web)
  - Consumer decision making
  - Recommending in ubiquitous environments
- **"RS research will become much more diverse"**
  - Various forms of feedback mechanisms and preference representation
  - More focus on interfaces, interaction processes, explaining and trust-building
  - Plurality of evaluation methods complementing offline experiments
- **More focus on causal relationships**
  - When, where and how to recommend?
  - Consumer / sales psychology
  - Consumer decision making theories

Thank you for your attention!

Questions?

Questions?

Questions?

Dietmar Jannach
e-Services Research Group
Department of Computer Science
TU Dortmund, Germany
M: dietmar.jannach@tu-dortmund.de
P: +49 231 755 7272

Markus Zanker and Gerhard Friedrich
Intelligent Systems and Business Informatics
Institute of Applied Informatics
University Klagenfurt, Austria
M: firstname.lastname@uni-klu.ac.at
P: +43 463 2700 3705

http://recsys.acm.org

http://www.recommenderbook.net

Recommender Systems – An Introduction by

Dietmar Jannach, Markus Zanker, Alexander Felfernig and Gerhard Friedrich
Cambridge University Press, 2010

References

[Adomavicius & Tuzhilin, IEEE TKDE, 2005] Adomavicius G., Tuzhilin, A. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions, IEEE TKDE, 17(6), 2005, pp.734-749.

[ALP03] Ariely, D., Loewenstein, G., Prelec, D. (2003) "Coherent Arbitrainess": Stable Demand Curves Without Stable Preferences. The Quarterly Journal of Economics, February 2003, 73-105.

[BKW+10] Bollen, D., Knijnenburg, B., Willemsen, M., Graus, M. (2010) Understanding Choice Overload in Recommender Systems. ACM Recommender Systems, 63-70.

[Brynjolfsson et al., Mgt. Science, 2003] Brynjolfsson, E., Hu, Y., Smith, M.: Consumer Surplus in the Digital Economy: Estimating the Value of Increased Product Variety at Online Booksellers, *Management Science*, Vol 49(11), 2003, pp. 1580-1596.

[BS97] Balabanovic, M., Shoham, Y. (1997) Fab: content-based, collaborative recommendation, Communications of the ACM, Vol. 40(3), pp. 66-72.

[FFG+07] Felfernig, A. , Friedrich, G., Gula, B. et al. (2007) Persuasive recommendation: serial position effects in knowledge-based recommender systems. 2nd international conference on Persuasive technology, Springer, 283–294.

[Friedrich& Zanker, AIMag, 2011] Friedrich, G., Zanker, M.: A Taxonomy for Generating Explanations in Recommender Systems. *AI Magazine*, Vol. 32(3), 2011.

[Jannach et al., CUP, 2010] Jannach D., Zanker M., Felfernig, A., Friedrich, G.: Recommender Systems an Introduction, Cambridge University Press, 2010.

[Jannach et al., JITT, 2009] Jannach, D., Zanker, M., Fuchs, M.: Constraint-based recommendation in tourism: A multi-perspective case study, *Information Technology & Tourism*, Vol 11(2), pp. 139-156.

[Jannach, Hegelich 09] Jannach, D., Hegelich K.: A case study on the effectiveness of recommendations in the Mobile Internet, ACM Conference on Recommender Systems, New York, 2009, pp. 205-208

[Ricci et al., JITT, 2009] Mahmood, T., Ricci, F., Venturini, A.: Improving Recommendation Effectiveness by Adapting the Dialogue Strategy in Online Travel Planning. *Information Technology & Tourism*, Vol 11(4), 2009, pp. 285-302.

[Teppan& Felfernig, CEC, 2009] Teppan, E., Felfernig, A.: Asymmetric Dominance- and Compromise Effects in the Financial Services Domain. *IEEE International Conference on E-Commerce and Enterprise Computing*, 2009, pp. 57-64

[TF09] Teppan, E., Felfernig, A. (2009) Impacts of decoy elements on result set evaluations in knowledge-based recommendation. Int. J. Adv. Intell. Paradigms 1, 358–373.

References

[Xiao & Benbasat, MISQ, 2007] Xiao, B., Benbasat, I.: E-Commerce Product Recommendation Agents: Use, Characteristics, and Impact, *MIS Quarterly*, Vol 31(1), pp. 137-209.

[Zanker et al., EC-Web, 2006] Zanker, M., Bricman, M., Gordea, S., Jannach, D., Jessenitschnig, M.: Persuasive online-selling in quality & taste domains, *7th International Conference on Electronic Commerce and Web Technologies*, 2006, pp. 51-60.

[Zanker, RecSys, 2008] Zanker M., A Collaborative Constraint-Based Meta-Level Recommender. *ACM Conference on Recommender Systems,* 2008, pp. 139-146.

[Zanker et al., UMUAI, 2009] Zanker, M., Jessenitschnig, M., Case-studies on exploiting explicit customer requirements in recommender systems, *User Modeling and User-Adapted Interaction*, Springer, 2009, pp.133-166.

[Zanker et al., JITT, 2009] Zanker M., Jessenitschnig M., Fuchs, M.: Automated Semantic Annotations of Tourism Resources Based on Geospatial Data, *Information Technology & Tourism*, Vol 11(4), 2009, pp. 341-354.

[Zanker et al., Constraints, 2010] Zanker M., Jessenitschnig M., Schmid W.: Preference reasoning with soft constraints in constraint-based recommender systems. *Constraints,* Springer, Vol 15(4), 2010, pp. 574-595.