Collaborative Filtering

Credits

- Dietmar Jannach
- Markus Zanker
- Alexander Felfernig
- Gerhard Friedrich
- Francesco Ricci

Agenda

- **Collaborative Filtering (CF)**
  - Pure CF approaches
  - User-based nearest-neighbor
  - The Pearson Correlation similarity measure
  - Memory-based and model-based approaches
  - Item-based nearest-neighbor
  - The cosine similarity measure
  - Data sparsity problems
  - Recent methods (SVD, Association Rule Mining, Slope One, RF-Rec, ...)
  - The Google News personalization engine
  - Discussion and summary
  - Literature

Recommender Systems

- In everyday life **we rely on recommendations** from other people either by word of mouth, recommendation letters, movie and book reviews printed in newspapers …
- In a typical recommender system **people provide recommendations as inputs, which the system then aggregates and directs to appropriate recipients**
  - Aggregation of recommendations
  - Match the recommendations with those searching for recommendations.

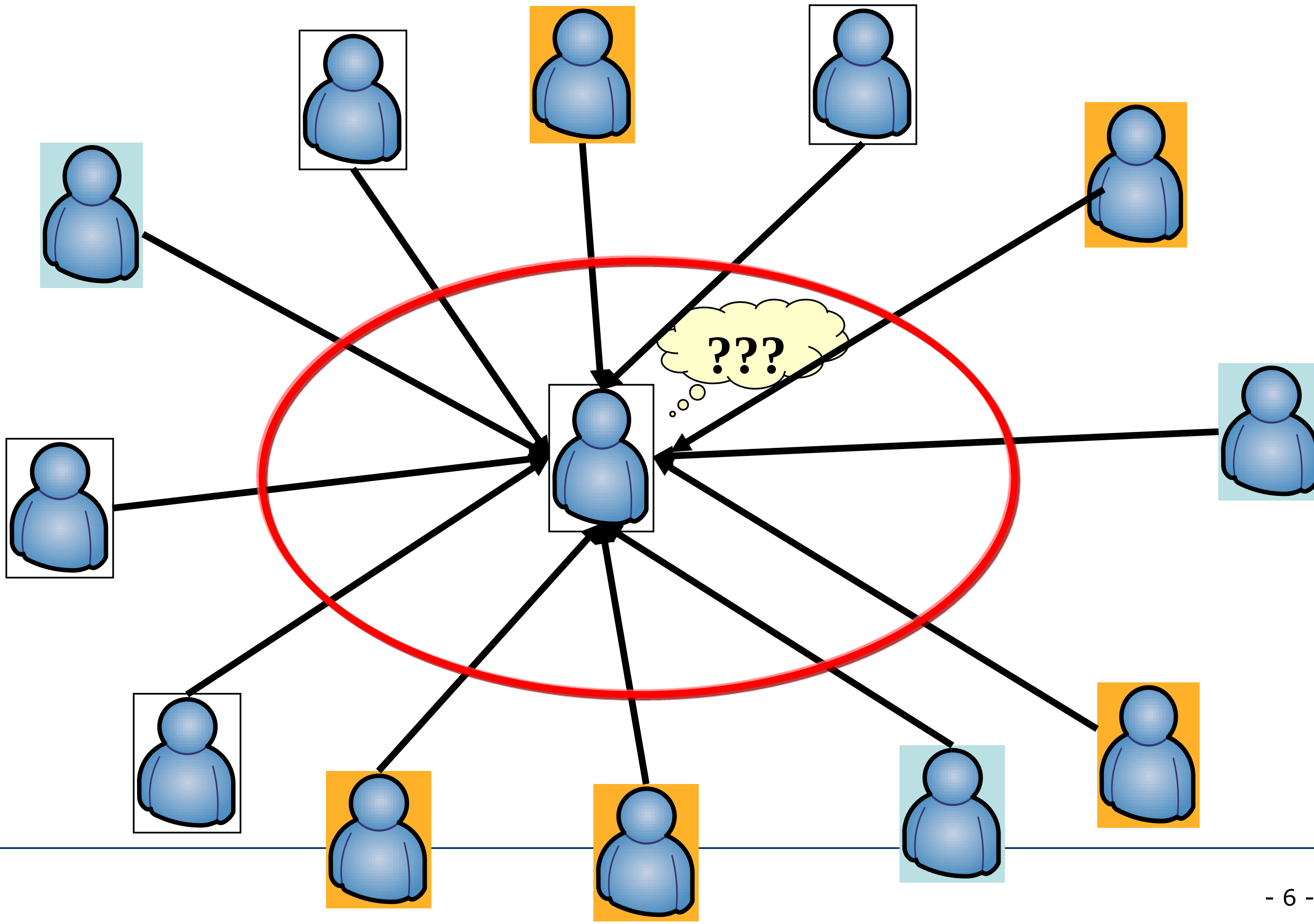
[Resnick and Varian, 1997]

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

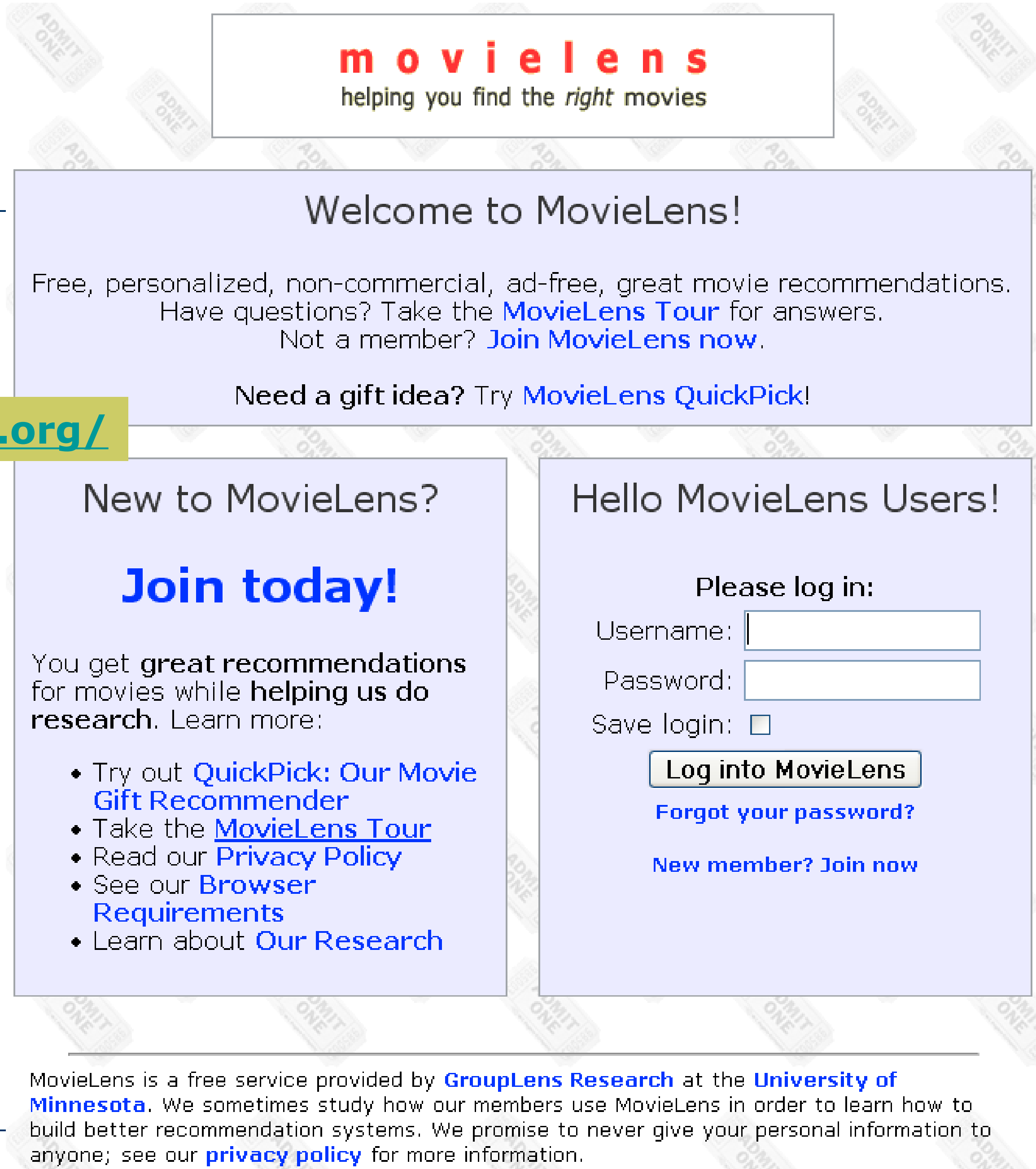
Collaborative or Social Filtering



Collaborative Filtering Idea

- Trying to **predict** the opinion the user will have on the different items and be able to recommend the “best” items to each user
- It is based on: **the user’s previous likings and the opinions of other like minded users**
- CF is a typical **Internet application** – it must be supported by a networking infrastructure
  - At least many users and one server
  - But also a distributed model with many servers
- There is no stand alone CF application.

MovieLens



http://www.movielens.org/

http://news.cnet.com/8301-17939_109-10200031-2.html

Welcome to the new MovieLens!

Existing MovieLens users: We'd like to welcome you back to MovieLens, and let you know we have a new MovieLens FAQ you might want to read. We hope you like what you will see!

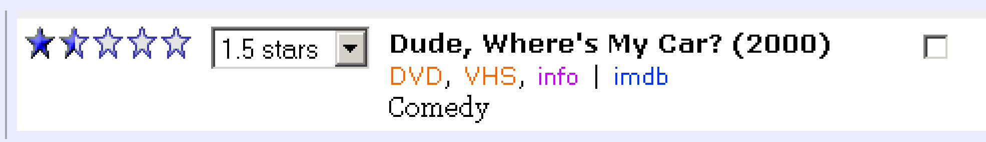
Take me to MovieLens!

New MovieLens users: Thank you for joining MovieLens! In order to generate personalized movie recommendations, we need to know a little about what movies you have already seen. MovieLens will now display several lists of movies. If you have seen any of the listed movies, please rate them using the rating scale shown below.

Ratings are on a scale of 1 to 5:

Remember: the more movies you rate, the more accurate MovieLens' predictions will be.

To rate a movie, just click on the pulldown next to the title of a movie you have seen. Blue stars will appear to indicate that your rating has been received.



This image shows that the movie 'Dude, Where's My Car?' was rated 1.5 stars.

I'm ready to start rating!

So far you have rated **0** movies.
MovieLens needs at least **15** ratings from you to generate predictions for you.
Please rate as many movies as you can from the list below.

next >

	Your Rating	Movie Information
★★★	3.0 stars	**Austin Powers: International Man of Mystery (1997)** Action, Adventure, Comedy
★★★★	4.0 stars	**Contact (1997)** Drama, Sci-Fi
???	Not seen	**Crouching Tiger, Hidden Dragon (Wu Hu Zang Long) (2000)** Action, Adventure, Drama, Fantasy, Romance
???	Not seen	**Demolition Man (1993)** Action, Comedy, Sci-Fi
???	Not seen	**Eraser (1996)** Action, Drama, Thriller
???	Not seen	**Maverick (1994)** Action, Comedy, Western
★★★★½	4.5 stars	**Philadelphia (1993)** Drama
★★★½	3.5 stars	**Piano, The (1993)** Drama, Romance
???	Not seen	**Toy Story 2 (1999)** Adventure, Animation, Children, Comedy, Fantasy
★★★½	3.5 stars	**X-Men (2000)** Action, Adventure, Sci-Fi

next >

To get a new set of movies click the **next>** link.

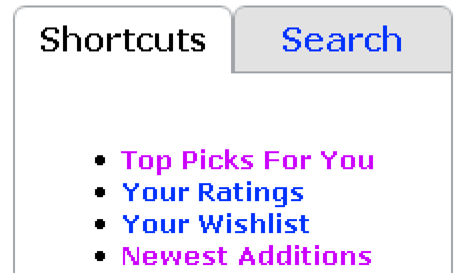
Congratulations!

MovieLens can now generate personalized movie recommendations for you.

Start Using MovieLens

Remember, you can always keep rating movies you have seen. The more movies you rate, the better your predictions will be. We'd also like to tell you about some other features of MovieLens you might be interested in:

- **Getting recommendations.** MovieLens has shortcuts like Top Picks For You that provide you with quick access to common searches. You can use the Search tab to perform more advanced searches that filter by genre, date, and more, and save your favorite searches as personal shortcuts.

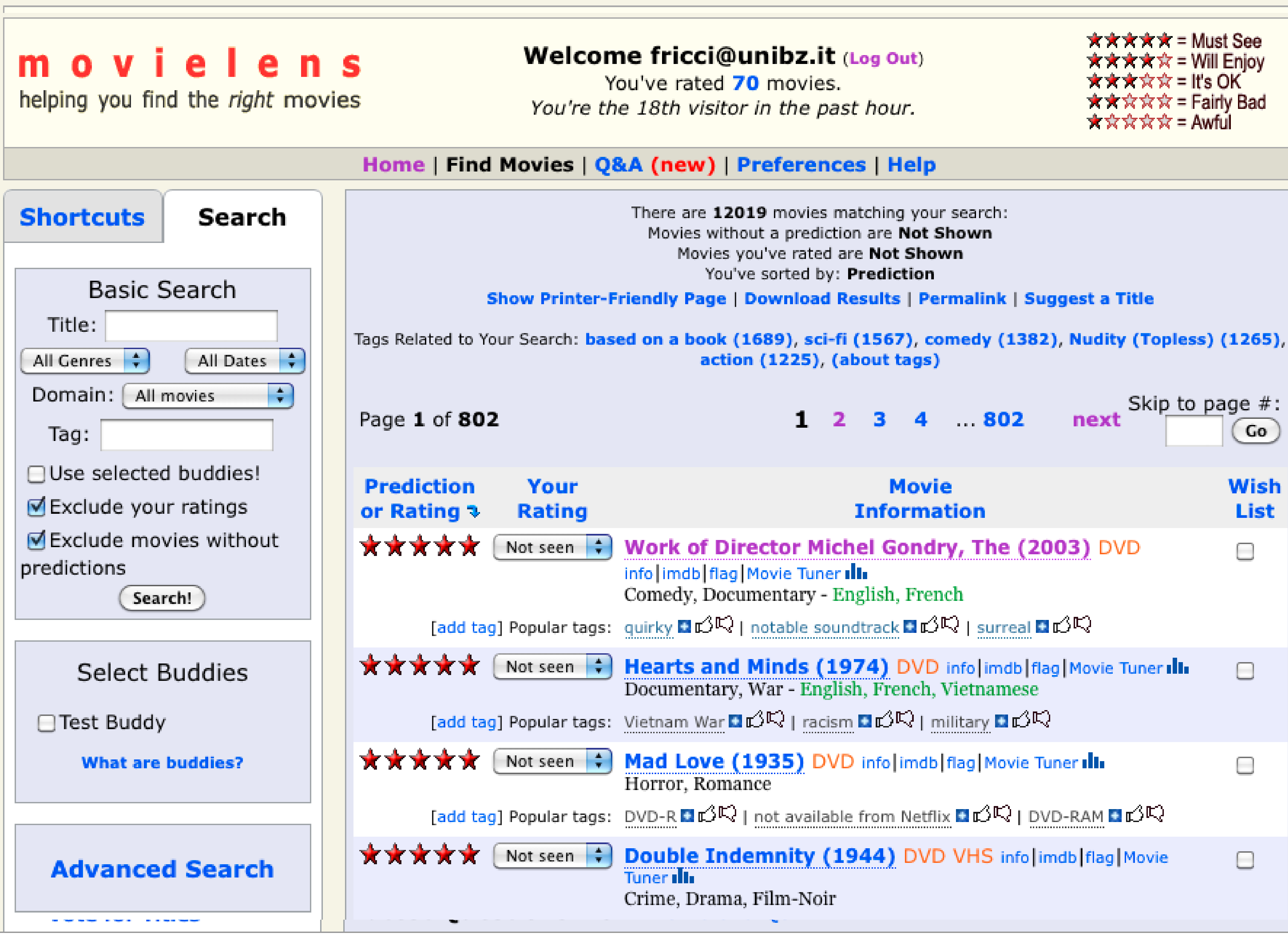


- **Your Wishlist.** Here you can keep track of movies you haven't yet seen. You can even print this list out and take it with you to your video store.

- **Movie buddies.** It can be a pain trying to decide what movie a group of people should see. Let MovieLens choose the right movie for you! You can add MovieLens users to be your buddies and be able to generate group movie recommendations

We will keep adding more great features as time goes on, so look for them!

Start Using MovieLens


m o v i e l e n s
helping you find the right movies
Welcome fricci@unibz.it (Log Out)
You've rated 70 movies.
You're the 18th visitor in the past hour.
= Must See
= Will Enjoy
= It's OK
= Fairly Bad
= Awful
Home | Find Movies | Q&A (new) | Preferences | Help
Shortcuts
Search
Basic Search
Title:
All Genres
All Dates
Domain: All movies
Tag:
Use selected buddies!
Exclude your ratings
Exclude movies without predictions
Search!
Select Buddies
Test Buddy
What are buddies?
Advanced Search
There are 12019 movies matching your search:
Movies without a prediction are Not Shown
Movies you've rated are Not Shown
You've sorted by: Prediction
Show Printer-Friendly Page | Download Results | Permalink | Suggest a Title
Tags Related to Your Search: based on a book (1689), sci-fi (1567), comedy (1382), Nudity (Topless) (1265), action (1225), (about tags)
Page 1 of 802
1 2 3 4 ... 802 next
Skip to page #:
Go
Prediction or Rating
Your Rating
Movie Information
Wish List
Not seen
Work of Director Michel Gondry, The (2003) DVD
info|imdb|flag|Movie Tuner
Comedy, Documentary - English, French
[add tag] Popular tags: quirky | notable soundtrack | surreal
Not seen
Hearts and Minds (1974) DVD info|imdb|flag|Movie Tuner
Documentary, War - English, French, Vietnamese
[add tag] Popular tags: Vietnam War | racism | military
Not seen
Mad Love (1935) DVD info|imdb|flag|Movie Tuner
Horror, Romance
[add tag] Popular tags: DVD-R | not available from Netflix | DVD-RAM
Not seen
Double Indemnity (1944) DVD VHS info|imdb|flag|Movie Tuner
Crime, Drama, Film-Noir

m o v i e l e n s
helping you find the *right* movies

Welcome fricci@unibz.it (Log Out)
You've rated **70** movies.
You're the 18th visitor in the past hour.

★★★★★ = Must See
★★★★☆ = Will Enjoy
★★★☆☆ = It's OK
★★☆☆☆ = Fairly Bad
★☆☆☆☆ = Awful

Home | **Find Movies** | **Q&A (new)** | **Preferences** | **Help**

Shortcuts | **Search**

Rate and Find Movies

- **Top Picks For You**
- **Newest Additions**
- **Most Often Rated**
- **Rate Random Movies**
- **Browse Movies by Tags**

Your Movies

- **Your Ratings**
- **About Your Ratings**
- **Your Wishlist**
- **Your Tags**

Your Account

- **Your Profile** (**edit**)
- **Preferences**
- **Manage Buddies**
- **Manage RSS Feeds**

Help MovieLens

- **Volunteer Center**
- **Vote for Titles**

Work of Director Michel Gondry, The (2003)

Your Prediction: ★★★★★
Rate This Movie: Not seen   Wish List: ☐

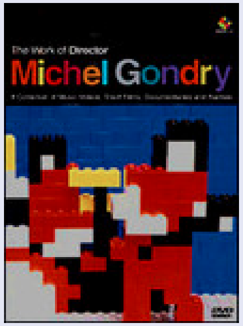
Movie Information **(edit info)** **(flag)**

Starring: **Michel Gondry**, **Björk**, **Beck**, **David Grohl**, **David Cross**, **Jack White**, **Meg White**, **Cibo Matto**
Directed By: **Michel Gondry**, **Lance Bangs**, **Olivier Gondry**
Genres: **Comedy**, **Documentary**
Languages: **English French**
Average rating: ★★★★ (4.11 stars)
Your Prediction: ★★★★★ (5.0 stars)
Rated by: 71 users
Links: **IMDb**, **Rotten Tomatoes**

Movie Tags **(more about tags)**
Add and edit tags here or **update all of your tags**

Community Tags (?)
Tags represent how MovieLens users feel about this movie

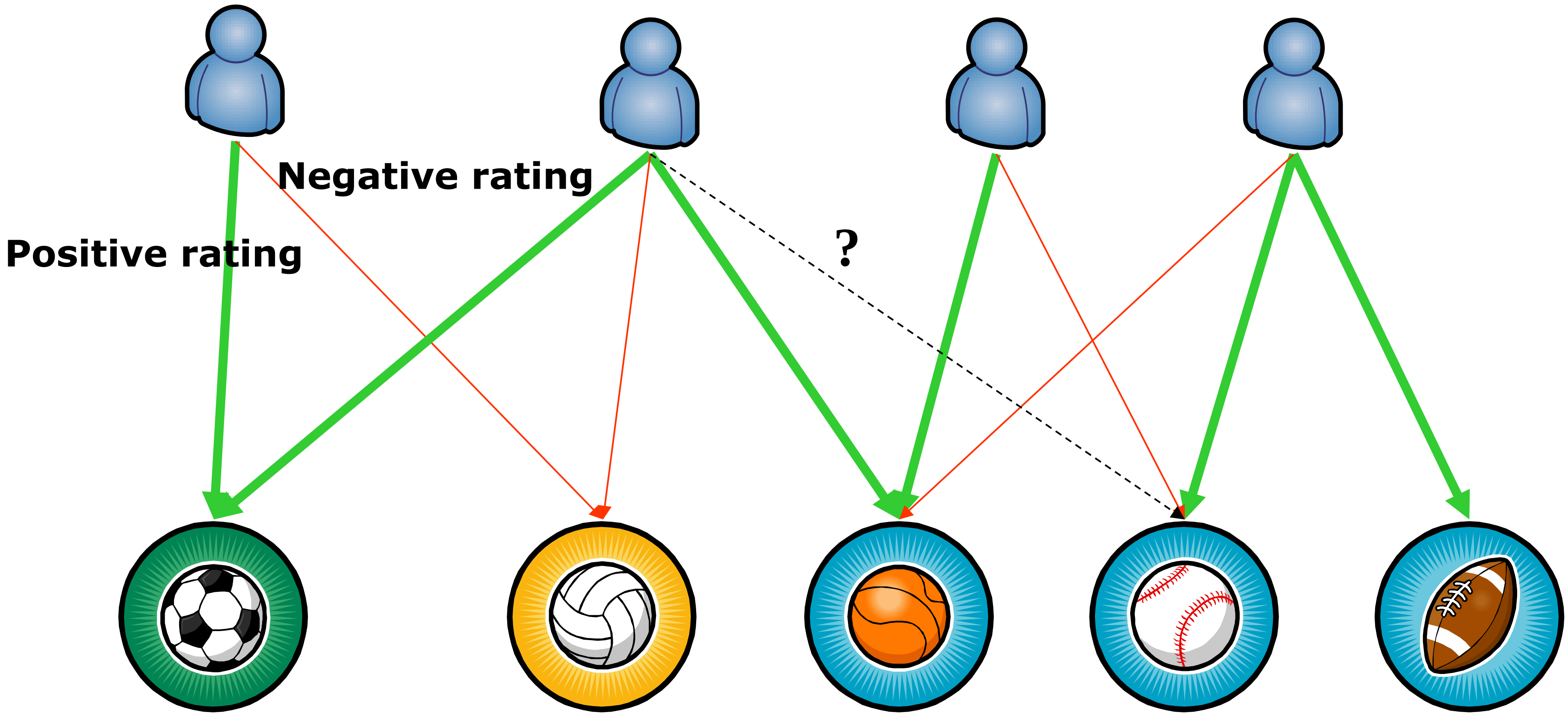
01/11 02/11 03/11 bjork view
creative David Cross



The Work of Director Michel Gondry
The tireless creativity of director Michel Gondry is on vivid display in this collection of 27 music videos and other whimsical oddities. Released the year before Gondry's feature breakthrough Eternal Sunshine of the Spotless Mind, the compilation includes Kylie Minogue's "Come into My World," Bjork's "Human Behavior," Massive Attack's "Protection" and the White Stripes' Lego-centric stunner "Fell in Love with a Girl."
Report Wrong MoviDelivered by Netflix (add to queue)

Movie Tuner (?) **New!**
Find similar movies with less or more of particular qualities. The movie list below will update as you indicate your preferences.

Collaborative Filtering



Collaborative Filtering Ingredients

- List of ***m*** **Users** and a list of ***n*** **Items**
- Each user has a **list of items** he/she expressed their **opinion** about (can be a null set)
- **Explicit opinion** - a rating score (numerical scale)
  - Sometime the rating is **implicitly** – purchase records
- **Active user** for whom the CF prediction task is performed
- A **metric** for measuring **similarity between users**
- A method for selecting a **subset of neighbors** for prediction
- A method for **predicting a rating** for items not currently rated by the active user.

Pure CF Approaches

- **Input**
  – Only a matrix of given user–item ratings
- **Output types**
  – A (numerical) prediction indicating to what degree the current user will like or dislike a certain item
  – A top-N list of recommended items

User-based nearest-neighbor collaborative filtering (1)

- **The basic technique**
  - Given an "active user" (Alice) and an item i not yet seen by Alice
    - find a set of users (peers/nearest neighbors) 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
- **Basic assumption and idea**
  - If users had similar tastes in the past they will have similar tastes in the future
  - User preferences remain stable and consistent over time

User-based nearest-neighbor collaborative filtering (2)

- **Example**
  - A database of ratings of the current user, Alice, and some other users is given:

	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

  - Determine whether Alice will like or dislike *Item5*, which Alice has not yet rated or seen

User-based nearest-neighbor collaborative filtering (3)

- **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 (1)

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

  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

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

Measuring user similarity (2)

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

  a, b : users

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

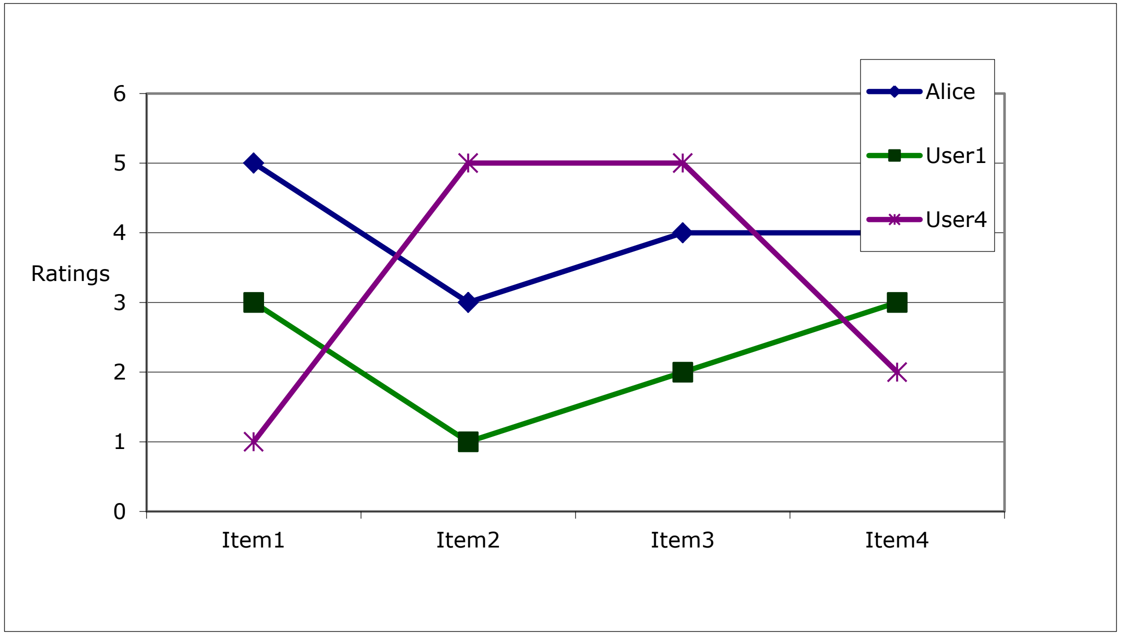
  P : set of items, rated both by a and b

  – Possible similarity values between -1 and 1

	Item1	Item2	Item3	Item4	Item5	
Alice	5	3	4	4	?	
User1	3	1	2	3	3	sim = 0.85
User2	4	3	4	3	5	sim = 0.70
User3	3	3	1	5	4	sim = 0.00
User4	1	5	5	2	1	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

Proximity Measure: Cosine

- Correlation can be replaced with a typical Information Retrieval (IR) similarity measure: **cosine**
- This has been shown to provide worse results by someone [Breese et al., 1998]
- But many uses cosine [Sarwar et al., 2000] and somebody reports that it performs better [Anand and Mobasher, 2005]

Comparison: Pearson vs. Cosine

	user 1	user 2	user 3
p1	1	2	5
p2	3	4	3
p3	4	5	2
p4	2	3	4
p5	1	2	5
p6	2	3	4
p7	2	3	4
p8	1	2	5

Pearson

	user 1	user 2	user 3
user 1	1	1	-1
user 2	1	1	-1
user 3	-1	-1	1

Cosine

	user 1	user 2	user 3
user 1	1,00	0,99	0,76
user 2	0,99	1,00	0,84
user 3	0,76	0,84	1,00

- User 2 ratings are those of user 1 incremented by 1
- User 3 has "opposite" preferences of user 1

Making predictions

- A common prediction function:

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

- 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 a 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

- **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
- **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
- **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

Improving the metrics / prediction function

- **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
  - Reduce the magnitude of a similarity weight when this weight is computed using only a few ratings
  - For example user similarity weight penalized by a factor proportional to the number of commonly rated items, if this number is less than a given parameter γ > 0

$$w'_{uv} = \frac{\min\{|\mathcal{I}_{uv}|, \gamma\}}{\gamma} \times w_{uv}$$

  - Likewise, an item similarity *wi j*, obtained from a few ratings, can be adjusted as

$$w'_{ij} = \frac{\min\{|\mathcal{U}_{ij}|, \gamma\}}{\gamma} \times w_{ij}$$

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
  - A recommendation approach that addresses this problem is the *Inverse User Frequency*. Based on the information retrieval notion of IDF, a weight λ_i is given to each item i, in proportion to the log-ratio of users that have rated i:

$$\lambda_i = \log \frac{|\mathcal{U}|}{|\mathcal{U}_i|}$$

  - While computing the Frequency-Weighted Pearson Correlation (FWPC) between users u and v, the correlation between the ratings given to an item i is weighted by λ_i:

$$\mathrm{FWPC}(u,v) = \frac{\sum\limits_{i \in \mathcal{I}_{uv}} \lambda_i (r_{ui} - \overline{r}_u)(r_{vi} - \overline{r}_v)}{\sqrt{\sum\limits_{i \in \mathcal{I}_{uv}} \lambda_i (r_{ui} - \overline{r}_u)^2 \sum\limits_{i \in \mathcal{I}_{uv}} \lambda_i (r_{vi} - \overline{r}_v)^2}}$$

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
  - *item*-based CF is an example for model-based approaches

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**
- **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
  - U: set of users who have rated both items a and b

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

Making predictions

- A common prediction function:

$$pred(u,p) = \frac{\sum_{i \in ratedItem(u)} sim(i,p) * r_{u,i}}{\sum_{i \in ratedItem(u)} sim(i,p)}$$

- Neighborhood size is typically also limited to a specific size
- Not all neighbors are taken into account for the prediction
- An analysis of the MovieLens dataset indicates that "in most real-world situations, a neighborhood of 20 to 50 neighbors seems reasonable" (Herlocker et al. 2002)

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
    - Limit the neighborhood size (might affect recommendation accuracy)

More on ratings – Explicit ratings

- Probably the most precise 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 dom.
  - An even more fine-grained scale was chosen in the joke recommender discussed by Goldberg et al. (2001), where a continuous scale (from −10 to +10) and a graphical input bar were used
    - No precision loss from the discretization
    - User preferences can be captured at a finer granularity
    - Users actually "like" the graphical interaction method
  - Multidimensional ratings (multiple ratings per movie such as ratings for actors and sound)
- Main problems
  - Users not always willing to rate many items
    - number of available ratings could be too small → sparse rating matrices → poor recommendation quality
  - How to stimulate users to rate more items?

More on ratings – Implicit ratings

- Typically collected by the web shop or application in which the recommender system is embedded
- When a customer buys an item, for instance, many recommender systems interpret this behavior as a positive rating
- Clicks, page views, time spent on some page, demo downloads …
- Implicit ratings can be collected constantly and do not require additional efforts from the side of the user
- Main problem
  - One cannot be sure whether the user behavior is correctly interpreted
  - For example, a user might not like all the books he or she has bought; the user also might have bought a book for someone else
- Implicit ratings 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
  - Default voting: assign default values to items that only one of the two users to be compared has rated (Breese et al. 1998)
- **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

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,...**

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**
- **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

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

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 (novelty and surprising effect of recommendations)
    - Not yet fully understood
- **What about multi-dimensional ratings?**

Literature (1)

- **[Adomavicius and Tuzhilin 2005]** Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions, IEEE Transactions on Knowledge and Data Engineering 17 (2005), no. 6, 734–749
- **[Breese et al. 1998]** Empirical analysis of predictive algorithms for collaborative filtering, Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence (Madison, WI) (Gregory F. Cooper and Seraf´in Moral, eds.), Morgan Kaufmann, 1998, pp. 43–52
- **[Gedikli et al. 2011]** RF-Rec: Fast and accurate computation of recommendations based on rating frequencies, Proceedings of the 13th IEEE Conference on Commerce and Enterprise Computing - CEC 2011, Luxembourg, 2011, forthcoming
- **[Goldberg et al. 2001]** Eigentaste: A constant time collaborative filtering algorithm, Information Retrieval **4** (2001), no. 2, 133–151
- **[Golub and Kahan 1965]** Calculating the singular values and pseudo-inverse of a matrix, Journal of the Society for Industrial and Applied Mathematics, Series B: Numerical Analysis **2** (1965), no. 2, 205–224
- **[Herlocker et al. 2002]** An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms, Information Retrieval 5 (2002), no. 4, 287–310
- **[Herlocker et al. 2004]** Evaluating collaborative filtering recommender systems, ACM Transactions on Information Systems (TOIS) **22** (2004), no. 1, 5–53

Literature (2)

- **[Hofmann 2004]** Latent semantic models for collaborative filtering, ACM Transactions on Information Systems 22 (2004), no. 1, 89–115
- **[Huang et al. 2004]** Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering, ACM Transactions on Information Systems 22 (2004), no. 1, 116–142
- **[Koren et al. 2009]** *Matrix factorization techniques for recommender systems*, Computer **42** (2009), no. 8, 30–37
- **[Lemire and Maclachlan 2005]** Slope one predictors for online rating-based collaborative filtering, Proceedings of the 5th SIAM International Conference on Data Mining (SDM ’05) (Newport Beach, CA), 2005, pp. 471–480
- **[Sarwar et al. 2000a]** Application of dimensionality reduction in recommender systems – a case study, Proceedings of the ACM WebKDD Workshop (Boston), 2000
- **[Zhang and Pu 2007]** A recursive prediction algorithm for collaborative filtering recommender systems, Proceedings of the 2007 ACM Conference on Recommender Systems (RecSys ’07) (Minneapolis, MN), ACM, 2007, pp. 57–64