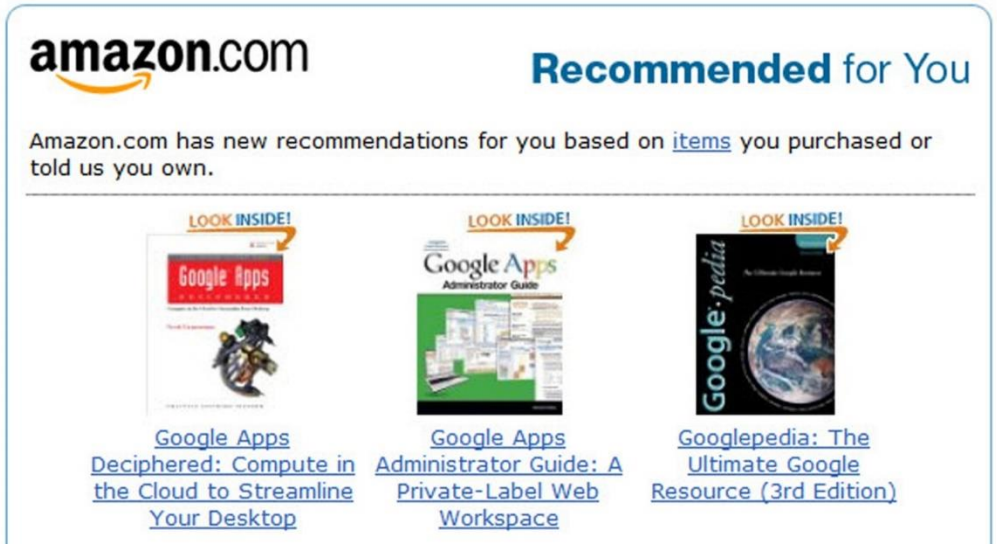
Subreddit Recommendation System

Bhavesh Bhatt

What is a Recommender System

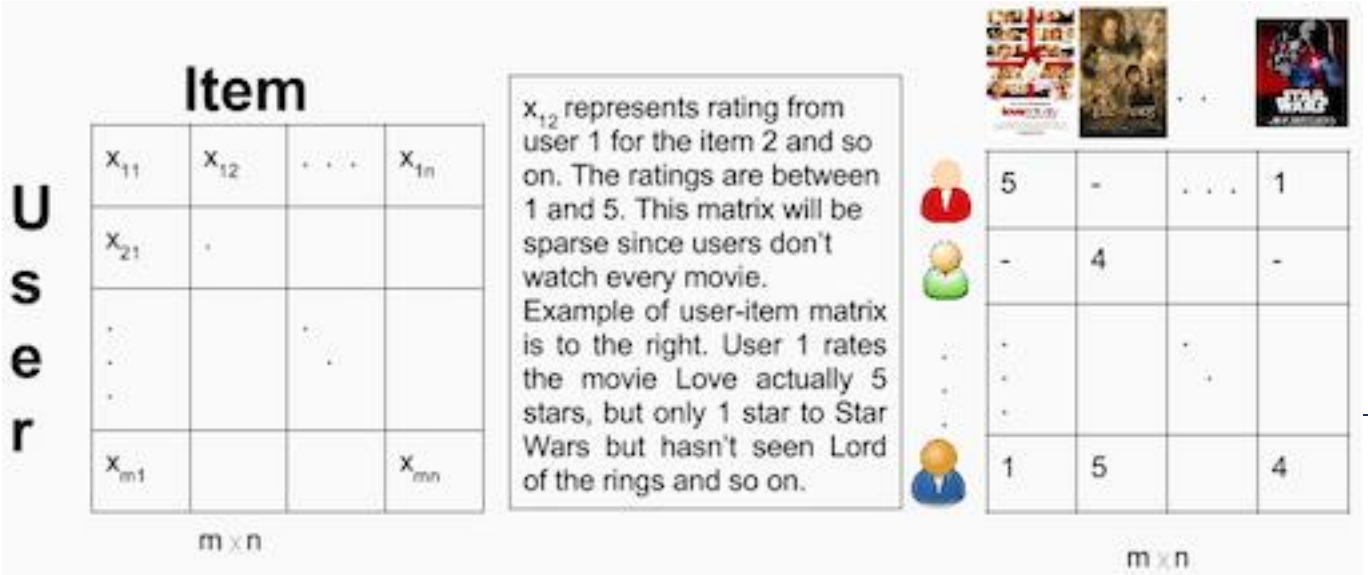
- Recommender systems apply statistical and knowledge discovery techniques to the problem of making product recommendations (Sarwar *et al.*, 2000)

- **Advantages of recommender systems**
  - Improve conversion rate: Help customers find a product she/he wants to buy.
  - Cross-selling: Suggest additional products.
  - Improve customer loyalty: Create a value-added relationship



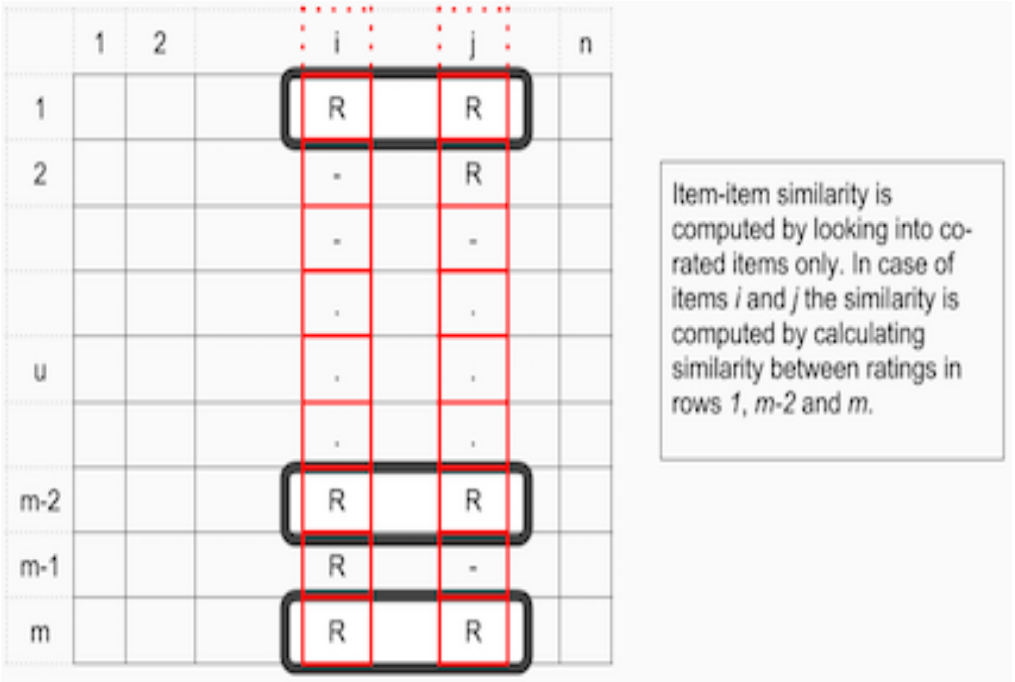
Memory-Based Collaborative Filtering

- **Item-Item Collaborative Filtering**
  - Users who liked this item also liked.
- **User-Item Collaborative Filtering**
  - Users who are similar to you also liked.
- **Memory based collaborating filtering -**
  - In both cases, we create a user-item matrix which you build from the entire datasets.



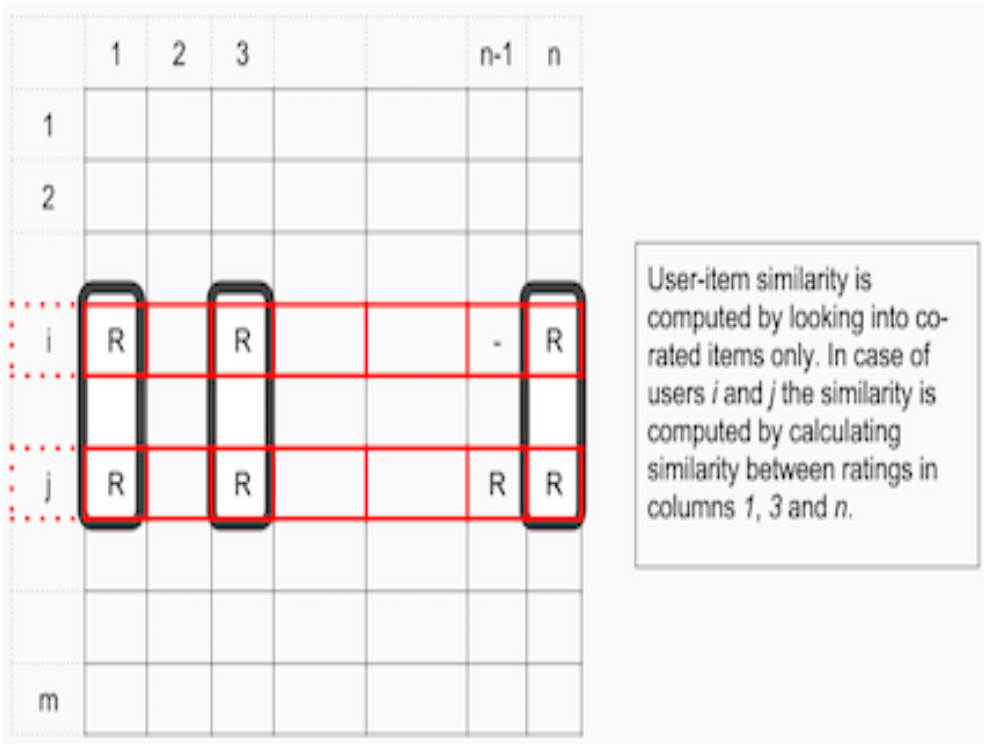
Item-Item Collaborative Filtering

- *Item-Item Collaborative Filtering* are measured by observing all the users who have rated both items.



User-Item Collaborative Filtering

- *User-Item Collaborative Filtering* the similarity values between users are measured by observing all the items that are rated by both users.



Spending some time with the Subreddit dataset

```
subreddit_df = read_csv('reddit_data.csv')
print "Top 5 rows of the dataset - \n"+ str(subreddit_df.head())
```

```
Top 5 rows of the dataset -
    username         subreddit           utc
0  kabanossi  photoshopbattles  1.482748e+09
1  kabanossi      GetMotivated  1.482748e+09
2  kabanossi            vmware  1.482748e+09
3  kabanossi           carporn  1.482748e+09
4  kabanossi               DIY  1.482747e+09
```

```
n_users = subreddit_df.username.unique().shape[0]
n_items = subreddit_df.subreddit.unique().shape[0]
print 'Number of users = ' + str(n_users) + '\nNumber of subreddits = ' + str(n_items)
```

```
Number of users = 22610
Number of subreddits = 34967
```

```
print "Number of rows in the dataset = " +str(subreddit_df.shape[0])+ \
'\nNumber of columns = ' + str(subreddit_df.shape[1])
```

```
Number of rows in the dataset = 14000000
Number of columns = 3
```

Spending some more time with the Subreddit dataset

In [12]:
```
subreddit_grouped['percentage']  = subreddit_grouped['subreddit_count'].div(grouped_sum)*100
subreddit_grouped.sort_values(['subreddit_count', 'subreddit'], ascending = [0,1]).head(10)
```

Out[12]:

	subreddit	subreddit_count	percentage
1402	AskReddit	1030290	7.359214
29812	politics	367860	2.627571
17058	The_Donald	216939	1.549564
28536	nfl	173883	1.242021
26783	leagueoflegends	157663	1.126164
34646	worldnews	156605	1.118607
24419	funny	152921	1.092293
28380	nba	150985	1.078464
29564	pics	143496	1.024971
28497	news	140492	1.003514

Lets calculate the sparsity of the dataframe

```
sparsity=round(1.0-len(subreddit_df)/float(len(users)*len(user_queries)),3)
print 'The sparsity level of Subreddit dataframe is ' +  str(sparsity*100) + '%'
```

```
The sparsity level of Subreddit dataframe is 98.2%
```

Matrix factorization

- Informally, the SVD theorem (Golub and Kahan 1965) states that a given matrix M can be decomposed into a product of three matrices as follows

$$M = U \times \Sigma \times V^T$$

  - where U and V are called *left* and *right singular vectors* and the values of the diagonal of Σ are called the *singular values*
- We can approximate the full matrix by observing only the most important features – those with the largest singular values
- In the example, we calculate U, V, and Σ (with the help of some linear algebra software) but retain only the two most important features by taking only the first two columns of U and V^T

Singular value decomposition (SVD)

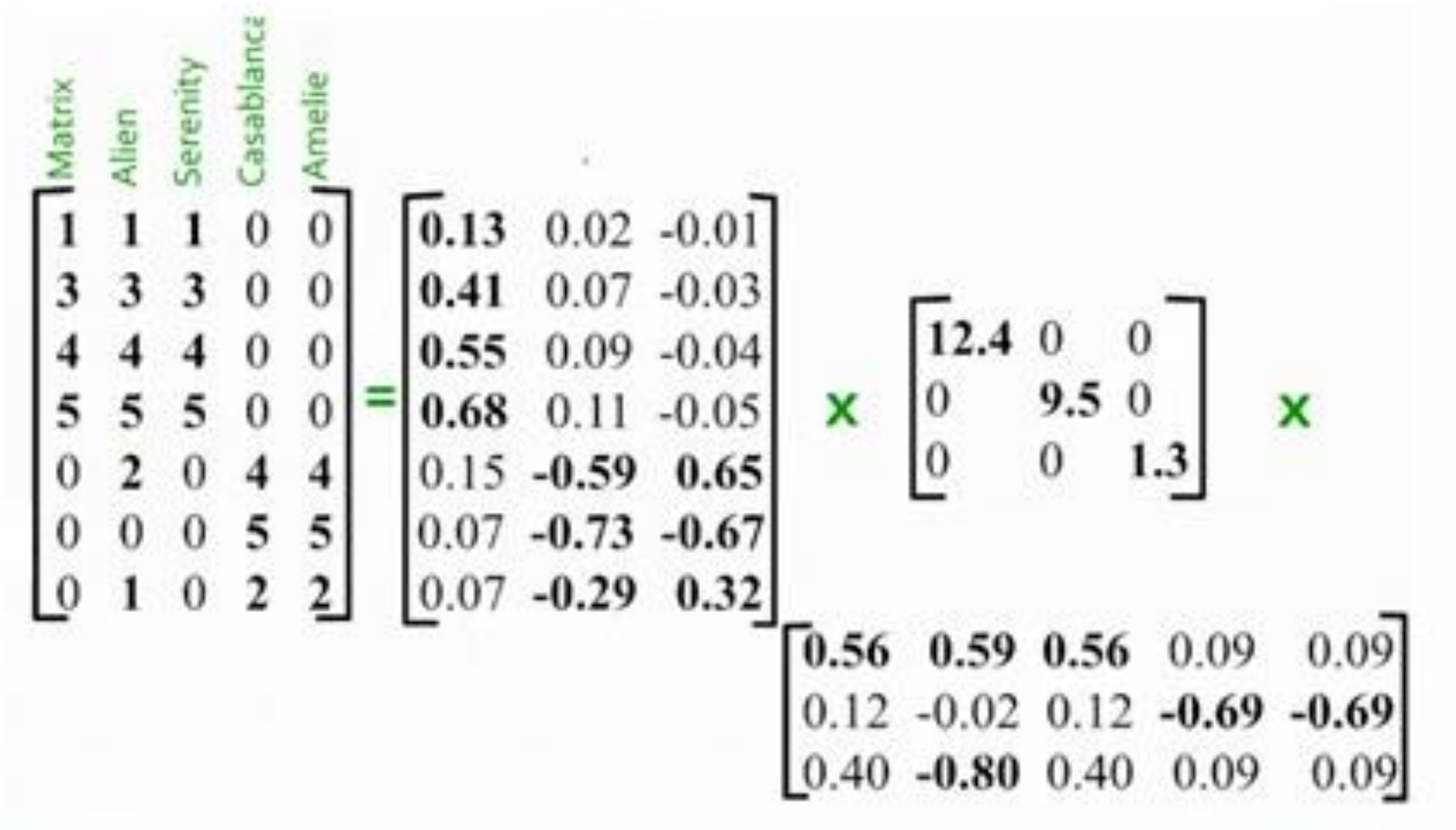
- **A well-known matrix factorization method is Singular value decomposition (SVD).**
- **Collaborative Filtering can be formulated by approximating a matrix X by using singular value decomposition.**
- **The winning team at the Netflix Prize competition used SVD matrix factorization models to produce product recommendations.**

Singular value decomposition (SVD) (2)

$$A_{[m \times n]} = U_{[m \times r]} \Sigma_{[r \times r]} (V_{[n \times r]})^T$$

- **A**: Input data matrix
  - m x n matrix (e.g., m documents, n terms)
- **U**: Left singular vectors
  - m x r matrix (m documents, r concepts)
- Σ: Singular values
  - r x r diagonal matrix (strength of each 'concept') (r : rank of the matrix **A**)
- **V**: Right singular vectors
  - n x r matrix (n terms, r concepts)

Singular value decomposition (SVD) (3)



Example for SVD-based recommendation

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

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

V_k^T	Terminator	Die Hard	Twins	Eat Pray Love	Pretty Woman
Dim1	-0.44	-0.57	0.06	0.38	0.57
Dim2	0.58	-0.66	0.26	0.18	-0.36

Σ_k	Dim1	Dim2
Dim1	5.63	0
Dim2	0	3.23

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

= 3 + 0.84 = 3.84

Making predictions

- Now you can make a prediction by taking dot product of U, S and VT depending on the latent factors.
- Since SVD makes recommendation based on the high varying concept vector. There are chances of the high varying concept already being consumed by the end user.
- So in order to provide accurate recommendations, its often useful to drop the recommendations already in the users history.

Some Observation

```
------------------------------------------------------------------

User for whom recommendations are needed: Podiumqueen

------------------------------------------------------------------

Previous Subreddit interactions -

pics
AskReddit
niceguys
childfree
asperger

------------------------------------------------------------------


------------------------------------------------------------------------

Recommendation for Podiumqueen are as follows -

------------------------------------------------------------------------

TwoXChromosomes
ProductTesting
Bowling
creepyPMs
talesfromtechsupport
------------------------------------------------------------------------
```

Some Observation (2)

```
np.where(user_queries == 'AskReddit')
```

```
(array([71]),)
```

```
user_subreddit_matrix = sp.csr_matrix(user_subreddit_matrix)
```

```
user_subreddit_matrix[8491,71] = 0
```

```
--------------------------------------------------------------------------------

User for whom recommendations are needed: Podiumqueen

--------------------------------------------------------------------------------

Previous Subreddit interactions -

pics
niceguys
childfree
asperger

--------------------------------------------------------------------------------
```

Some Observation (3)

```
--------------------------------------------------------------------------------

Recommendation for Podiumqueen are as follows -

--------------------------------------------------------------------------------

AskReddit
TwoXChromosomes
ProductTesting
Bowling
creepyPMs
--------------------------------------------------------------------------------
```

- As you see AskReddit subreddit was already in the history of the user Podiumqueen and when forcefully I made the count of AskReddit 0 it got removed from the history.
- However when next time a recommendation was suggested for Podiumqueen, AskReddit turns up as part of the recommended output.


Thank
You
reddit
♡Lucas