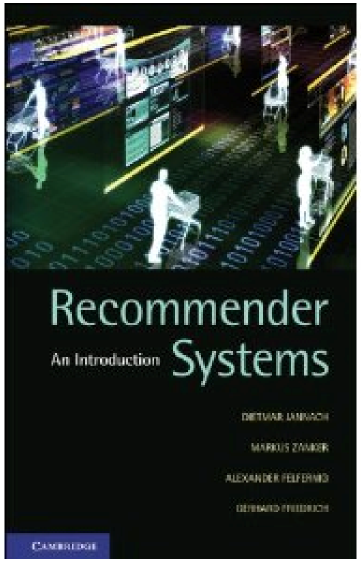
Recommender Systems
推荐系统

Qingcai Chen (Edt.)

Ref:

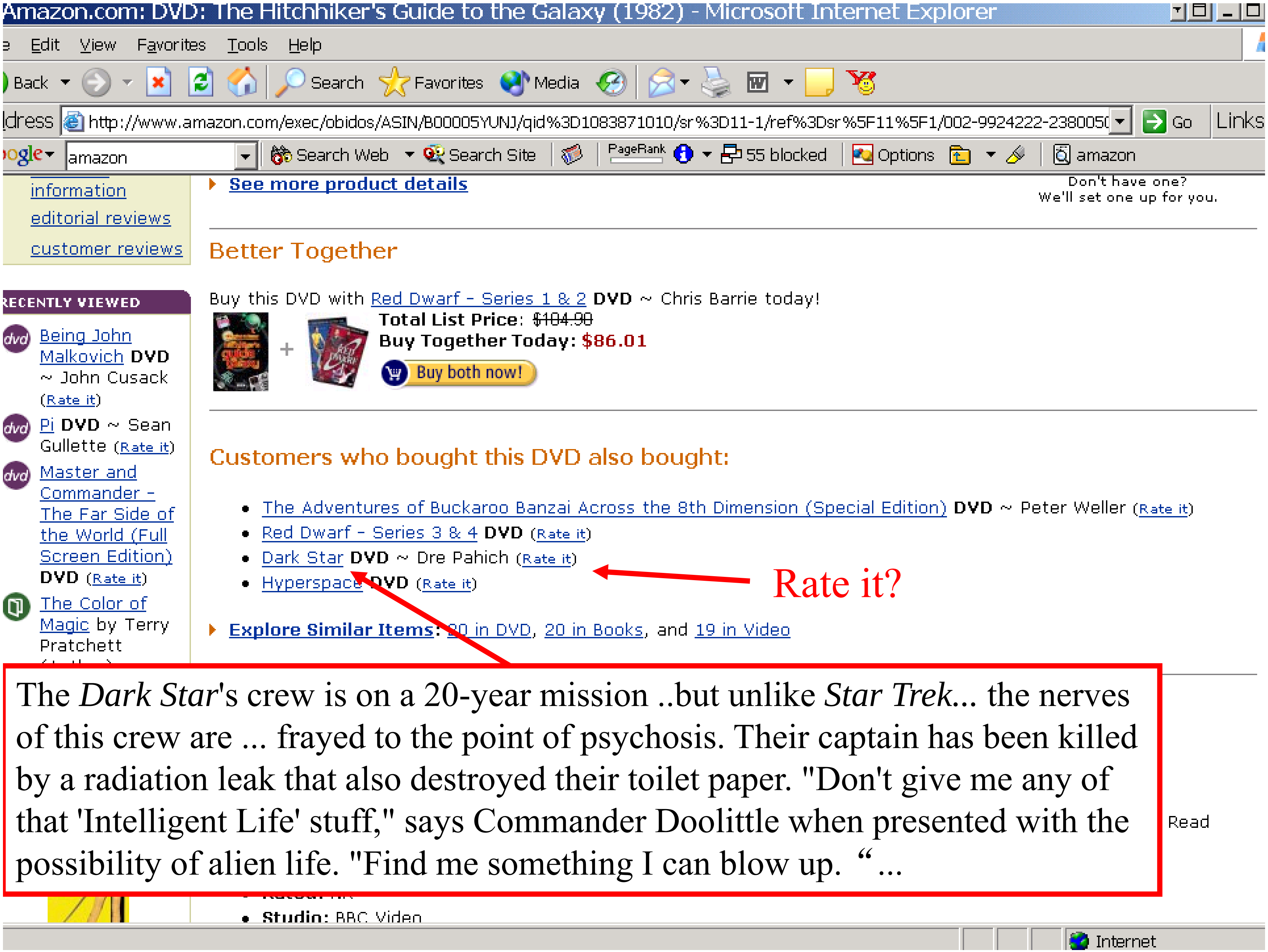
Dietmar Jannach, Gerhard Friedrich, Tutorial on International Joint Conference on Artificial Intelligence, 2020


Recommender Systems
An Introduction
DIETMAR JANNACH
MARKUS ZANKER
ALEXANDER FELFERNIG
GERHARD FRIEDRICH
CAMBRIDGE


Recommender Sy
~ Dietmar Jannach (作者),
分享我的评价 | 天天低价·正品
价格: ¥ 527.30 免运费
配送至: 广东深圳市南山
由亚马逊直接销
全新品 3 售价从 ¥ 527.30
中文图书全场满100元返150元Z券
亚马逊自营
活动期间购
可获得一共
乐器、食品
实惠产品)
点击在线试读!
Recommender Systems
An Introduction
推荐系统
人民邮电出版社


推荐系统 [平装]
~ 詹尼士 (Dietmar Jannach) (作者), Markus Zank
(20 条商品评论) | 天天低价·正品质优
市场价: ¥ 59.00
定价: ¥ 41.30
为您节省: ¥ 17.70 (7折)
配送至: 广东深圳市南山区 现在有货
由中商新华图书专营店直接销售和发货,
送达日期: 后天(3月7日)–下周四(3月13日), 快递
支持礼品卡、一键信用卡支付、银行卡等, 但不支持货
全新品 31 售价从 ¥ 41.00 起


经常一起购买的商品
价格合计: ¥ 90.00
立即购买组合
这些商品由不同的卖家进行配送和销售。 查看发货和库存信息
本商品: 推荐系统 - 詹尼士 (Dietmar Jannach) 平装 ¥ 41.30
机器学习实战 - 哈林顿 (Peter Harrington) 平装 ¥ 48.70
购买此商品的顾客也同时购买
第 1 页, 共 15 页
机器学习实战
哈林顿 (Peter Ha...
(38)
平装
¥ 48.70
数据挖掘十大算法
Xindong Wu (作...
(6)
平装
¥ 31.10
统计思维:程序员数学之概率统计
唐尼 (Allen B.D...
(16)
平装
¥ 22.90
推荐系统实践
项亮 (作者)
(60)
平装
¥ 36.30
机器学习:实用案例解析
康威 (Drew Conw...
(20)
平装
¥ 37.00
迷茫的旅行商:一个无处不在的计算机算法问题(全 ...
库克 (William J...
(9)
平装
¥ 39.20
七周七数据库
雷德蒙 (Eric Red...
(12)
平装
¥ 43.40



Rate it?

The *Dark Star*'s crew is on a 20-year mission ..but unlike *Star Trek*... the nerves of this crew are ... frayed to the point of psychosis. Their captain has been killed by a radiation leak that also destroyed their toilet paper. "Don't give me any of that 'Intelligent Life' stuff," says Commander Doolittle when presented with the possibility of alien life. "Find me something I can blow up. "...


分组
特别关注
名人明星
同事
同学
NLP
公共
展开
3月4日 23:30 来自皮皮时光机
(20) | 转发(61) | 收藏 | 评论(11)
好友赞过的图片
好友 @邱锡鹏 赞过
17 人赞过
好友 @蒋涛CSDN 赞过
33 人赞过
好友 @刘璐_PKU 赞过
261 人赞过
18人赞过
深圳天气
雾不散亲们都开始秀上了哇 好消息是北风增大了些，北部地区的能见度在升高。不过南部沿海雾都还没散涅~
@我_来_自_金_星
今天深圳雾霾很重，就从公司到站台等车那么一会，我就一直咳嗽，连夜赶制了一个简易口罩，有了他，我和我的小伙伴就再也不怕雾霾了[胜利]。没有雾霾的时候还能当头巾，以后万一成名了，还可以用来躲避记者和粉丝，真是一物多用，一举多得啊！话说回来@深圳天气 ，雾霾什么时候结束啊
#露额头才是美女# 25万
#18岁结婚# 394万
#任何事情都不简单# 496
#广州地铁踩踏# 本地 1284万
#温暖昆明# 98万
#都敏俊就是张起灵# 2万
#胖子瘦了都惊艳# 1万
#国产剧vs韩剧# 28万
24小时热点 更多»
可能感兴趣的人 换一换
张俊格
+ 关注
冯志伟文化...等6个好友也关注
刘未鹏pongba
+ 关注
知名博客作者、《暗时...
52nlp等4个好友也关注
找朋友 更多»


电脑、办公 > 电脑整机 > 笔记本
电脑整机
笔记本
超级本
游戏本
平板电脑
平板电脑配件
台式机
服务器/工作站
笔记本配件
电脑配件
外设产品
网络产品
办公打印
办公文仪
服务产品
IT数码限时抢购
剩余 32 小时 51 分 31 秒
抢
双飞燕（A4TECH） N-740X 针光变速鼠 铁灰皮革
抢购价：49.00(8.3折)
热卖推荐
惠普（HP） Pavilion m4-1009tx 14.0英寸笔记本（i5-
特价：¥4199.00
立即抢购
千颂伊同款，更多精彩活动点击抢购
特价：¥4399.00
立即抢购
宏碁（acer） V5-471G-53334G50Dass 14.0英寸超薄
特价：¥3699.00
立即抢购
促销活动
·重磅有料，昂达品牌盛宴送
·芬尚文房，每满99减40元。
·希捷移动硬盘领券立减10元
·开学购平板赢来自星星的旅
·10元抢订四核通话平板！
笔记本 - 商品筛选
品牌： 联想（ThinkPad） 宏碁（acer） 联想 华硕（ASUS） 索尼（SONY） 更多
戴尔（DELL） 惠普（HP） 三星（SAMSUNG） 神舟（HASEE） 东芝（TOSHIBA）
微星（msi） 苹果（Apple） 清华同方（THTF） 雷神（ThundeRobot） 海尔（Haier）
价格： 1000-2999 3000-3499 3500-3999 4000-4499 4500-4999 5000-5999 6000-6999 更多
尺寸： 10.1英寸及以下 11英寸 12英寸 13英寸 14英寸 15英寸 16英寸-17英寸 17英寸以上
触控： 普通触控 变形触控 非触控
更多选项（处理器，厚度，显卡）
排序： 销量 价格 评论数 上架时间
共683个商品 1/19 上一页 下一页
库存： 北京朝阳区管庄 仅显示有货 商品类型： 全部 京东配送 第三方配送
品牌特卖
人气
惊爆价
爆款
调查问卷
返回顶部

Netflix Prize: $1M competition (2006)

On 21 September 2009, the grand prize of US$1,000,000 was given to the team "BellKor's Pragmatic Chaos", a merger of teams "Bellkor in BigChaos" and "Pragmatic Theory", achieved a 10.05% improvement over Cinematch (a Quiz RMSE of 0.8558). BellKor's Pragmatic Chaos team which bested Netflix's own algorithm for predicting ratings by 10.06%.

Introduction

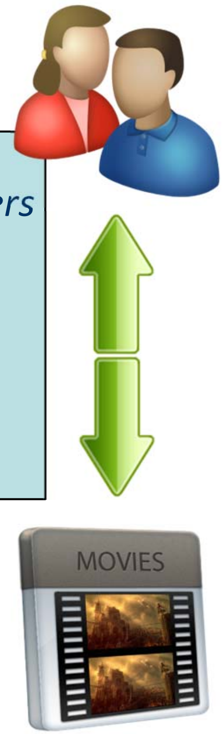
- Problem domain
- Purpose and success criteria
- Paradigms of recommender systems（推荐系统的常见模式）
    - Collaborative Filtering（协同过滤）
    - Content-based Filtering（基于内容的过滤）
    - Knowledge-Based Recommendations（基于知识的推荐）
    - Hybridization Strategies（混合策略）

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 2007[1])



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

(1) Xiao and Benbasat, *E-commerce product recommendation agents: Use, characteristics, and impact*, MIS Quarterly **31** (2007), no. 1, 137–209

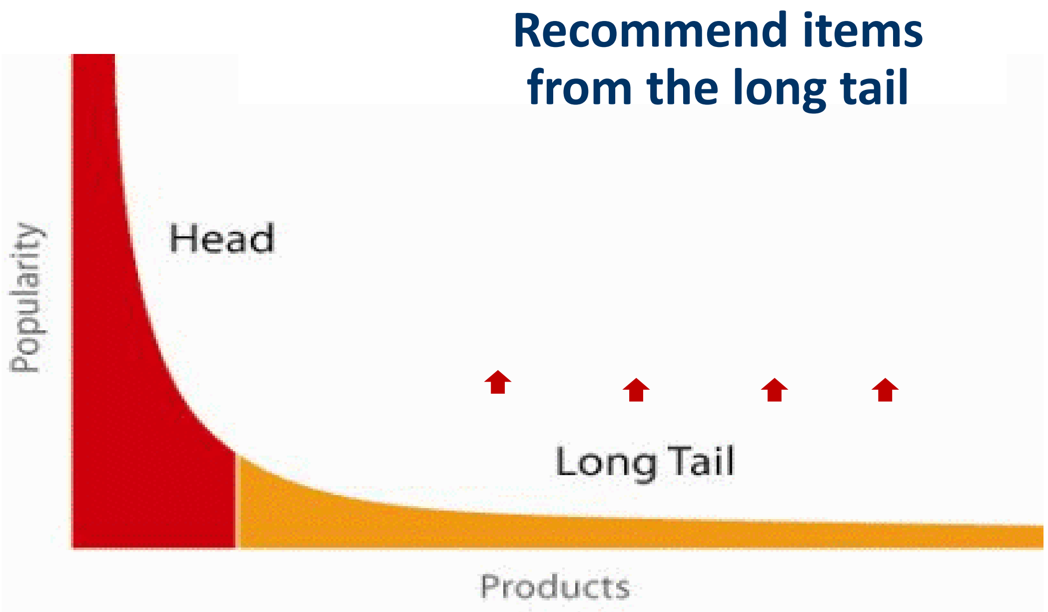
Purpose and success criteria 目的与评价准则 (1)

- **Different perspectives/aspects**
  - Depends on domain and purpose
  - No holistic evaluation scenario exists （缺乏一个全面的评价标准）

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

- **Recommendation perspective**
  - Identify items from the Long Tail
  - Users did not know about existence

When does a RS do its job well?



- "Recommend widely unknown items that users might actually like!"

- 20% of items accumulate 74% of all positive ratings

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 *（转化的角度）**
  - Commercial situations
  - Increase “hit”, “clickthrough”, “lookers to bookers” rates（预订转化率）
  - Optimize sales margins and profit

Recommender systems

- **RS seen as a function**
- **Given:**
  - User model (e.g. ratings, preferences, demographics（人口统计学特征）, 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. 2008[1])

(1) Manning, Raghavan, and Schütze, *Introduction to information retrieval*, Cambridge University Press, 2008

Paradigms of recommender systems

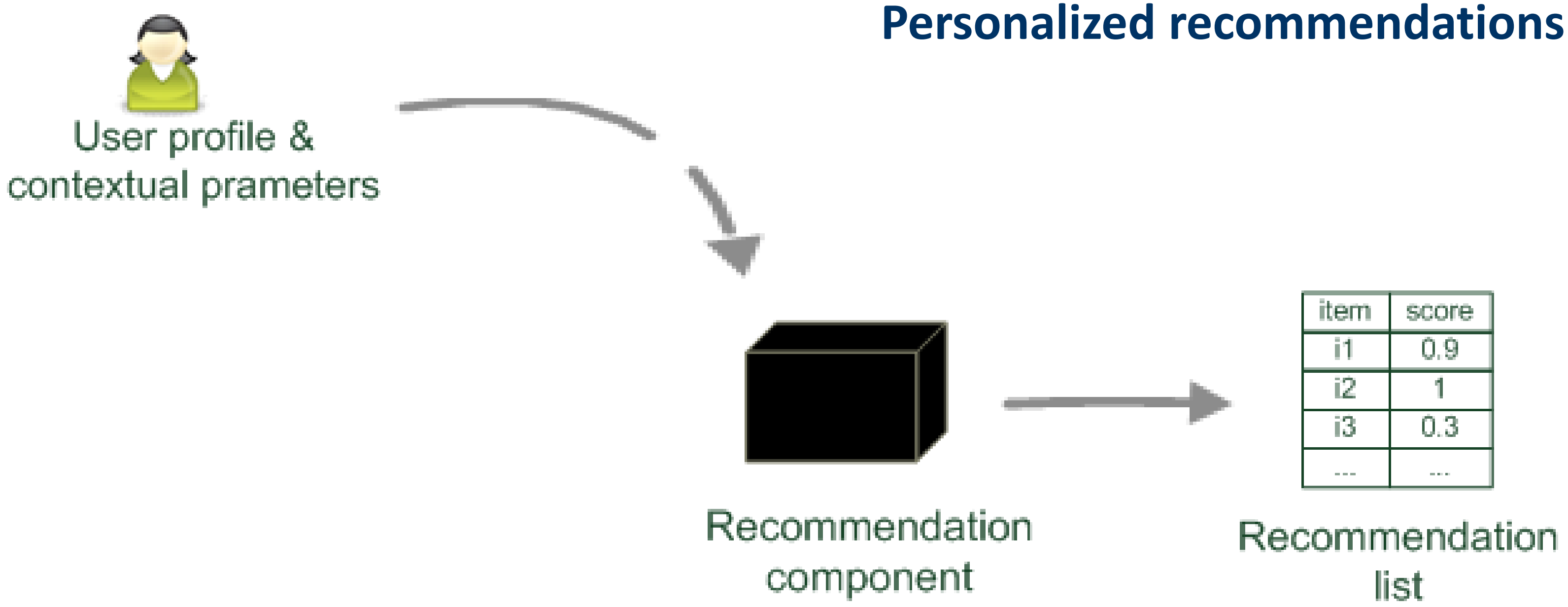
Recommender systems reduce information overload by estimating relevance

item	score
i1	0.9
i2	1
i3	0.3
...	...

Recommendation
list

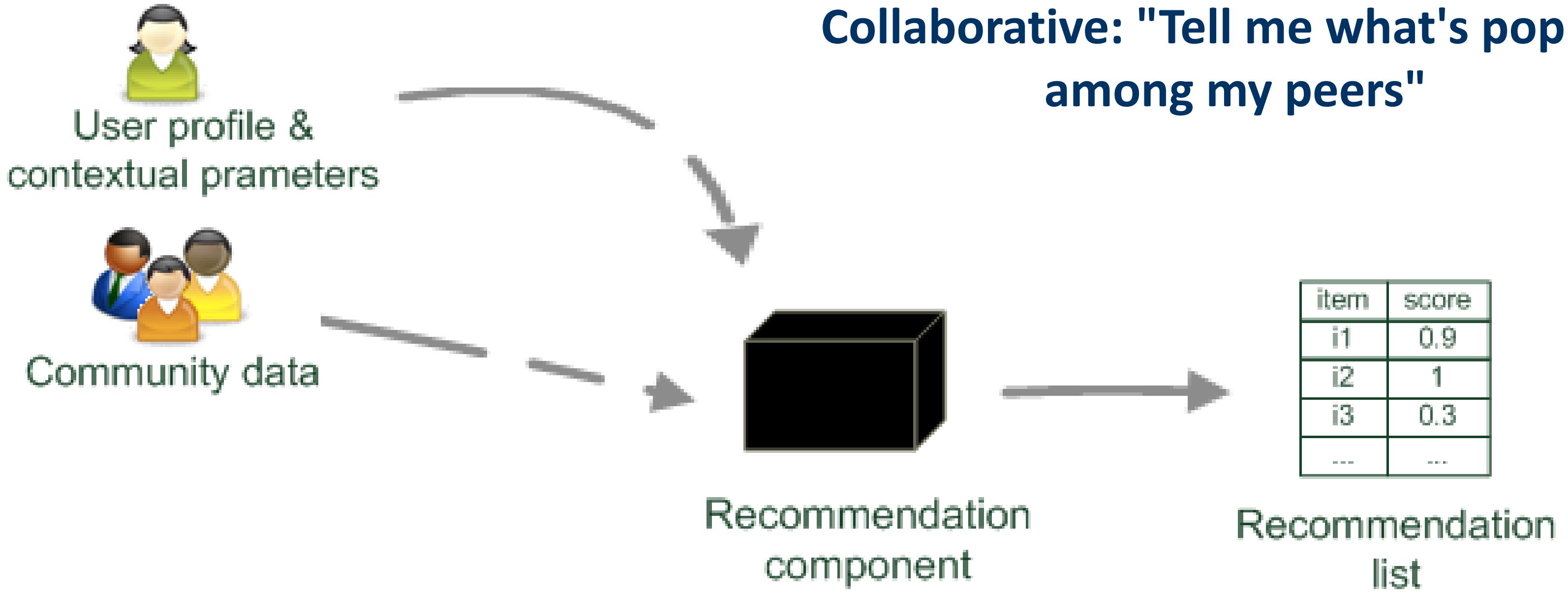
Paradigms of recommender systems

Personalized recommendations



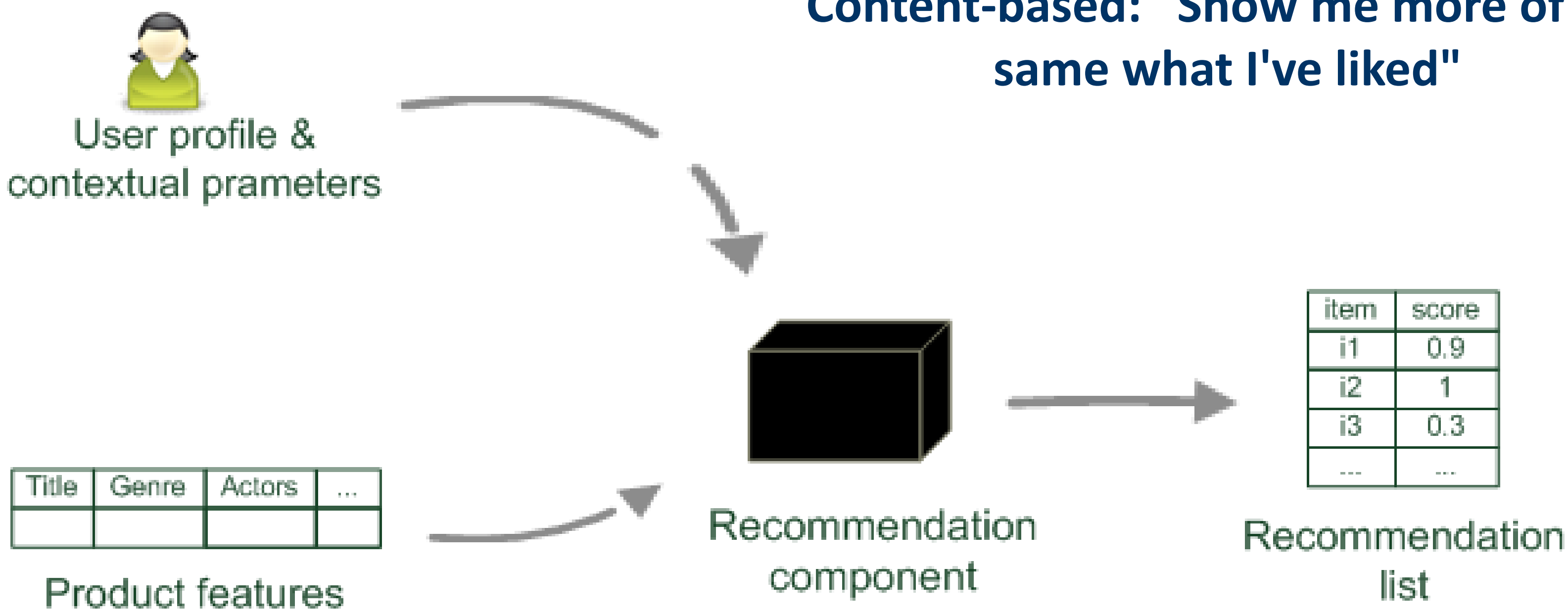
Paradigms of recommender systems

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



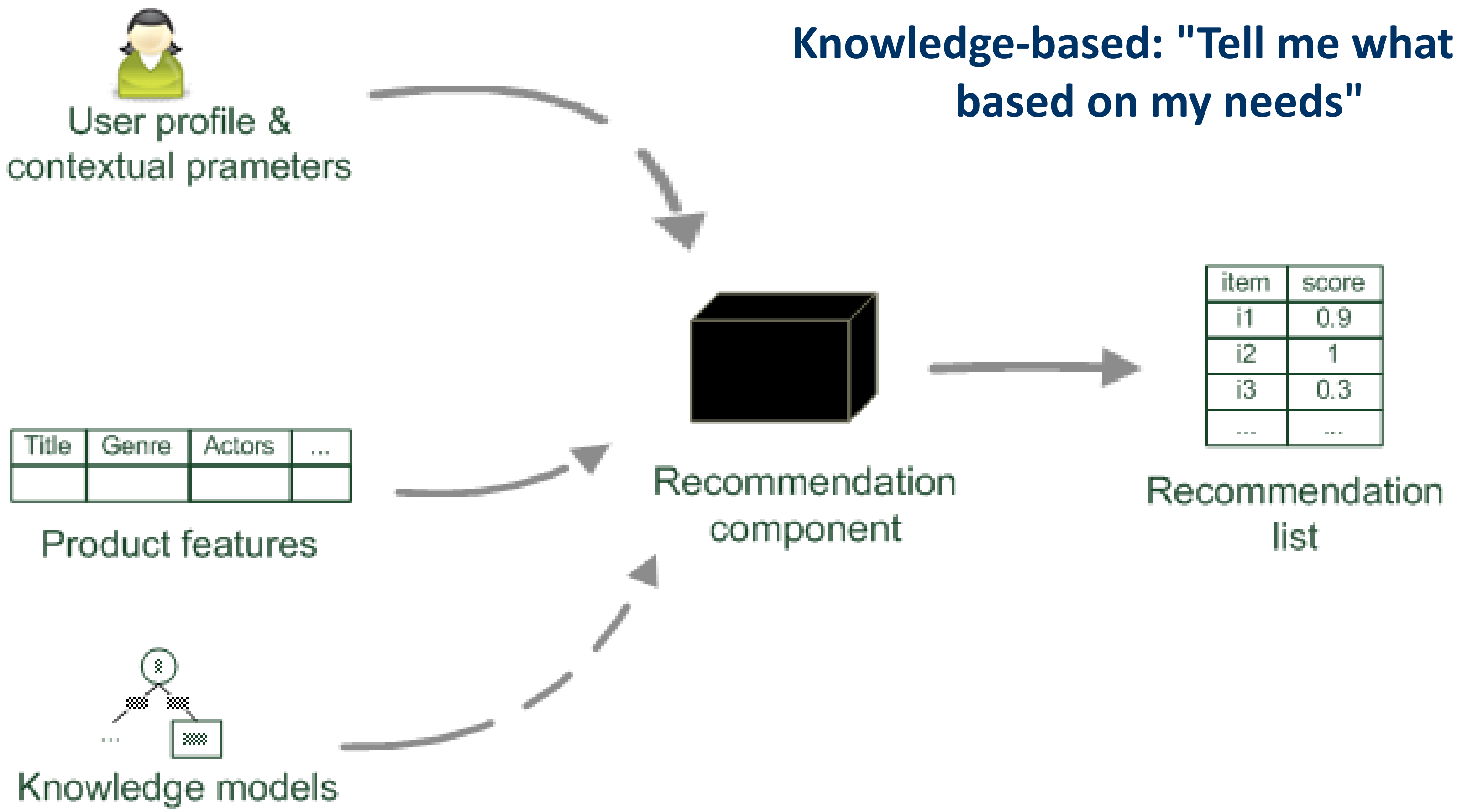
Paradigms of recommender systems

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



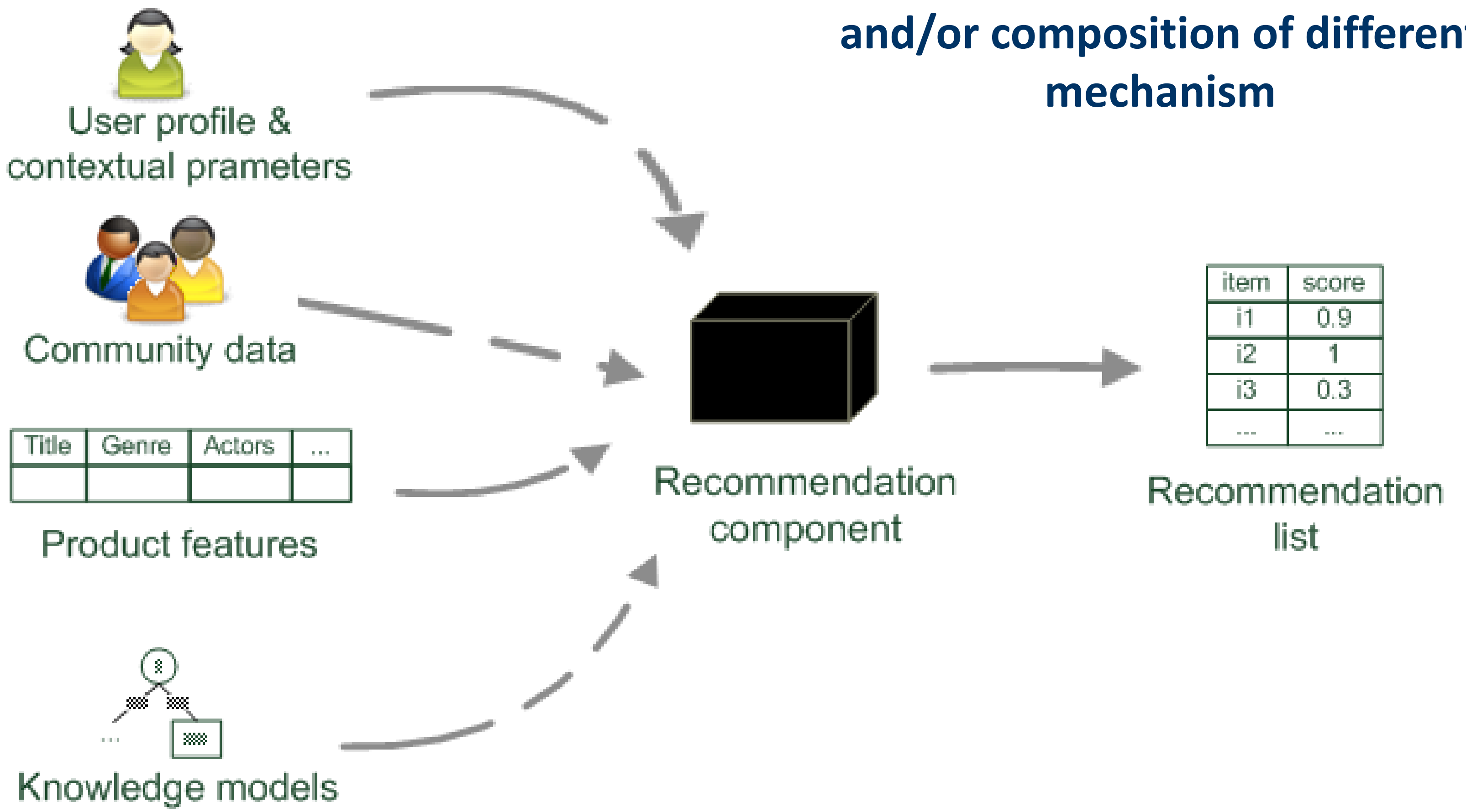
Paradigms of recommender systems

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



Paradigms comparison

	Pros	Cons
Collaborative	No knowledge-engineering 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)

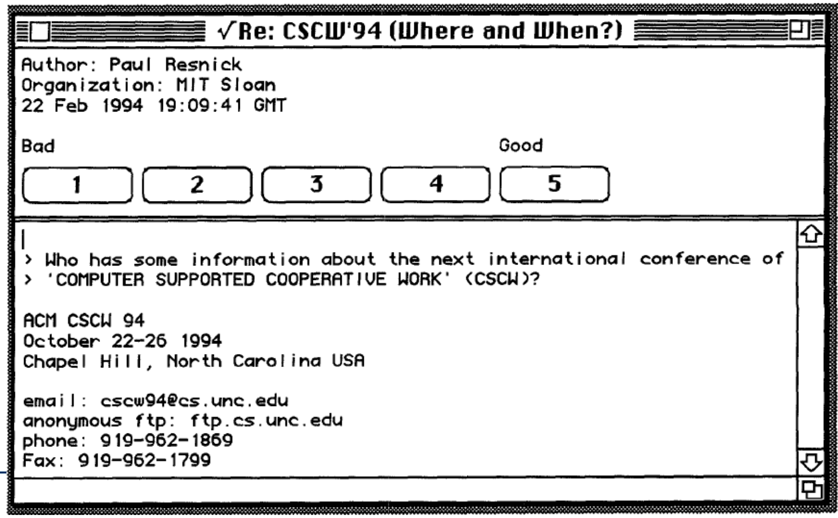
- **The most prominent approach to generate recommendations**
  - used by large, commercial e-commerce sites
  - well-understood, various algorithms and variations exist
  - applicable in many domains (book, movies, DVDs, ..)
- **Approach**
  - use the "wisdom of the crowd" （群智）to recommend items
- **Basic 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*（织锦）, 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



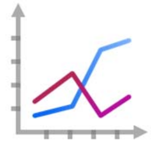
User-based nearest-neighbor collaborative filtering (1)

- **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

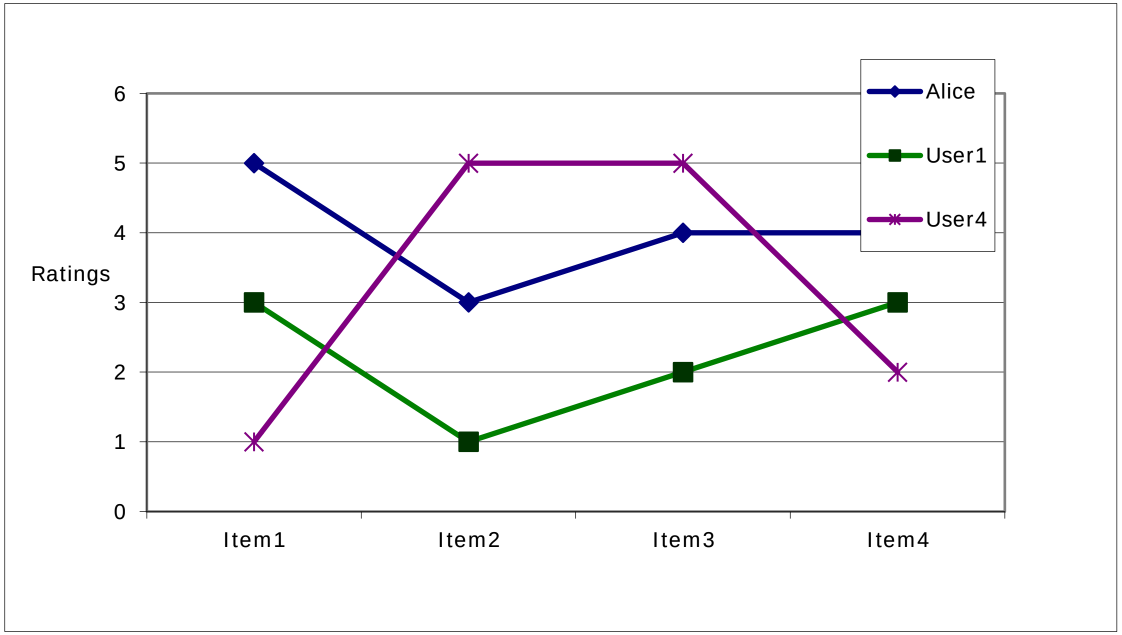
	Item1	Item2	Item3	Item4	Item5
Alice	5	3	4	4	?
User1	3	1	2	3	3
User2	4	3	4	3	5
User3	3	3	1	5	4
User4	1	5	5	2	1

sim = 0.85
sim = 0.70
sim = -0.79

© Dietmar Jannach, Markus Zanker and Gerhard Friedrich

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:基于用户相似度的加权平均

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

- 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(有争议的条目)
  - **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 (评价矩阵) 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 人口统计学 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 too 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

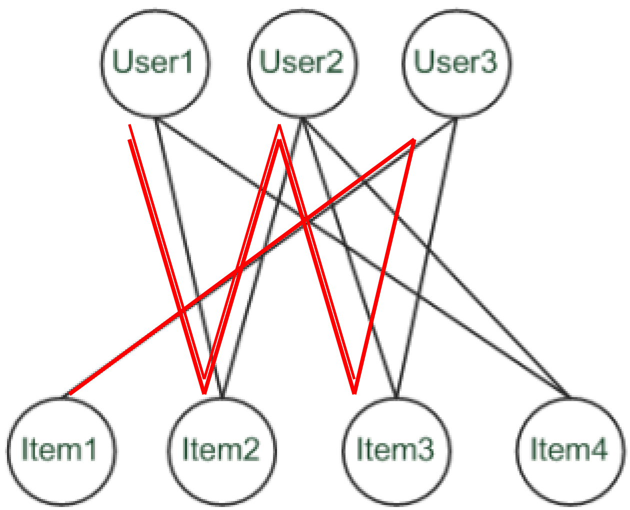
	Item1	Item2	Item3	Item4	Item5
Alice	5	3	4	4	?
User1	3	1	2	3	?
User2	4	3	4	3	5
User3	3	3	1	5	4
User4	1	5	5	2	1

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

- **Plenty 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?

Evaluations of CF systems

- **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

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?**

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(幻想小说) to people who liked fantasy novels in the past
- **What do we need:**
  - some information about the available items such as the genre(体裁、流派）("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)**

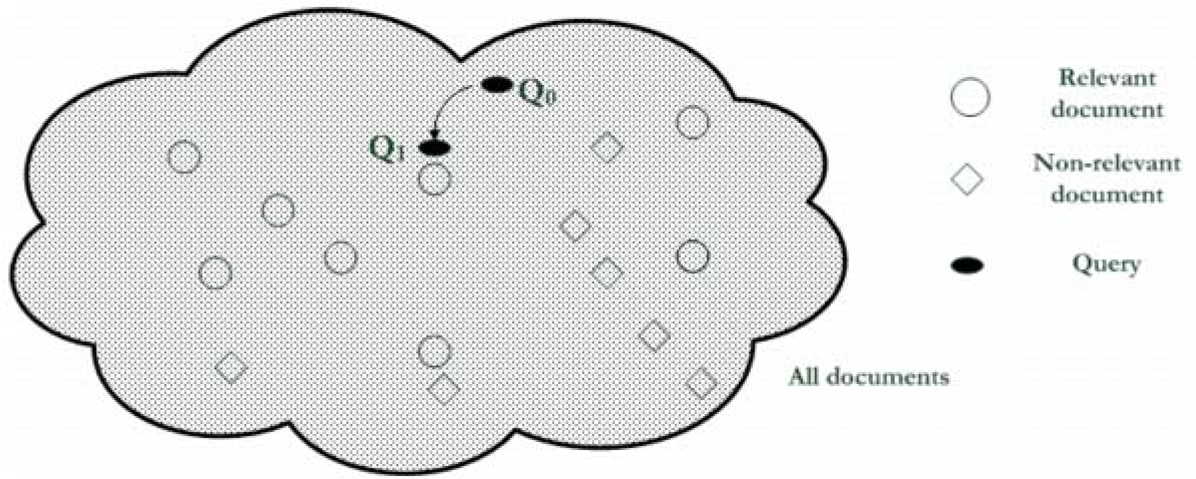
Recommending items

- **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	**?**

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

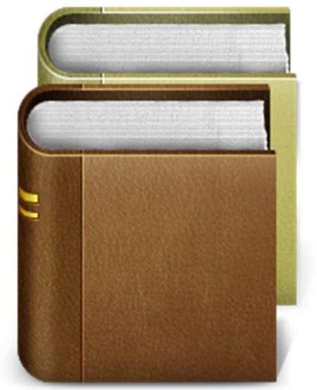
Improvements

- **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(基于效用的推荐系统)
    - 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（镜头）短焦长度, purpose（相机）用途)
  - 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)	
R1	Motives	*Landscape*
R2	Brand preference	*Canon*
R3	Max. cost	*350 EUR*

CS1

CS2

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

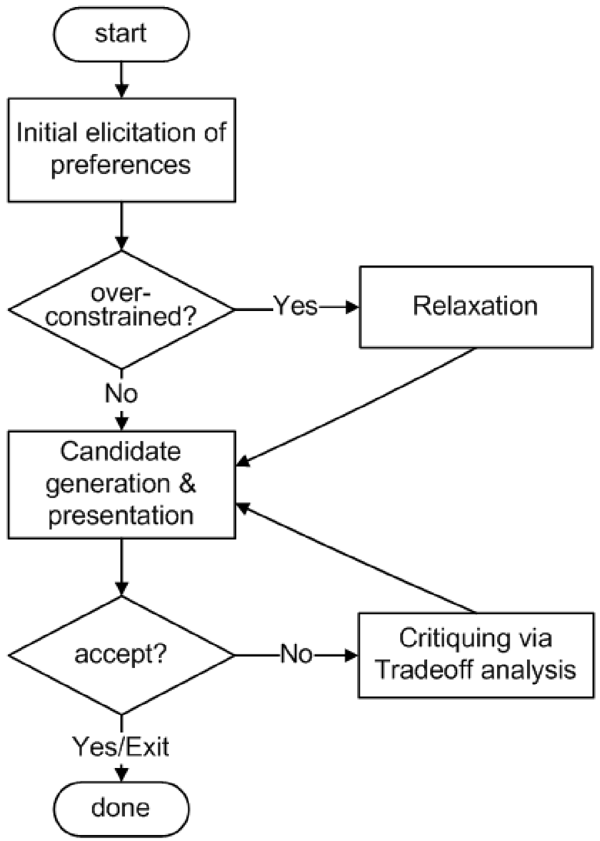
- **Be aware of possible revisions** (e.g. age, family status, …)

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， then 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(投资组合)
      - RS for different item categories, CSP(constraint satisfaction problem) 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



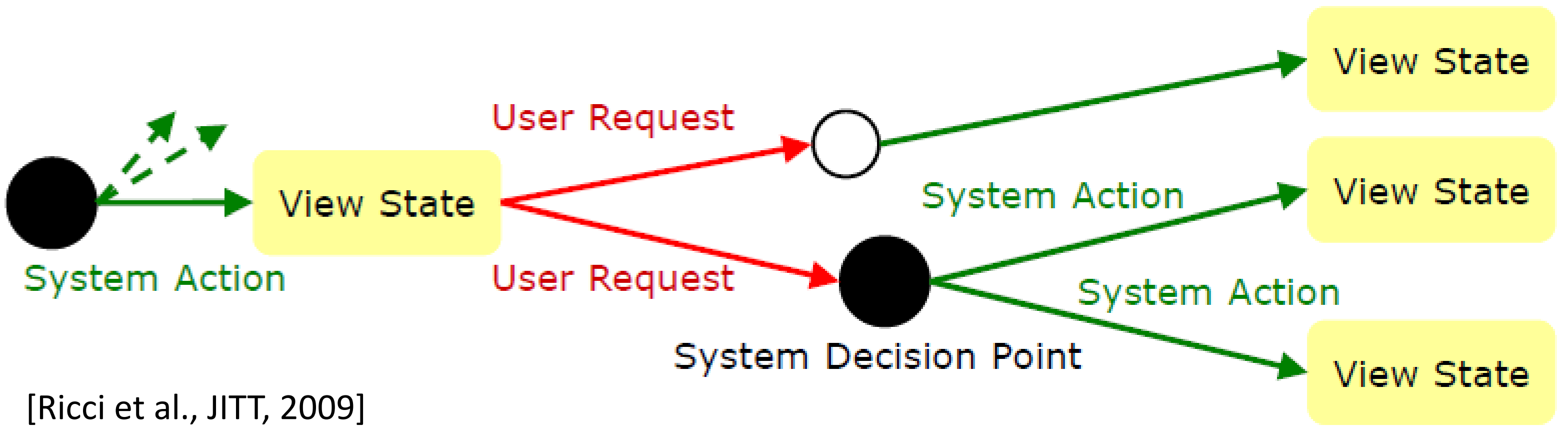
Example: critiquing(发表评论)

- **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,…



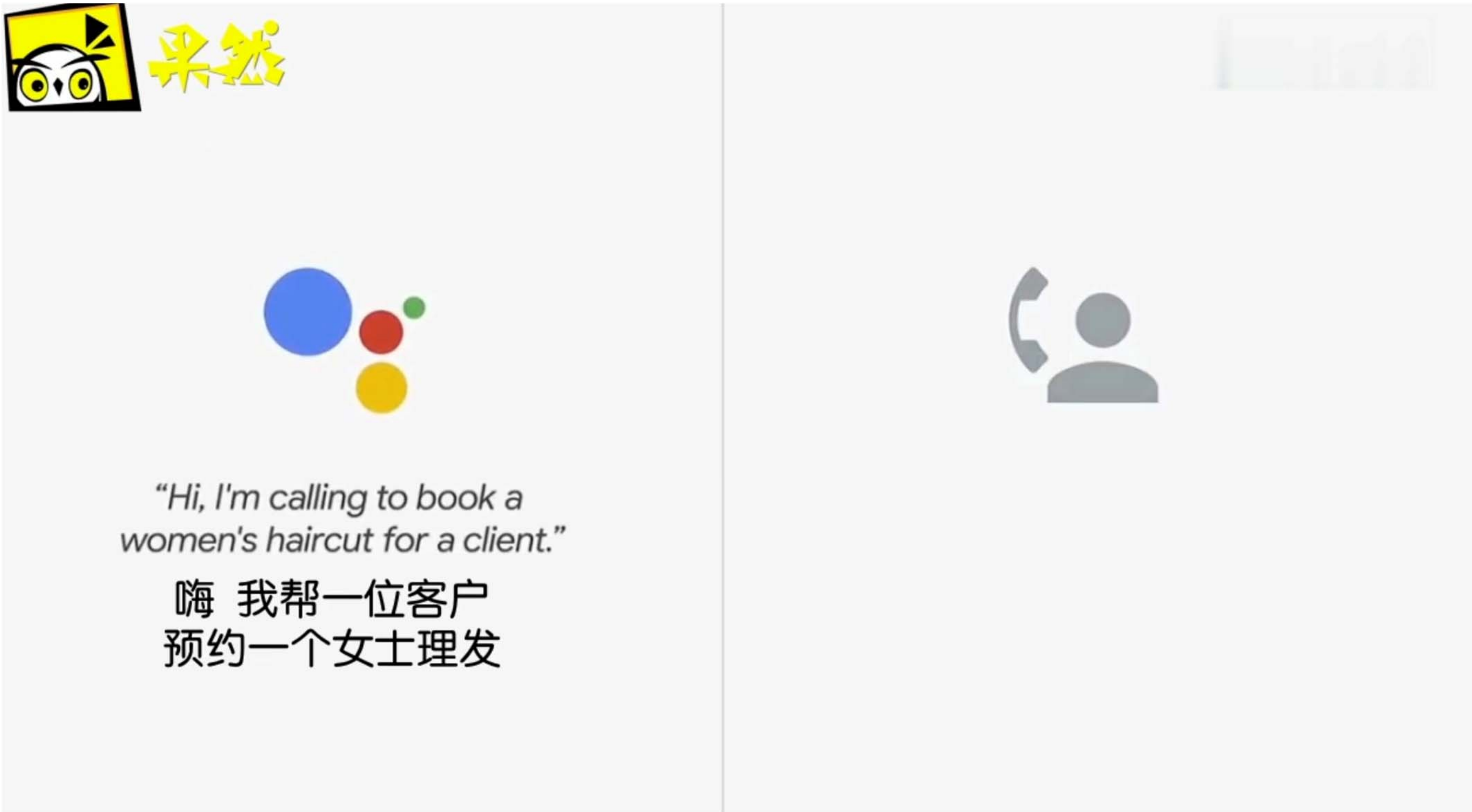
[Ricci et al., JITT, 2009]

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（诱饵）items


果然
“Hi, I'm calling to book a
women's haircut for a client.”
嗨 我帮一位客户
预约一个女士理发

Hybridization Strategies

Hybrid recommender systems

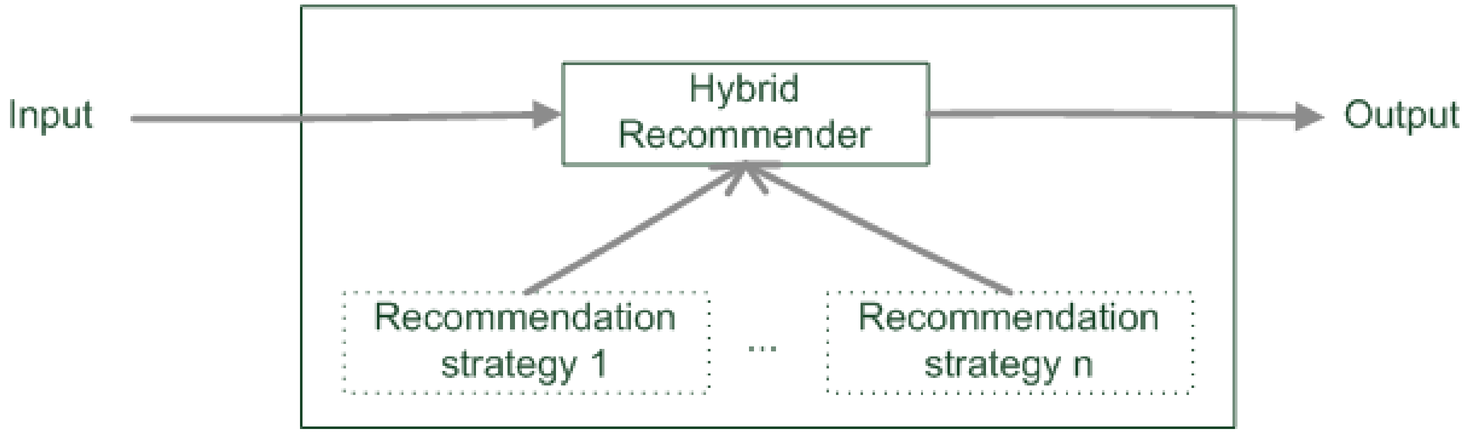
- **All three base techniques are naturally incorporated by a good sales assistance (at different stages of the sales act) but have their shortcomings**

- **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（整体） 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]

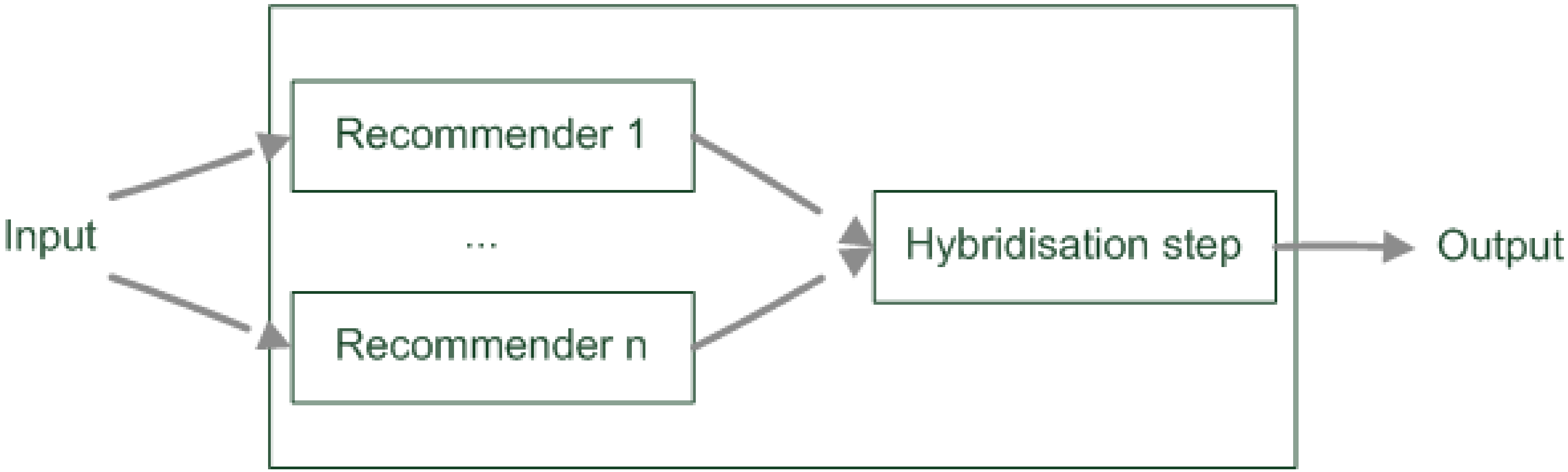
Monolithic hybridization designs: Feature augmentation

- **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

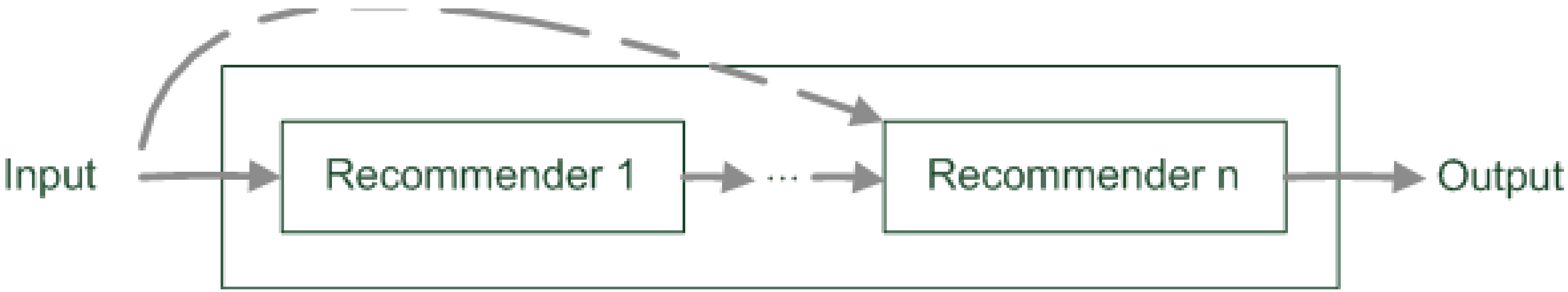


Parallelized hybridization design: Switching

- **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**

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

- **$\Delta_{rec_{n-1}}$ 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)

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.

Shuai Zhang and Lina Yao and Aixin Sun, Deep Learning based Recommender System: Survey and New Perspectives, http://arxiv.org/abs/1707.07435