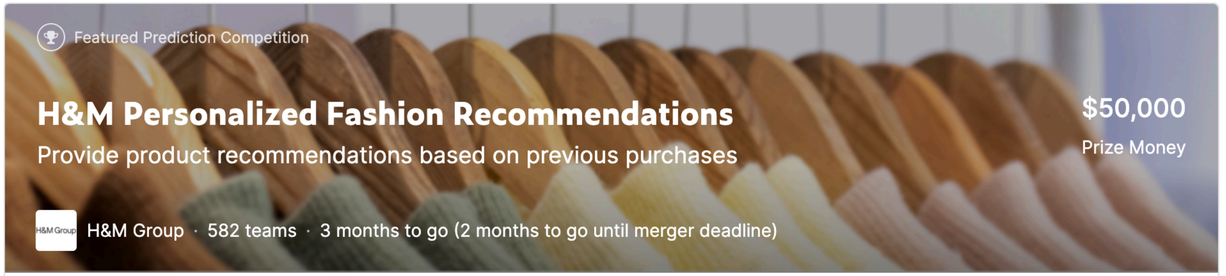


Overview Data Code Discussion Leaderboard Rules Team My Submissions **Submit Predictions** ...

Data Description

For this challenge you are given the purchase history of customers across time, along with supporting metadata. Your challenge is to predict what articles each customer will purchase in the 7-day period immediately after the training data ends. Customer who did not make any purchase during that time are excluded from the scoring.

Files

- **images/** - a folder of images corresponding to each `article_id`; images are placed in subfolders starting with the first three digits of the `article_id`; note, not all `article_id` values have a corresponding image.
- **articles.csv** - detailed metadata for each `article_id` available for purchase
- **customers.csv** - metadata for each `customer_id` in dataset
- **sample_submission.csv** - a sample submission file in the correct format
- **transactions_train.csv** - the training data, consisting of the purchases each customer for each date, as well as additional information. Duplicate rows correspond to multiple purchases of the same item. Your task is to predict the `article_id`s each customer will purchase during the 7-day period immediately after the training data period.

NOTE: You must make predictions for all `customer_id` values found in the sample submission. All customers who made purchases during the test period are scored, regardless of whether they had purchase history in the training data.

```
kaggle competitions download -c h-and-m-personalized-fashion-recommendations
```

Data Explorer

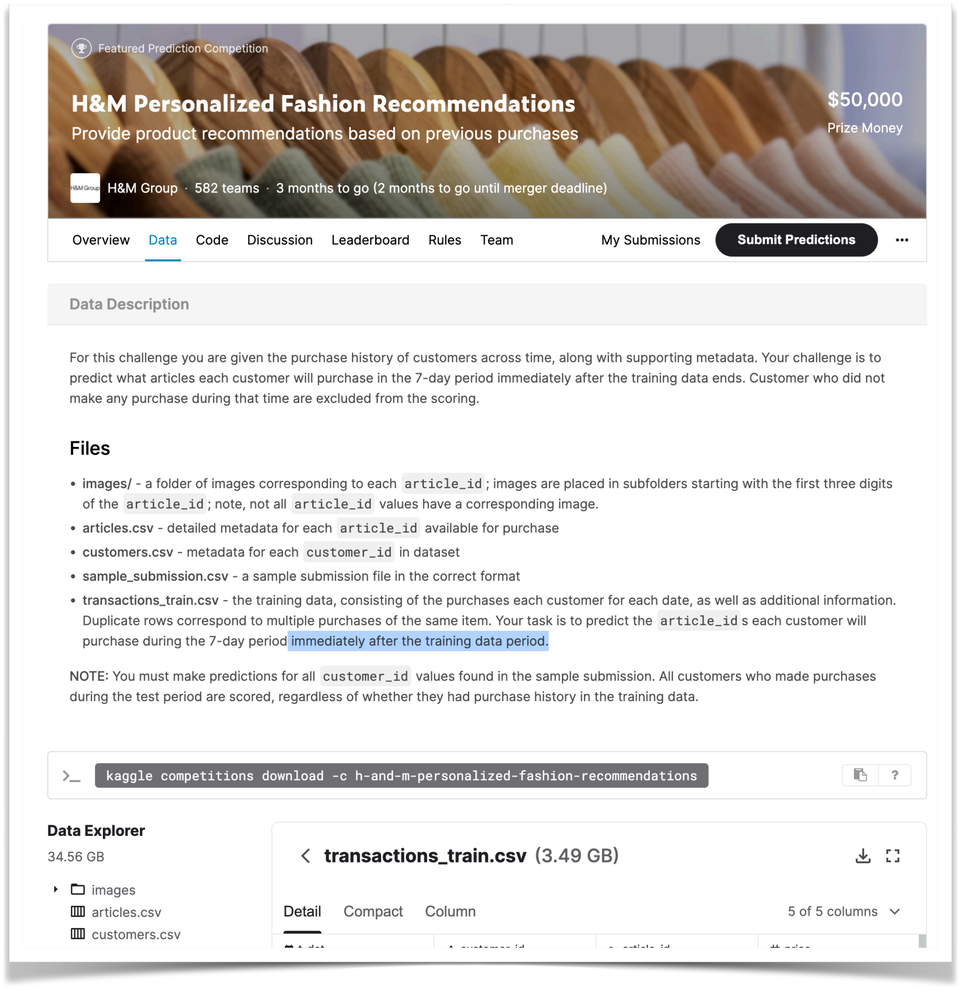
34.56 GB

- images
- articles.csv
- customers.csv

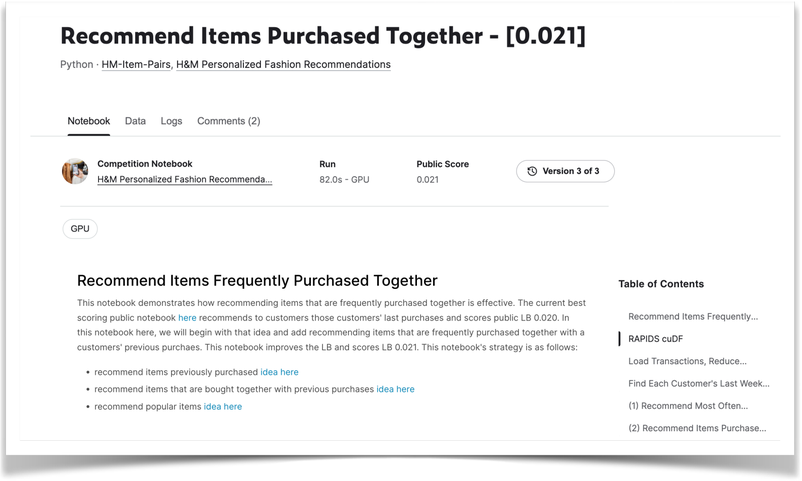
transactions_train.csv (3.49 GB)

Detail Compact Column 5 of 5 columns

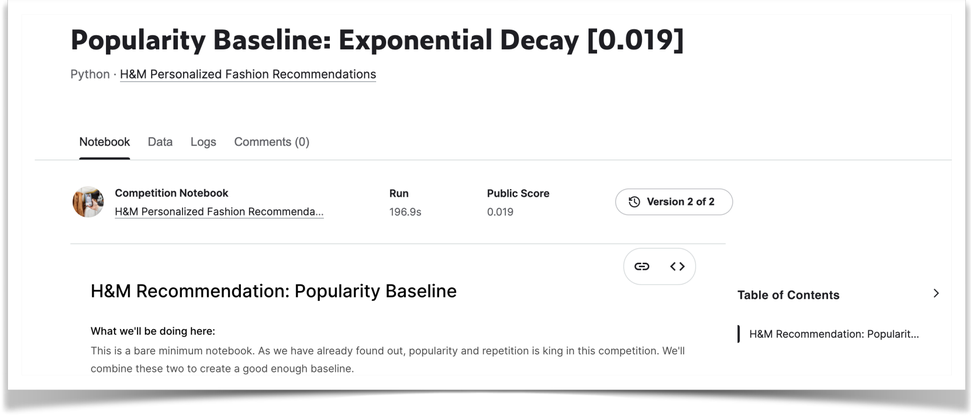
- Check these two baselines:



- https://www.kaggle.com/cdeotte/recommend-items-purchased-together-0-021



- https://www.kaggle.com/mayukh18/popularity-baseline-exponential-decay-0-019



Master on Foundations of Data Science

Recommender Systems

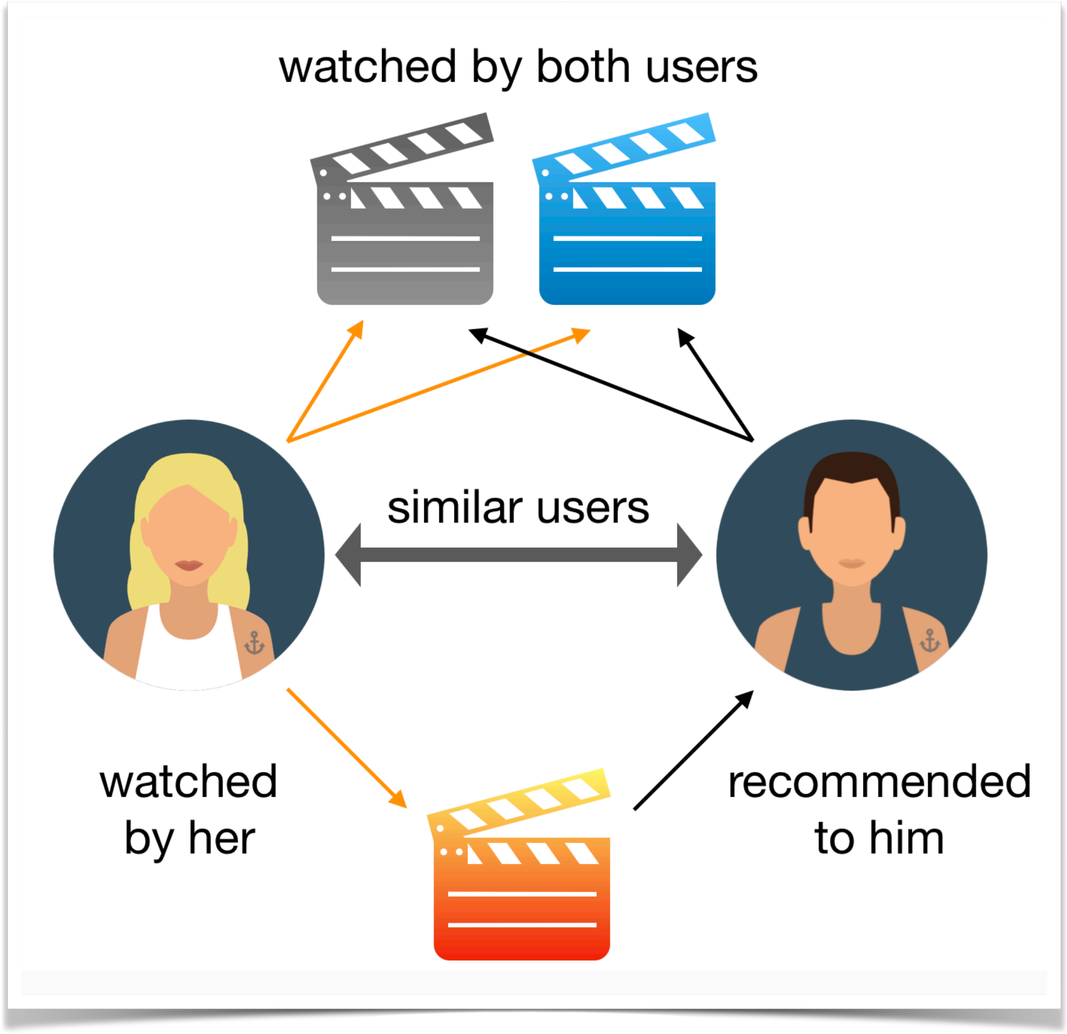
Collaborative Filtering

Santi Seguí | **2021-2022**

Collaborative-based methods

Conceptual goal:
Give me recommendation based on an approach that leverages the ratings and actions of my peers and myself

Input:
User ratings + community ratings


watched by both users
similar users
watched
by her
recommended
to him

Generalization of Supervised Classification

Features

	x1	x2	x3	x4	x5	x6	x7
U1	13,1	4	2,34	1	5,3	0,32	?
U2	1,1	3	2	4,5	4,5	9,9	?
U3	4	4,4	4,5	0,3	7,4	2,3	?
U4	9,3	32	3	5	3,2	7.54	?
U5	-2	3	5.3	5,3	3,5	9,9	?
U6	-6,3	46,3	6,2	5	8,3	4,5	?
U7	3,5	5	3,2	5,3	6,2	7,8	?

Items

	I1	I2	I3	I4	I5	I6	I7
U1	1	?	?	?	?	?	3
U2	?	3	?	4,5	4,5	?	?
U3	4	?	4,5	?	?	?	4
U4	?	?	3	5	?	?	?
U5	?	3	?	?	3,5	?	?
U6	?	?	?	5	?	4,5	3
U7	3,5	5	?	5	?	?	3

background

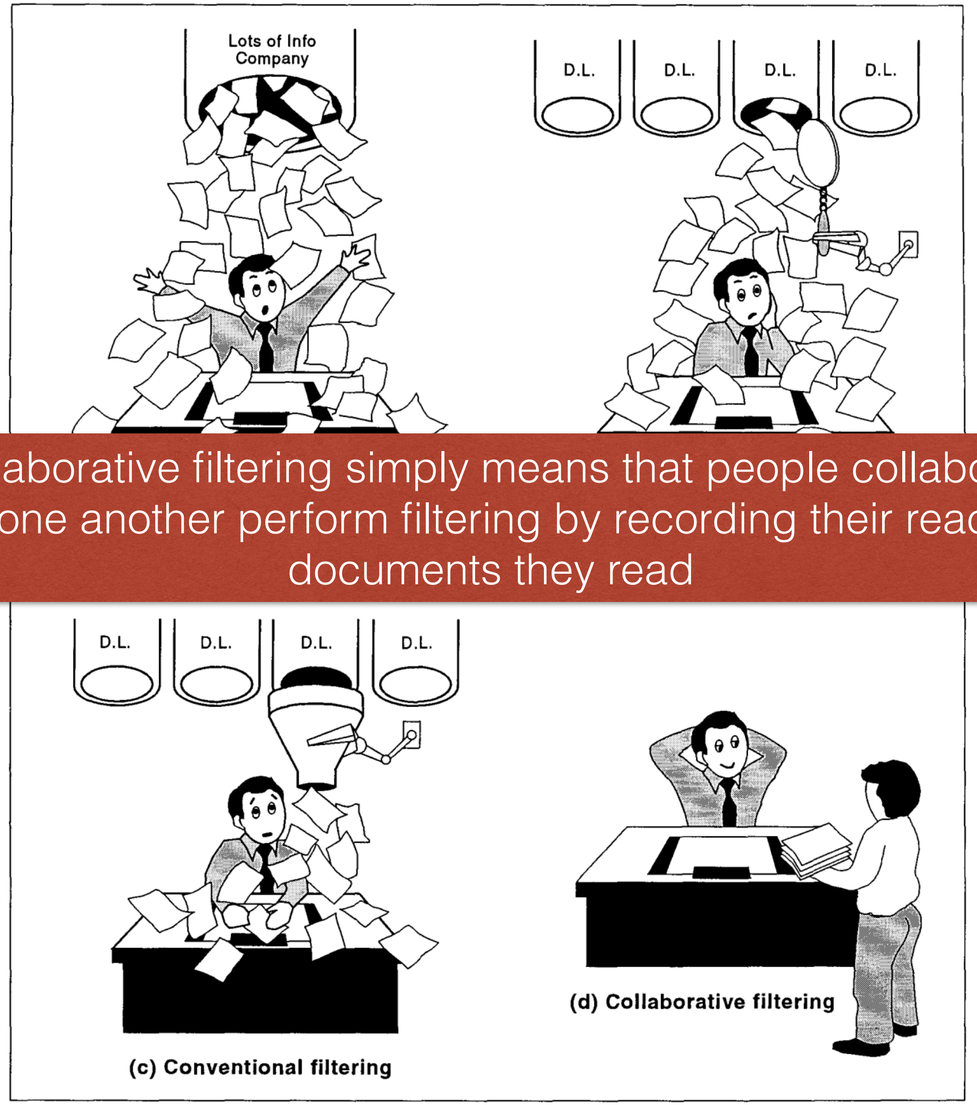
Using COLLABORATIVE FILTERING to Weave an Information TAPESTRY

David Goldberg, David Nichols, Brian M. Oki, and Douglas Terry

Tapestry is an experimental mail system developed at the Xerox Palo Alto Research Center. The motivation for Tapestry comes from the increasing use of electronic mail, which is resulting in users being inundated by a huge stream of incoming documents [2, 7, 12]. One way to handle large volumes of mail is to provide mailing lists, enabling users to subscribe only to those lists of interest to them. However, as illustrated in Figure 1, the set of documents of interest to a particular user rarely map neatly to existing lists. A better solution is for a user to specify a *filter* that scans all lists, selecting interesting documents no matter what list they are in. Several mail systems support filtering based on a document's contents [3, 5, 6, 8]. A basic tenet of the Tapestry work is that more effective filtering can be done by involving humans in the filtering process.

Using collaborative filtering to weave an information tapestry
D Goldberg, D Nichols, BM Oki, D Terry
Communications of the ACM 35 (12), 61-70

3887 1992



Collaborative filtering simply means that people collaborate to help one another perform filtering by recording their reactions to documents they read

Image from : http://dl.acm.org/citation.cfm?id=138867

Using collaborative filtering to weave an information tapestry D Goldberg, D Nichols, BM Oki, D Terry Communications of the ACM 35 (12), 61-70	3887	1992

Collaborative Filtering

- Collaborative filtering methods are based on collecting and **analyzing a large amount of information** on users' behaviors, activities or preferences and predicting what users will like based on their **similarity to other users**.

- Hypothesis: **Similar users tend to like similar items**.

- Requires a user community.

Collaborative Filtering

- R is the MxN rating matrix where M is the number of users and N the number of items.
- Rating can be defined in a variety of ways:
  - **Continuous ratings**: from -10 to 10
  - **Interval-based ratings**: 5 stars, 3 stars
  - **Ordinal ratings**: {strongly disagree, disagree, neutral, agree and strongly agree}
  - **Binary ratings**: Like/dislike
  - **Unary ratings**: Buy

Collaborative Filtering **Problems**:

- **Cold Start**: There needs to be enough other users already in the system to find a match.
- **Sparsity**: If there are many items to be recommended, even if there are many users, the user/ratings matrix is sparse, and it is hard to find users that have rated the same items.
- **First Rater**: Cannot recommend an item that has not been previously rated.
  - New items
  - Esoteric items
- **Popularity Bias**: Cannot recommend items to someone with unique tastes.
  - Tends to recommend popular items.

Two main approaches

1. **Memory-based methods**

   - **Neighbourhood-based methods** focused on computing the relationship between items or between users.

     - Two sub approaches:

       - **User based** and **Item based** Collaborative Filtering

2. **Model-based methods**

   - **Latent factor methods** that explains the rating by characterizing both items and users on many factors inferred from the rating pattern

1 - Neighbourhood methods

Neighborhood-based methods

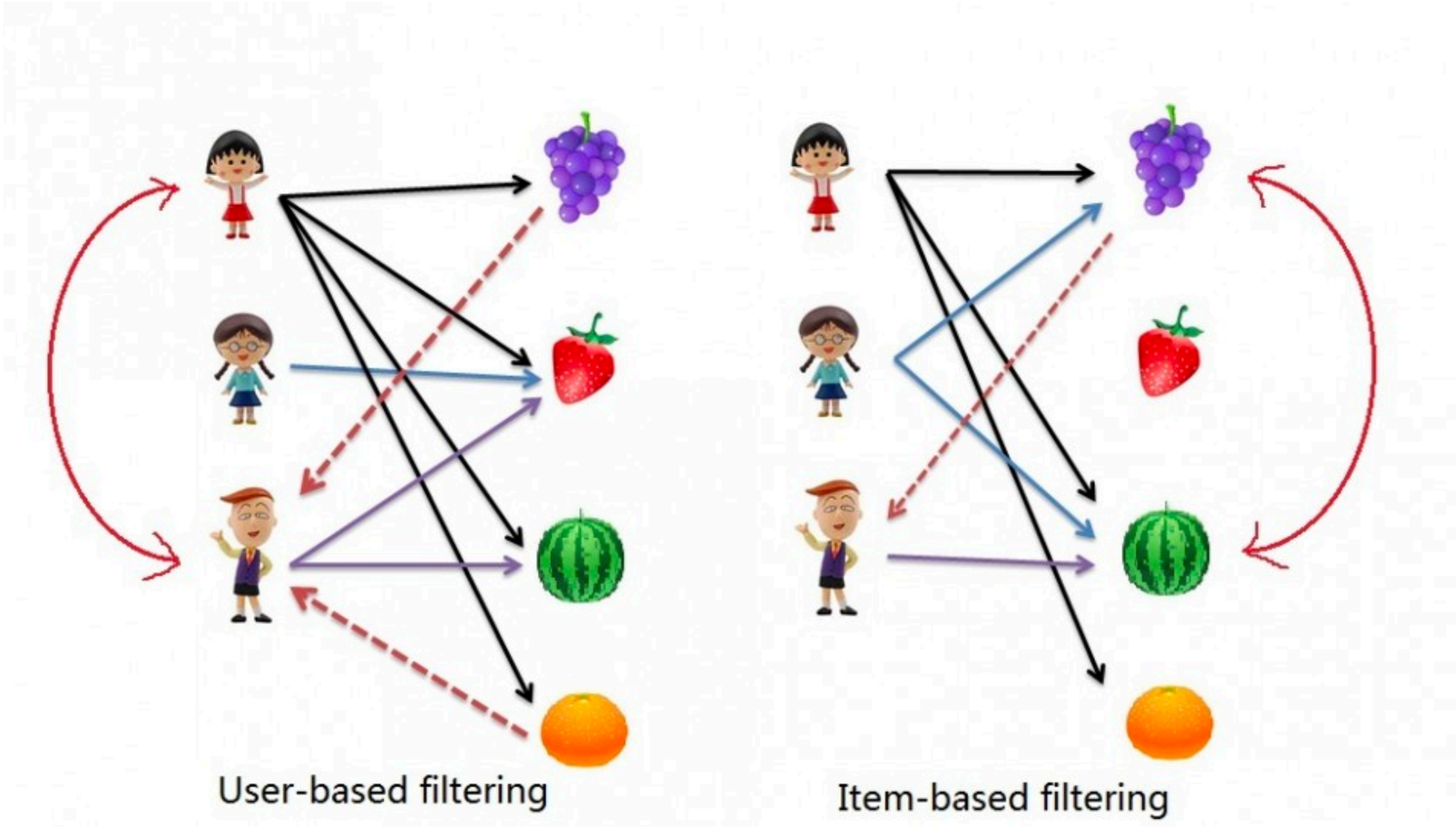
- Neighbourhood-based methods were among the earliest algorithms developed for collaborative filtering. These methods are based on the fact that similar users display similar patters or rating behaviors and similar items receive similar ratings.

- There are two primary types of neighbourhood -based algorithms:

  - **User-based** CF works like this: take a user U and a set of other users D whose ratings are similar to the ratings of the selected user U (user with similar preferences). And, use the ratings from those like-minded users to calculate a prediction for the selected user U.

  - In **Item-based** CF you build an item-item matrix determining relationships between pairs of items and using this matrix and data on the current user, infer the user's taste.

Neighborhood-based methods



Let’s see how we can create a **User-Based CF** for Movie recommendations.

Example: Movie Recommender System. User-Based Collaborative Filtering

- Given an "active **user**" and an **item** that has not been seen by the **user**, the goal is to estimate the rating for the **item**.

	Superman	Star Wars 1	Matrix	Spiderman
Santi	3	3.5	4.5	-
Jake	3.5	4	5	5
Anne	3	-	4.5	3
Caroline	3.5	5	3.5	2

The basic technique

- User-based nearest-neighbour collaborative filtering
  - Given an **"active user" (e.g. *Santi*)** and the items not yet seen (**e.g. *Spiderman***) the goal is to estimate Santi's rating for the those (e.g. ***Spiderman*)** items,
    - find a set of **users** (peers) **who have rated the item** (*Spiderman*) **and** has **similar tastes than the active user** (*Santi*) in the past and estimate the rating for the active user, e.g. the weighted averaged with similarly between the users and the given score.
    - Repeat this estimation for all items not seen by the active user (***Santi***) and recommend the best-rated

	Superman	Star Wars 1	Matrix	Spiderman
Santi	3	3.5	4.5	-
Jake	3.5	4	5	5
Anne	3	-	4.5	3
Caroline	3.5	5	3.5	2

What is needed?

- We need to define two things:

  1. How to compute **similarities** between users

  2. How to make the **prediction** using the similar users

How to measure **similarity** between users?

- The computation of the **similarity** between the items is one **critical step** in the CF algorithms.

- The basic idea in similarity computation between two users a and b is to first isolate the items commonly rated by both users (set P), and then to apply a similarity computation technique to determine the similarity.

Similarity Measures

- Euclidean Distance
- Pearson Correlation
- Person Correlation corrected
- Spearman Correlation
- Cosine Distance

How to measure similarity between users?

- Euclidean distance

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

- Pearson Correlation

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

- Cosine distance

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

Where:

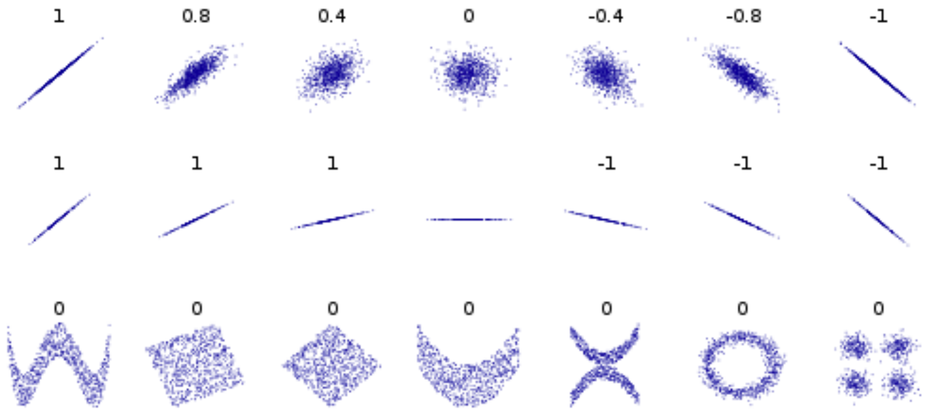
- $sim(a,b)$ is the similarity between user "a" and user "b"
- P is the set of common rated movies by user "a" and "b"
- $r_{a,p}$ is the rating of movie "p" by user "a"
- $\bar{r}_a$ is the mean rating given by user "a"

Similarity Measures: Euclidean distance

$$sim(u,v) = \sqrt{\sum_{j \in P} (r_{uj} - r_{vj})^2}$$

CAUTION: if the users use to rate with a different mean and standard deviation euclidean distance can give some problems

Similarity Measures: Pearson Correlation



Negative Values?

Strange correlations are rare,
and do not carries interesting information

Which is the best measure?

Hands on time!

Which is the best similarity function?

- …. there is not a clear answer…but, there are some tips:
  - Pearson Correlation used to work better than euclidean distance since it is based more on the ranking than on the values.
  - In general, Pearson Correlation coefficient is preferable to the raw cosine because of the bias adjustment effect of mean-centering
  - Cosine distance is usually used when our data is binary/unary, i.e. “like vs. not like” or “buy vs. not buy”.

Similarity Measures

- *Significance weighting*: When two users (or items) have **very few** items (or users) **in common** the **reliability** of the similarity scores if **low**. In these cases the similarity score can be reduced with a discount factor to de-emphasize the importance of that user pair.

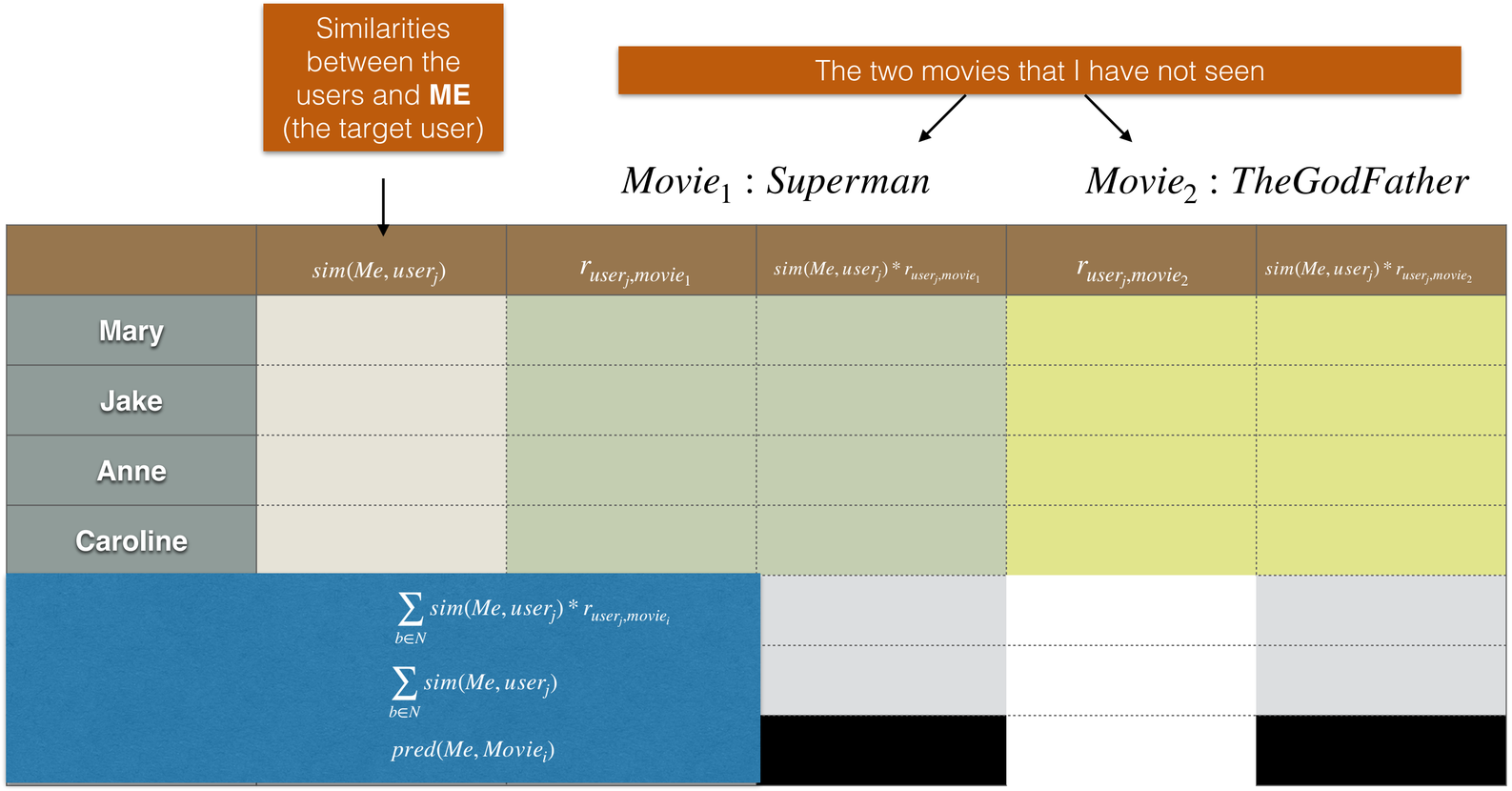
$$sim_c(u,v) = sim(u,v) \times \frac{min(50, |I_u \cap I_v|)}{50}$$

How do we generate a prediction from the neighbour's ratings?

$$\hat{r}_{u,j} = \frac{\sum_{v \in P_u(j)} sim(u,v) \times r_{v,j}}{\sum_{v \in P_u(j)} sim(u,v)}$$

Where $P_u(j)$ is denoted to the set of the top-k similar users to target user u, $sim(u,v)$ the similarity between user u and v and $r_{u,j}$ the rating of the user u to the movie j.

How do we generate a prediction from the neighbour's ratings?



How do we generate a prediction from the neighbour's ratings?

Similarities between the users and **ME** (the target user)

The two movies that I have not seen

$Movie_1 : Superman$ $Movie_2 : TheGodFather$

	$sim(Me, user_j)$	$r_{user_j,movie_1}$	$sim(Me, user_j) * r_{user_j,movie_1}$	$r_{user_j,movie_2}$	$sim(Me, user_j) * r_{user_j,movie_2}$
Mary	0.99	3	2.97	2.5	2.48
Jake	0.38	3	1.14	3	1.14
Anne	0.89	4.5	4.0	-	-
Caroline	0.92	3	2.77	3	2.77
	$\sum_{b \in N} sim(Me, user_j) * r_{user_j,movie_i}$		10.87		6.39
	$\sum_{b \in N} sim(Me, user_j)$		3.18		2.29
	$pred(Me, Movie_i)$		**3.41**		**2.79**

Other prediction functions

- **Different users** may provide ratings on **different scales**. Some users rate all items highly, whereas another rate all items negatively

$$\hat{r}_{u,j} = \bar{r}_u + \frac{\sum_{v \in P_u(j)} sim(u,v) \times (r_{v,j} - \bar{r}_v)}{\sum_{v \in P_u(j)} sim(u,v)}$$

Caution: predicted scores can goes outside the range.
However, the rank is correct.

Other prediction functions

$$\hat{r}_{u,j} = \bar{r}_u + \sigma_u \frac{\sum_{v \in P_u(j)} sim(u,v) \times z_{v,j}}{\sum_{v \in P_u(j)} |sim(u,v)|}$$

Where z_{uj} is the standardized rating computed as follows:

$$z_{uj} = \frac{r_{uj} - \bar{r}_u}{\sigma_u} = \frac{s_{uj}}{\sigma_u}$$

and $$\sigma_u = \sqrt{\frac{\sum_{j \in I_u} (r_{uj} - \bar{r}_u)^2}{|I_u| - 1}}$$

Caution: predicted scores can goes outside the range.
However, the rank is correct.

Some tricks

Some tricks (I)

- **Top-k** most similar users to the target user in order to do the predictions.
  - Weakly correlated users might add to the error in prediction, as well as, negative correlations often do not have a predictive value.

Some tricks (II)

Recursive methods: In order to avoid cold-start we can apply a recursive method for new users or for sparse data sets.

Some tricks (III)

Similarity amplification:

$$sim(u,v) = Pearson(u,v)^{\alpha}$$

Where $\alpha > 1$

Some tricks (IV)

- **Clustering** applied as a “first step” for **shrinking** the candidate set of users.
- Computing the similarity between users is computationally expensive.
- Clustering is cheaper to than computing the MxM similarity matrix
- Redefine top similar users using only the subset of users in the same cluster
- Problem: MxN is an incomplete, really sparse, matrix.

Some tricks (V): Long Tail Problem

- Difficult to provide good rating prediction for those item in the long tail.
- **Popular items** use to provide **less information about tastes** than non-popular items
- Usually, **popular** items provide **less profit** than non-popular.
- Usually, **non-popular** items generates more **surprise** to the user

Impact of the long Tail

Some movies are really popular since other very unpopular. **Popular items** can sometimes **worsen the quality** of the recommendations since they tend to be **less discriminative** across different users

Each item j can be weighted by w_j as follows

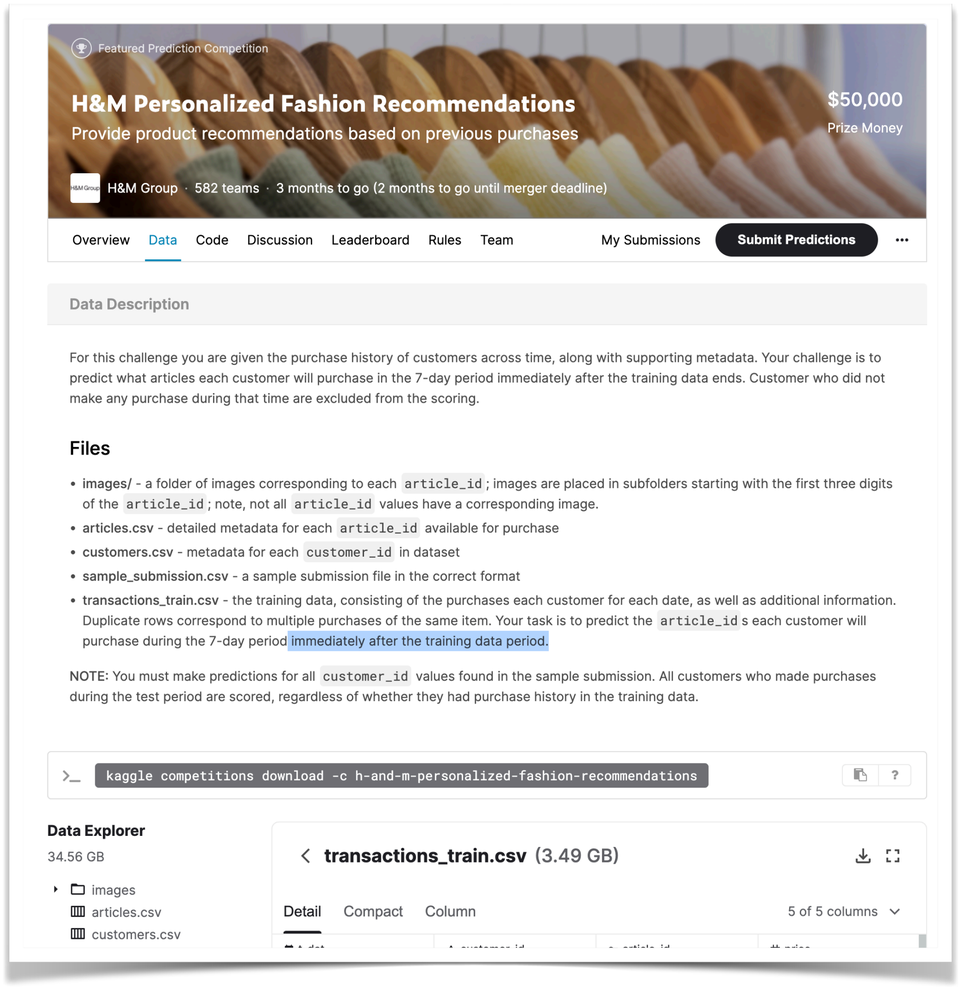
$$w_j = log\left(\frac{m}{m_j}\right)$$

where m is the total number of users, and m_j is the number of users who have rated the item j

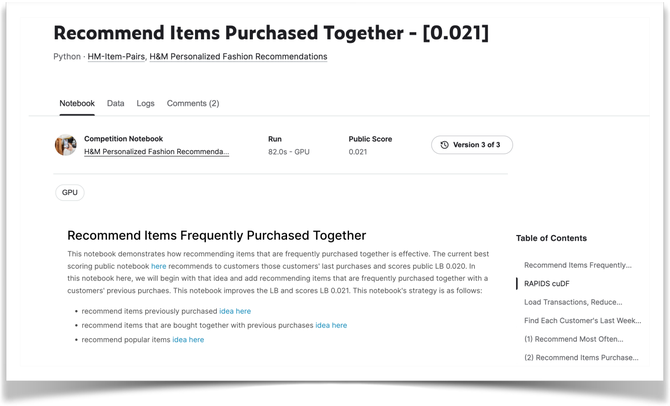
$$Pearson(u,v) = \frac{\sum_{k \in I_u \cap I_v} w_k (r_{uk} - \bar{r}_u)(r_{vk} - \bar{r}_v)}{\sqrt{\sum_{k \in I_u \cap I_v} w_k (r_{uk} - \bar{r}_u)^2} \sqrt{\sum_{k \in I_u \cap I_v} w_k (r_{vk} - \bar{r}_v)^2}}$$

Task:

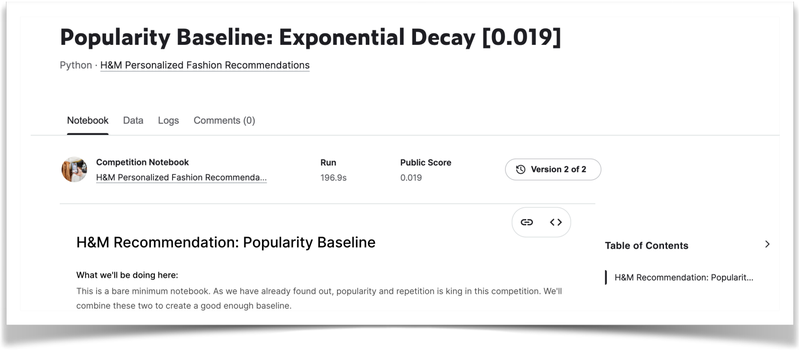
- Check these two baselines:



-

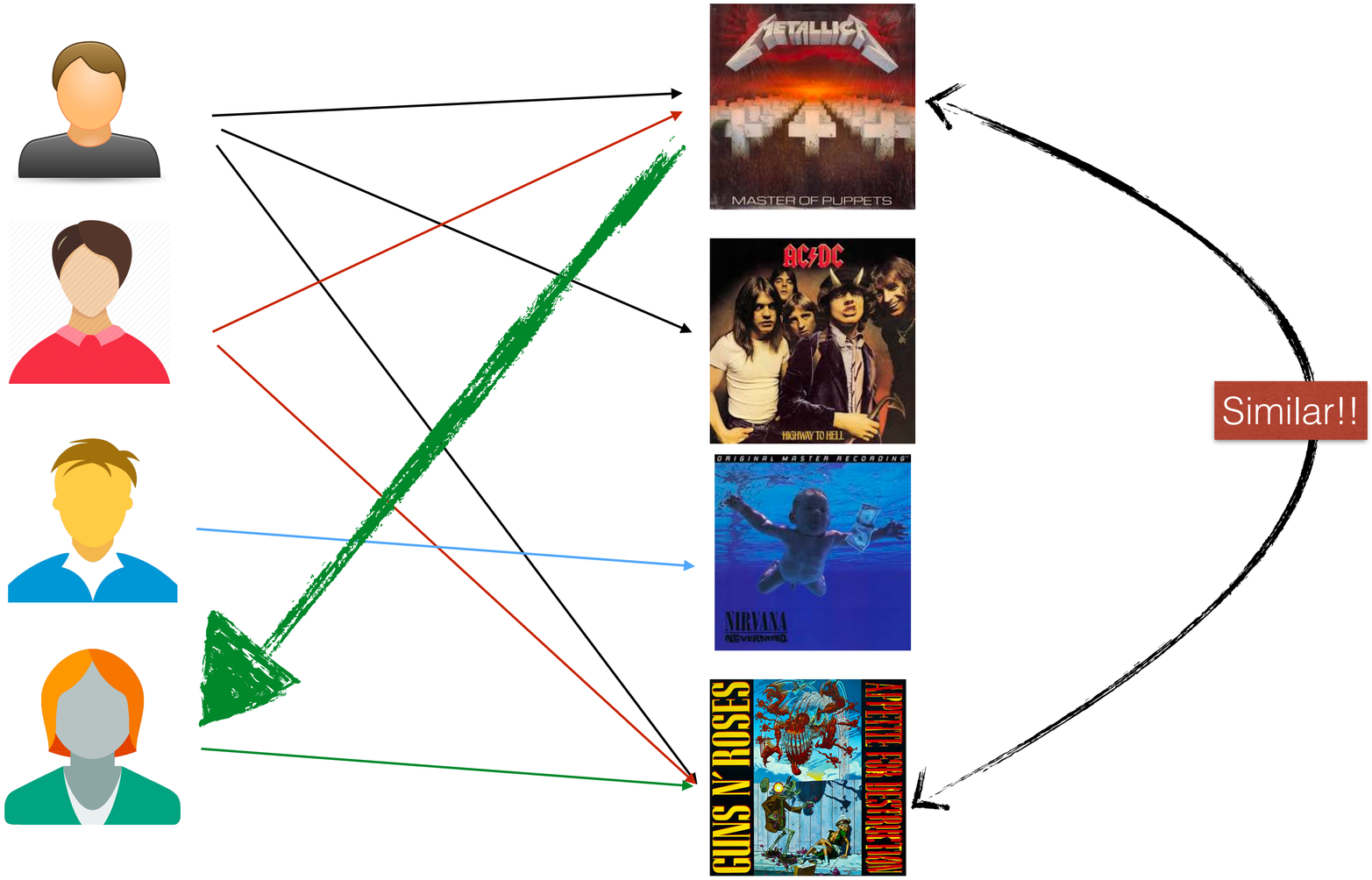


- https://www.kaggle.com/mayukh18/popularity-baseline-exponential-decay-0-019



- Can we apply a Collaborative RecSys? How?

Item-Based Recommender



Memory-based Methods

Let’s see how we can create an **Item-Based CF** for Movie recommendations.

Item-based Recommenders

- Instead on relying on the user similarity, prediction can rely on **item similarities**.
- Item similarity used to be **more stable** than user-similarity. So, the the update frequency of the items similarity is not as critical than user-similarity
  - Item-similarities are more static, while user-similarities are more dynamic

Item-based collaborative filtering recommendation algorithms B Sarwar, G Karypis, J Konstan, J Riedl Proceedings of the 10th international conference on World Wide Web, 285-295	5944	2001

Similarity Measures:
What happens with item-base systems?

- Pearson Correlation

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

- Cosine distance

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

Where:

- $sim(a,b)$ is the similarity between user "a" and user "b"
- P is the set of common rated movies by user "a" and "b"
- $r_{a,p}$ is the rating of movie "p" by user "a"
- $\bar{r}_a$ is the mean rating given by user "a"

Are these measures good?

Adjusted Cosine Similarity

- Computing similarity using basic cosine measure in item-based case has one important drawback: **The differences in rating scale between different users are not taken into account.**

- The Adjusted Cosine Similarity offsets this drawback by subtracting the corresponding user average from each co-rated pair:

$$sim(i,j) = \frac{\sum_{u \in U}(R_{u,i} - \bar{R}_u)(R_{u,j} - \bar{R}_u)}{\sqrt{\sum_{u \in U}(R_{u,i} - \bar{R}_u)^2}\sqrt{\sum_{u \in U}(R_{u,j} - \bar{R}_u)^2}}$$

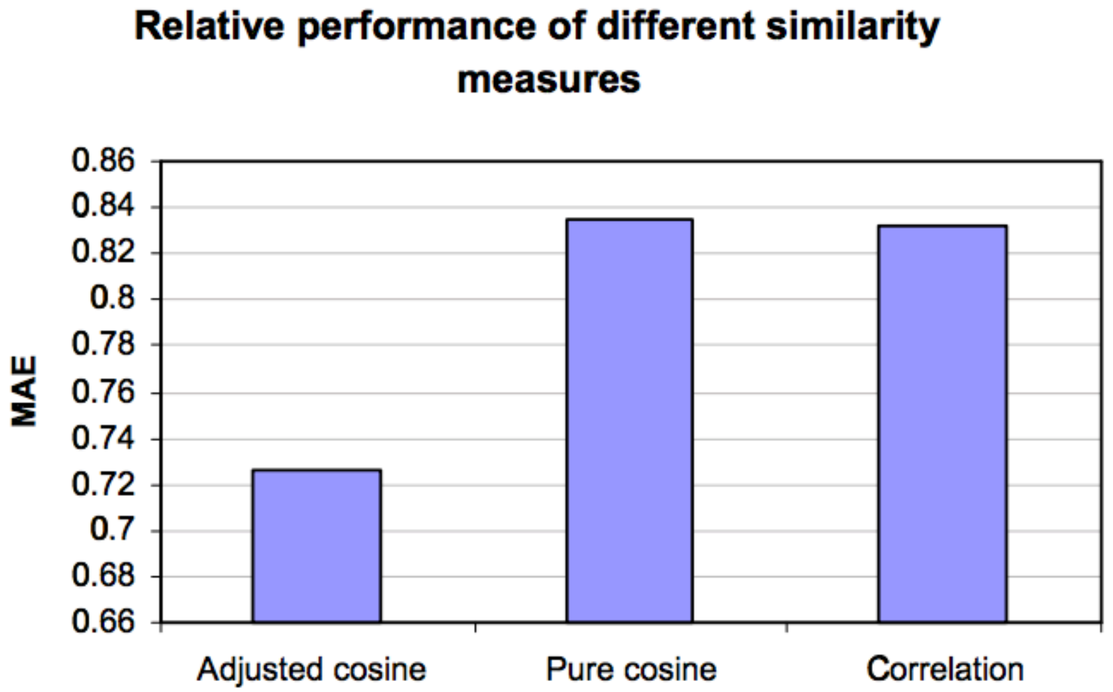


Figure 4: Impact of the similarity computation measure on item-based collaborative filtering algorithm.

Item-based collaborative filtering recommendation algorithms
B Sarwar, G Karypis, J Konstan, J Riedl
Proceedings of the 10th international conference on World Wide Web, 285-295

5944 2001

Item-Based:
How do we generate a prediction?

$$\hat{r}_{u,j} = \bar{r}_u + \frac{\sum_{v \in P_u(j)} sim(u,v) \times (r_{v,j} - \bar{r}_v)}{\sum_{v \in P_u(j)} sim(u,v)}$$

Why not another equation?

User-Based vs. Item-Based

- m = #users ; n = #items
- Normally, the number of users is much bigger than the number of items.

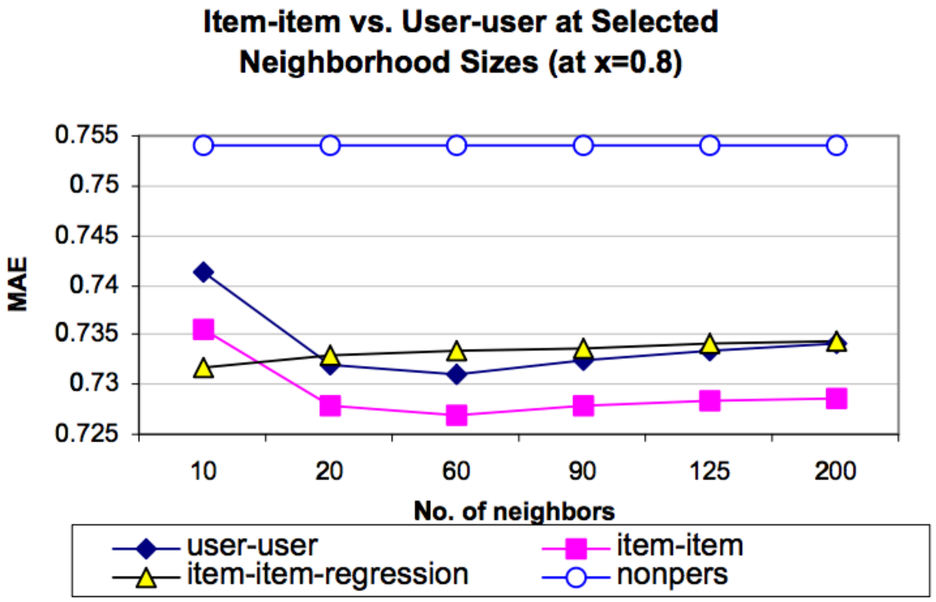
Computational time:

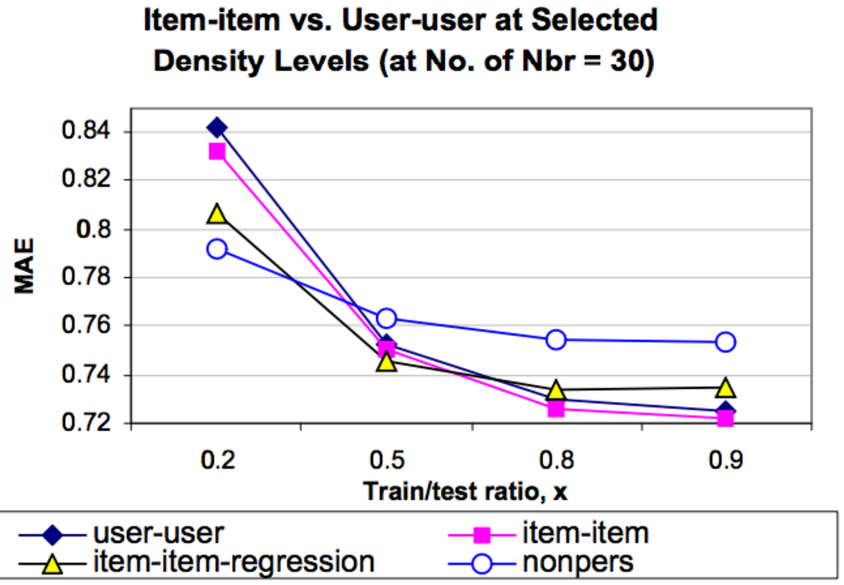
O(m^2 n)	O(n^2 m)

Memory Requirements:

O(m^2)	O(n^2)

User-Based vs. Item-Based





Item-based collaborative filtering recommendation algorithms
B Sarwar, G Karypis, J Konstan, J Riedl
Proceedings of the 10th international conference on World Wide Web, 285-295
5944 2001

User-Based vs. Item-Based

- **Pros User-based**
  - Tend to provide higher diversity (more serendipity)
- **Pros Item-based**
  - Better results (in terms of RMSE)
  - More stable to changes

User-Based vs. Item-Based

	User-Based	Item-Based
Scalability		
Explanation		
Novelty		
Coverage		
Cold start		
Performance		

User-Based vs. Item-Based

	User-Based	Item-Based
Scalability	Bad when #users is huge	Bad when #items is huge
Explanation	Bad	Good
Novelty	Bad	Good
Coverage	Bad	Good
Cold start	Bad for new users	Bad for new items
Performance	Need to get many users history	Only need to get current users's history

Item-Based Nearest Neighbor - **Regression**

- We can replace the (normalized) similarity coefficient $AdjustedCosine(j, i)$ with an unknown parameter.

  w_{ji}^{item} to model the rating prediction of a user u for target item i.

$$\hat{r}_{ui} = \sum_{j \in Q_i(u)} w_{ij}^{item} \cdot r_{uj}$$

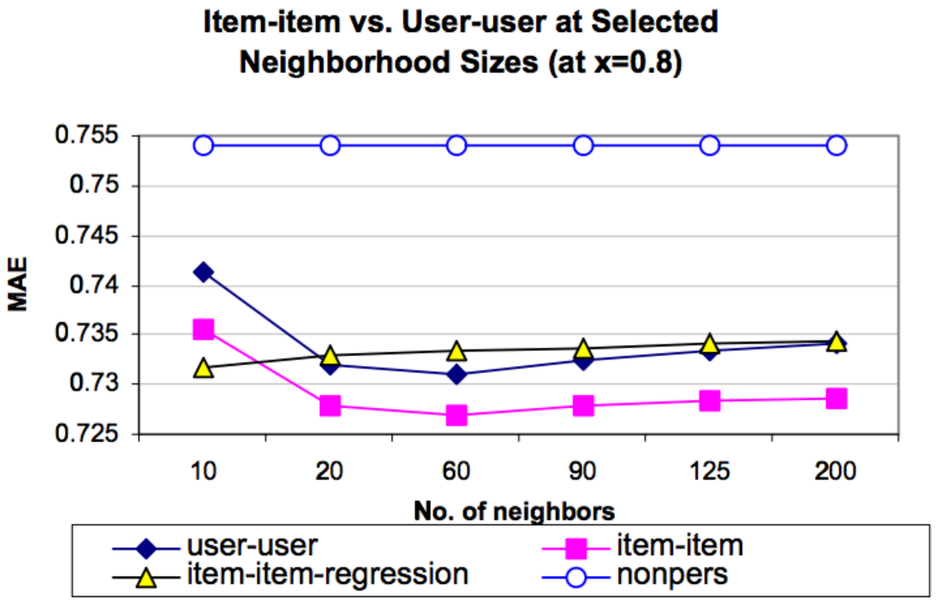
$Q_i(u)$ is the most similar item to i that have been seen by user u
The nearest items in $Q_i(u)$ can be determined using the adjusted cosine.

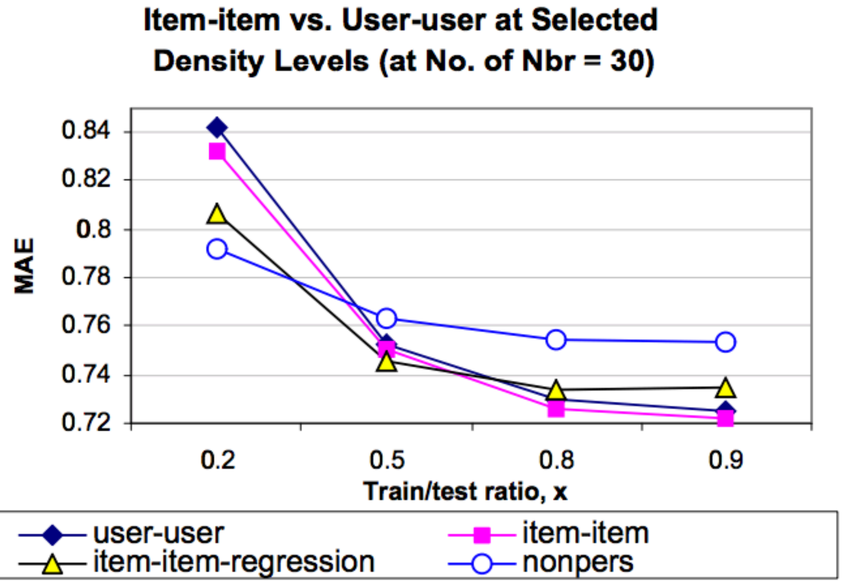
Item-Based Nearest Neighbor - **Regression**

$$Minimize\ J_t = \sum_{u \in U_t} (r_{ut} - \hat{r}_{ut})^2$$

$$= \sum_{u \in U_t} \left(r_{ut} - \sum_{j \in Q_t(u)} w_{jt}^{item} \cdot r_{uj} \right)^2$$

User-Based vs. Item-Based





Item-based collaborative filtering recommendation algorithms
B Sarwar, G Karypis, J Konstan, J Riedl
Proceedings of the 10th international conference on World Wide Web, 285-295
5944 2001

Dimensionality Reduction

- Pairwise similarities are hard to robustly be computed in sparse matrices.
- Dimensionality reduction can be used to **improve** neighborhood-based methods both in terms of **quality** and in terms of **efficiency**
- A reduced representation of the data can be created in terms of either row-wise latent factors or in terms of column-wise latent factors.

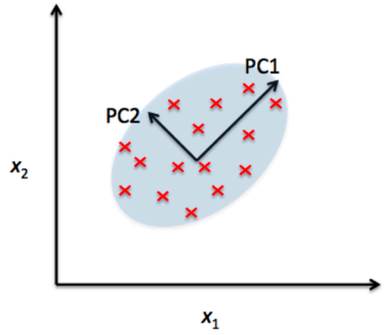
Dimensionality Reduction and Neighborhood Methods

- Dimensionality reduction can **improve** neighborhood methods in terms of **accuracy** and also in terms of **efficiency**.

- Similarities are hard to be computed in huge dimensional sparse rating matrices.

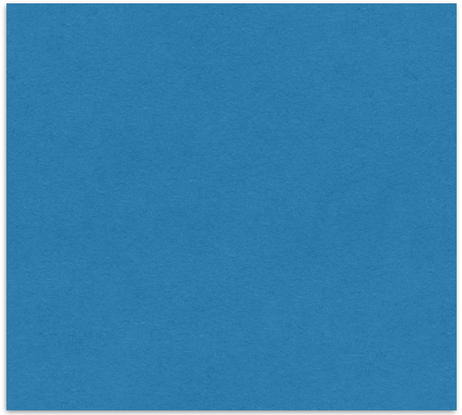
  - Latent factor models

Dimensionality Reduction



Items

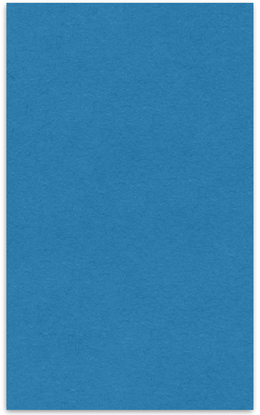
users



m x n **sparse matrix**

≈

users



m x d **matrix**
where d<<n

Dimensionality Reduction

- The low-dimensional representation can be computed using **PCA** or **SVD-Like** methods.
- After the d-dimensional representation of each user is estimated, the similarity between users can be computed
- Cosine or dot product on the reduced vectors can be used in order to compute the similarity
- More robust since the feature vector is fully specified
- More efficient

Dimensionality Reduction

- How to **obtain** the **d-dimensional representation** on the sparse matrix?
- **SVD Method**. Steps:
  - Augment the *m x n* incomplete rating matrix R -> R_f
    - Mean-user rating or mean-item rating for each row/column
  - Let's define the similarity matrix S as $\mathbf{S = R_f^T R_f}$. S is a positive semi-definite of size *n x n*
  - Determine the dominant basis vectors of R_f by computing the **diagonalization** of the similarity matrix S.
    - $S = P\Lambda P^T$, where P is an *n x n* matrix, whose columns contain the orthonormal eigenvectors of S. Λ is a diagonal matrix containing the non-negative eigenvalues of S along its diagonal.
  - Let denote P_d the *n x d* matrix only containing the columns of P with the largest eigenvalues
  - The low representation of R is obtained by the multiplication of $\mathbf{R_f P_d}$

Dimensionality Reduction

- How to **obtain** the **d-dimensional representation** on the sparse matrix?
- **PCA Method**. Steps:
  - Augment the *m x n* incomplete rating matrix R -> R_f
    - Mean-user rating or mean-item raring for each row/column
  - Lets define the similarity matrix S as **the Covariance Matrix of Rf**
  - Determine the dominant basis vectors of R_f by computing the diagonalization of the similarity matrix S.
    - $S = P\Lambda P^T$, where P is an *n x n* matrix, whose columns contain the orthonormal eigenvectors of S. Λ is a diagonal matrix containing the non-negative eigenvalues of S along its diagonal.
  - Let denote P_d the *n x d* matrix only containing the columns of P with the largest eigenvalues
  - The low representation of R is obtained by the multiplication of $\mathbf{R_f P_d}$

Dimensionality Reduction

- Challenges:
  - Missing Values
    - Need a way to fill it
    - Several alternatives, including clever averages and predictions
  - Computational Complexity
  - Lack of transparency/explainability

Explaining Recommendations

- Help on:
  - Transparency
  - Trust

A survey of explanations in recommender systems
N Tintarev, J Masthoff
Data Engineering Workshop, 2007 IEEE 23rd International Conference on, 801-810

226 2007

What is an explanation

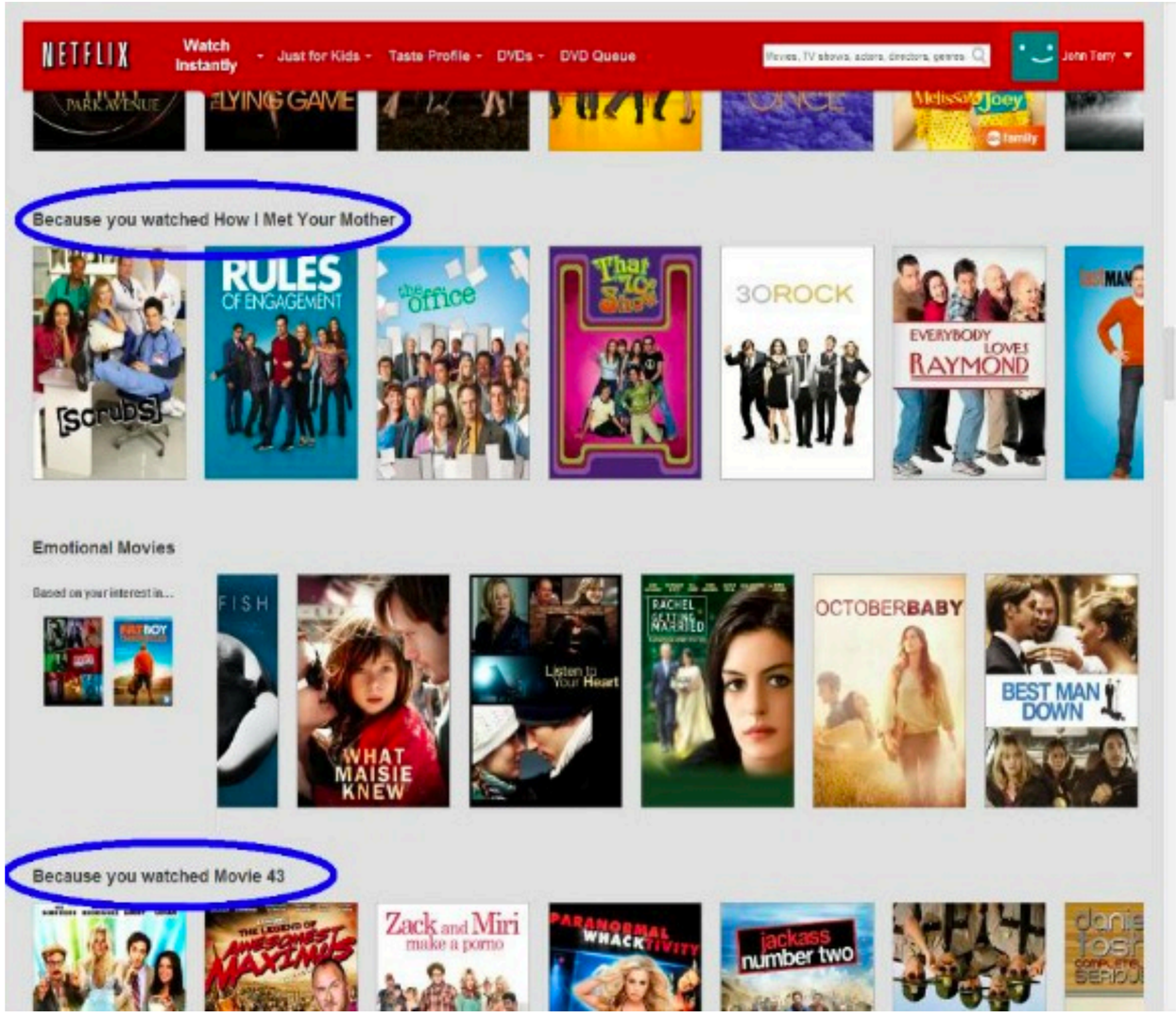
- Additional data to help users understand a specific recommendation
  - It is totally separate from an explanation how the system works as a whole
- Some explanation are confidence values
- Show distributions
- Sometimes tied to ability to edit profile to **improve recommendations**

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

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

Figure 1. One of the twenty-one different explanation interfaces given shown in the user survey. Notice that the title has been encoded, so that it does not influence a user's decision to try a movie.

Explaining collaborative filtering recommendations
JL Herlocker, JA Konstan, J Riedl
Proceedings of the 2000 ACM conference on Computer supported cooperative ... 1264 2000


NETFLIX
Watch Instantly
Just for Kids
Taste Profile
DVDs
DVD Queue
John Terry
Because you watched How I Met Your Mother
RULES OF ENGAGEMENT
the office
30ROCK
EVERYBODY LOVES RAYMOND
Emotional Movies
Based on your interest in...
FISH
WHAT MAISIE KNEW
Listen to Your Heart
OCTOBERBABY
BEST MAN DOWN
Because you watched Movie 43
Zack and Miri make a porno
PARANORMAL WHACKTIVITY
jackass number two