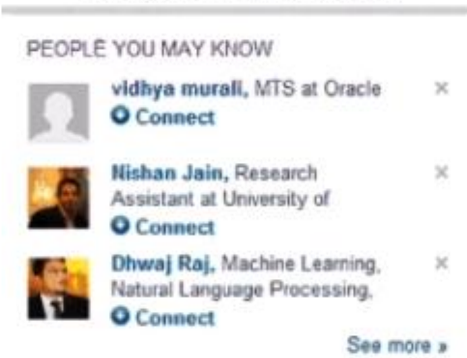
Recommendation Systems

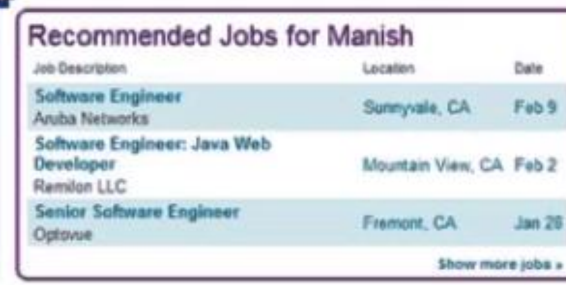
Recommendation systems are everywhere

LinkedIn People Recommendations



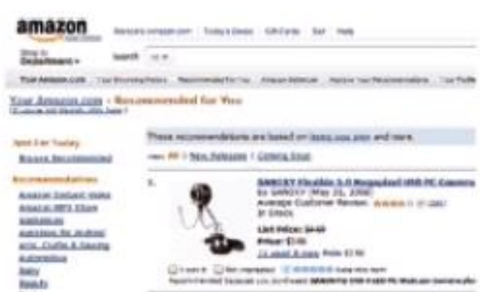
Facebook People Recommendations

HotJobs Job Recommendations



Bing Query Recommendations

Amazon Product Recommendations



Netflix Movie Recommendations

Recommendation Systems

Social overload

- Facebook – largest social network site
  - 600,000,000 users, half login every day
  - 35,000,000,000 online "friendships"
  - 900,000,000 objects people interact with
  - 30,000,000,000 shared content items / month
- YouTube – largest video sharing site
  - 2,000,000,000 views per day
  - 1,000,000 video hours uploaded per month
- Twitter – largest microblogging site
  - 200,000,000 users per month
  - 65,00,000 tweets per day (750 per second)
  - 8,000,000 followers of most popular user

Recommendation Systems

Social overload

- Information Overload
  - Blogs, microblogs, forums, wikis, news, bookmarked webpages, photos, videos, etc.
- Interaction Overload
  - Friends, followers, followees, commenters, co-members, voters, likers, taggers, review writers, etc.

Recommendation Systems

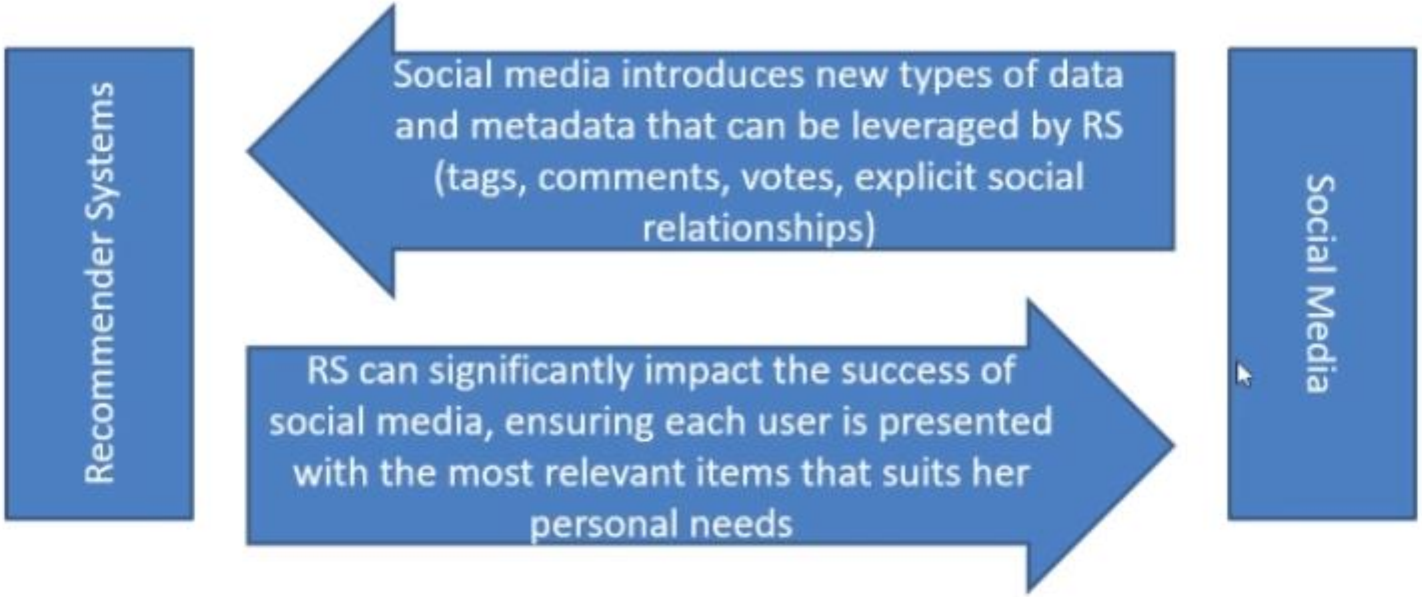
Social Recommender Systems

- Recommender Systems that target the social media domain
- Aim at coping with the challenge of social overload by presenting the most attractive and relevant content
- Also aim at increasing adoption and engagement
- Often apply personalization techniques

Recommendation Systems

Recommender Systems & Social Media

- Recommender Systems are an augmentation of the social process, in which we rely on advices or suggestions from other people
- Social Media and Recommender Systems can mutually benefit each other



Recommendation Systems

Fundamental Recommendation Approaches

- Collaborative filtering based Recommendation
  - Aggregate ratings of objects from users and generate recommendation based on inter-user similarity
- Demographic Recommendation
  - Categorize users based on personal attributes (age, gender, income..) and make recommendation based on demographic classes
- Content-based recommendation
  - A user profile is constructed based on the features of the items the user has rated/consumed. This profile is used to identify new interesting items for the user (that match his profile)
- Hybrid methods
  - Combine several approaches together

Recommendation Systems

Collaborative Filtering

- In the real world we seek advices from our trusted people (friends, colleagues, experts)
- CF automates the process of "word-of-mouth"
  - Weight all users with respect to similarity with the active user.
  - Select a subset of the users (neighbors) to use as recommenders
  - Predict the rating of the active user for specific items based on its neighbors' ratings
  - Recommend items with maximum prediction

Recommendation Systems

User based Collaborative Filtering Algorithm

- The User x Item Matrix

	Shrek	Snow-white	Superman
Alice	Like	Like	Dislike
Bob	?	Dislike	Like
Chris	Like	Like	Dislike
John	Like	Like	?

- Shall we recommend Superman for John?
- John’s taste is similar to both Chris and Alice tastes ⇒ Do not recommend Superman to John

Recommendation Systems

User based Collaborative Filtering Algorithm

- Let R be the rating matrix
  - r_{uj} is then the vote of user u for item j
- I_u be the set of items for which user u has provided the rating
- Voting
  - Mean vote for user u: $\bar{r_u} = \frac{1}{|I_u|}\sum_{i \in I_u} r_{ui}$
  - Prediction rating: $p_{uj} = \bar{r_u} + \gamma \sum_{v=1}^{n} w(u,v)(r_{vj} - \bar{r_v})$
    - $w(u,v)$ = similarity between users u and v
    - γ is a normalization constant $\gamma = \frac{1}{\sum_{v=1}^{n} w(u,v)}$

Recommendation Systems

Similarity Functions

- Cosine based similarity between users
  - $w(u,v) = \frac{\sum_{i\in I} r_{ui} r_{vi}}{\sqrt{\sum_{i\in I} r_{ui}^2}\sqrt{\sum_{i\in I} r_{vi}^2}}$
- Pearson based similarity between users
  - $w(u,v) = \frac{\sum_{i\in I}(r_{ui}-\overline{r_u})(r_{vi}-\overline{r_v})}{\sqrt{\sum_{i\in I}(r_{ui}-\overline{r_u})^2}\sqrt{\sum_{i\in I}(r_{vi}-\overline{r_v})^2}}$

Recommendation Systems

Collaborative Filtering: Practical Challenges

- Ratings data is often sparse, and pairs of users with few co-ratings are prone to skewed correlations
- Fails to incorporate agreement about an item in the population as a whole
  - Agreement about a universally loved item is much less important than agreement for a controversial item
    - Some algorithms account for global item agreement by including weights inversely proportional to an item's popularity
- Calculating a user's perfect neighborhood is expensive – requiring comparison against all other users
  - Sampling: a subset of users is selected prior to prediction computation
  - Clustering: can be used to quickly locate a user's neighbors

Item-Based Nearest Neighbor Algorithm

- The transpose of the user-based algorithms
  - Generate predictions based on similarities between items
  - The prediction for an item is based on the user's ratings for similar items

	Shrek	Snow-white	Superman
Alice	Like	Like	Dislike
Bob	?	Dislike	Like
Chris	Like	Like	Dislike
John	Like	Like	?

- Bob dislikes Snow-white (which is similar to Shrek) ⇒ do not recommend Shrek to Bob
- Predicted rating: $p_{uj} = \gamma \sum_{i=1}^{m} w(i,j) r_{ui}$
- Traverse over all m items rated by user u and measure their rating, averaged by their similarity to the predicted item
- w(i,j) is a measure of item similarity - usually the cosine measure
- Average correction is not needed because the component ratings are all from the same target user