IIIT-H

Web Mining
Lecture 7: Social Recommender Systems (Part 1)

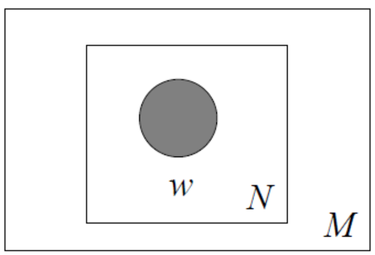
Manish Gupta

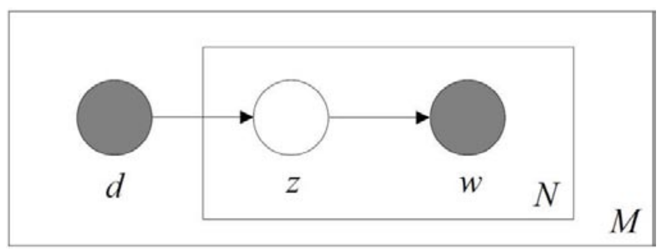
21st Aug 2013

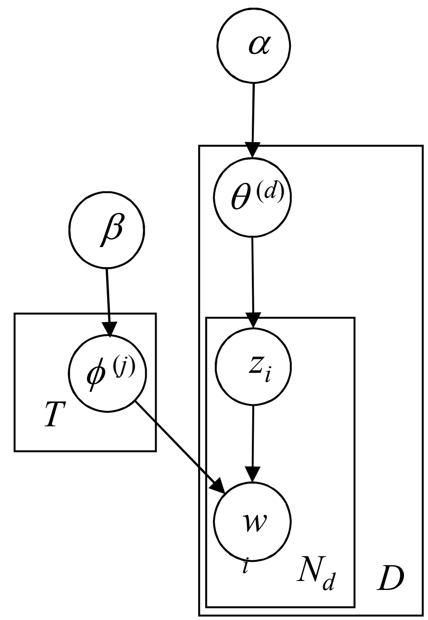
Slides borrowed (and modified) from
http://www.slideshare.net/idoguy/social-recommender-systems-tutorial-www-2011-7446137

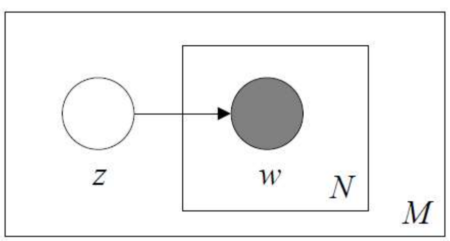
Recap of Lecture 6: Topic Models

- Probabilistic Latent Semantic Analysis (PLSA)
- Latent Dirichlet Allocation (LDA)
- Other Topic Models









Announcements

- Assignment 1 deadline is tomorrow (Aug 22, 2013).
  - Do not upload data
  - Upload only README and code
  - Assignment evaluation session on Friday, 23rd August 2013 6:30 pm to 8:00 pm at SIEL Lab 2
- Rescheduling of lectures
  - Makeup class for Aug 24 lecture will be on Aug 22 6-7:30pm
  - Makeup class for Aug 28 lecture will be on Sep 2 6-7:30pm

Today's Agenda

- Introduction to Recommender Systems
- Fundamental Recommendation Approaches
- Content Recommendation
- Tag Recommendation
- People Recommendation
- Community Recommendation

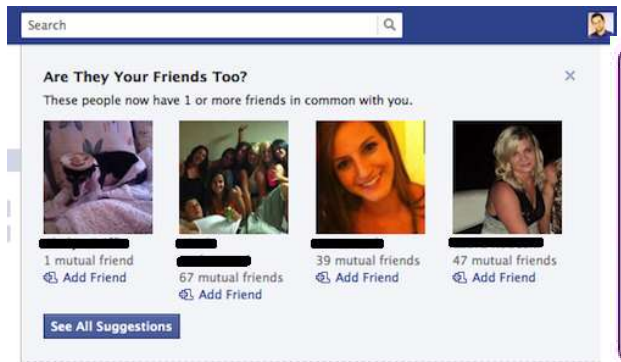
Today's Agenda

- **Introduction to Recommender Systems**
- Fundamental Recommendation Approaches
- Content Recommendation
- Tag Recommendation
- People Recommendation
- Community Recommendation

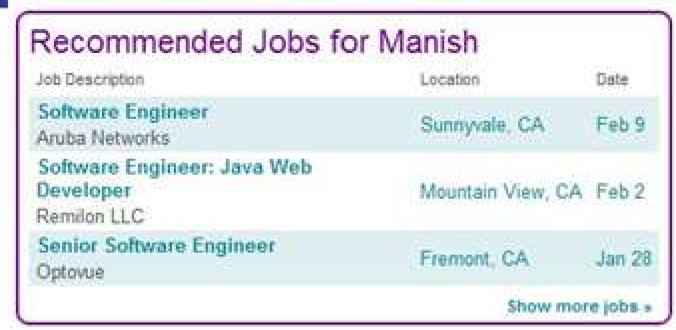
Recommendation Systems Everywhere

LinkedIn People Recommendations

Facebook People Recommendations

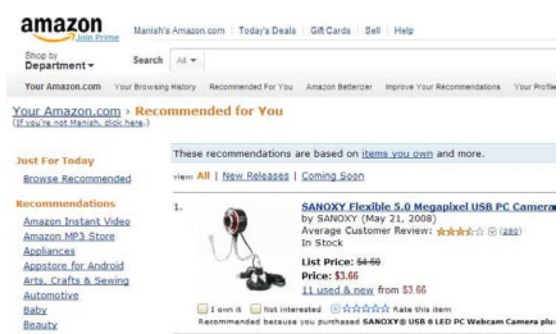


HotJobs Job Recommendations

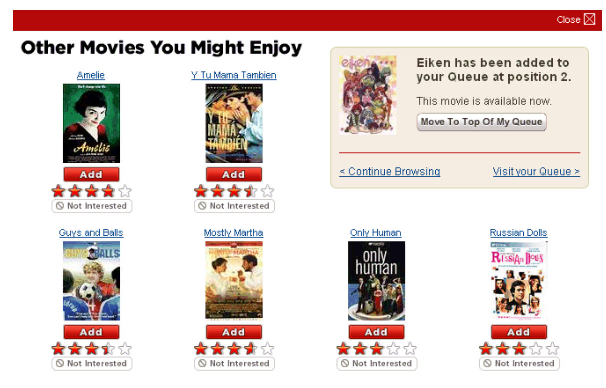


Bing Query Recommendations

Amazon Product Recommendations



Netflix Movie Recommendations



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

Social Overload

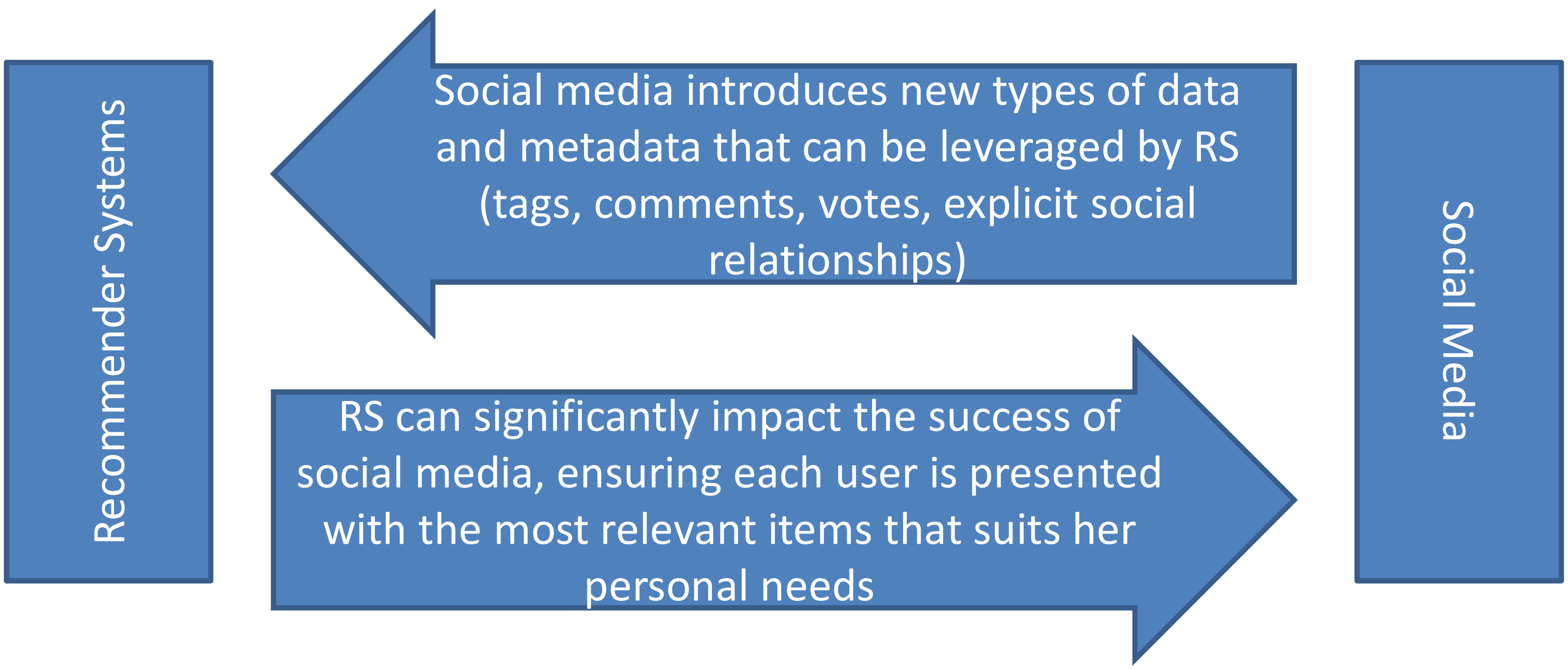
- Information Overload
  - Blogs, microblogs, forums, wikis, news, bookmarked webpages, photos, videos, etc.
- Interaction Overload
  - Friends, followers, followees, commenters, co-members, voters, likers, taggers, review writers, etc.

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

Recommender Systems and 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



Today's Agenda

- Introduction to Recommender Systems
- **Fundamental Recommendation Approaches**
- Content Recommendation
- Tag Recommendation
- People Recommendation
- Community Recommendation

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)
- Knowledge-based recommendation
  - Compute the utility of each item to the user and the user needs
- Hybrid methods
  - Combine several approaches together

Recommendation Techniques

Technique	Background	Input	Process
CF	User-item Matrix	Rating from u to items	Identify similar users, extrapolate from their rating
Demographic	Demographic Information about Users	Demographic Information about u	Identify similar users, extrapolate from their rating
Content-based	Features of Items	Rating from u to items	Generate a classifier based on u's ratings, use it to classify new items
Knowledge-based	Features of Items	User needs	Infer a match between items and u's needs

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

User-based CF Algorithm

- The User x Item Matrix

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

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

User-based CF 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 $\gamma = \frac{1}{\sum_{v=1}^{n} w(u,v)}$

User-based CF Algorithm

- 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

  – $w(u,v) = \frac{\sum_{i \in I}(r_{ui} - \overline{r_u})(r_{vi} - \overline{r_v})}{\sqrt{\sum_{i \in I} (r_{ui} - \overline{r_u})^2}\sqrt{\sum_{i \in I}(r_{vi} - \overline{r_v})^2}}$

CF - 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

Enhancing CF with Friends

- The user's network of friends and people of interest has become more accessible in the Web 2.0 era (Facebook, LinkedIn, Twitter,...)
- Such social relationships can be very effective for recommendation compared to traditional CF
  - Recommendation from people the user knows
  - Spare explicit feedback such as ratings
  - Effective for new users
- Various works have shown the effectiveness of friend-based recommendation over CF, e.g.:
  - Movie and book recommendation - Comparing Recommendations Made by Online Systems and Friends [Sinha & Swearingen, 2001]
  - Friends as trusted recommenders for movies [Golbeck, 2006]
  - Club recommendation within a German SNS - Collaborative Filtering vs. Social Filtering [Groh & Ehmig, Group 2007]

Item-Based Nearest Neighbor Algorithms

- 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
Jon	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

Dimensionality Reduction Algorithms

- Reduce domain complexity by mapping the item space to a smaller number of underlying "dimensions"
  - Represent the latent topics present in those items
  - Improve accuracy in predicting ratings in most cases
  - Reduce run-time performance needs and lead to larger numbers of co-rated dimensions
- Popular techniques: Singular Value Decomposition and Principal Component Analysis
  - Require an extremely expensive offline computation step to generate the latent dimensional space

SVD Decomposition

$$\underset{\mathbf{n\ x\ m}}{\begin{pmatrix} \\ \mathbf{R} \\ \\ \end{pmatrix}} = \underset{\mathbf{n\ x\ r}}{\begin{pmatrix} \\ U \\ \\ \end{pmatrix}} \cdot \underset{\mathbf{r\ x\ r}}{\begin{pmatrix} \Sigma \end{pmatrix}} \cdot \underset{\mathbf{r\ x\ m}}{\begin{pmatrix} \quad V \quad \end{pmatrix}^T}$$

m items**, n** users**, $\mathbf{R_{uj}}$** = Rating of user u for item j

U (V): orthogonal matrix containing the left (right) singular vectors of **R**

Σ: diagonal matrix containing the **singular values** of **R:**

($\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r$)

Trunated SVD: Simply zero Σ after a certain row/column k

$\mathbf{R_k}$ is the best approximation of **R** under Frobenius norm for all rank-k matrices

Recommendations are then given based on R_k

Hybrid Recommendation Methods

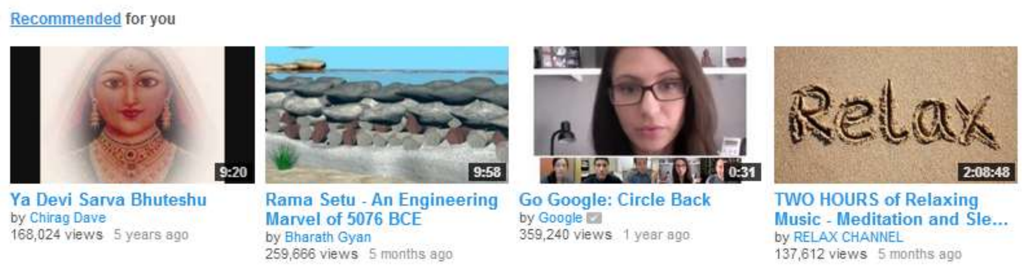
- Any Recommendation approach has pros and cons
  - e.g. CF & CB both suffer from the cold start problem
  - but CF can recommend “outside the box” compared to Content-based approaches
- Hybrid recommender system combines two or more techniques to gain better performance with fewer drawbacks
- Hybrid methods:
  - Weighted: scores of several recommenders are combined together
  - Switching: switch between recommenders according to the current situation
  - Mixed: present recommendations that are coming from several recommenders
  - Cascade: One recommender refines the recommendations given by another

Today's Agenda

- Introduction to Recommender Systems
- Fundamental Recommendation Approaches
- **Content Recommendation**
- Tag Recommendation
- People Recommendation
- Community Recommendation

Content Recommendation: Videos

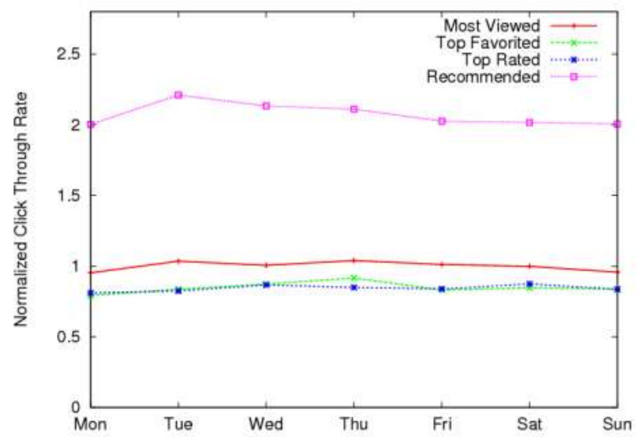
- The YouTube Video Recommendation System [Davidson et al., RecSys '10]



- Goals
  - recent and fresh
  - diverse
  - relevant to the user's recent actions
  - users should understand why a video was recommended to them
- Based on user's personal activities (watched, favorited, liked)
- Using co-visitation graph of videos
- Ranking based on a variety of signals for relevance and diversity

Content Recommendation: Videos

- Calculating related videos (CB)
  - Association rule mining (co-visitation count)
    - Relatedness score of videos $r(v_i, v_j) = \frac{c_{ij}}{f(v_i, v_j)}$
      - c_{ij} is the co-citation count; $f(v_i, v_j)$ is a normalization function which can be set to $c_i \times c_j$
  - Additional issues: presentation bias, noisy watch data
  - More data sources: sequence and timestamp of video watches, video metadata
- Expansion through related video graph
- Ranking
  - Video quality
  - User specificity (personalization): starting from watched or liked videos and traversing the graph
- Topic diversification by removing very similar videos
- Evaluation – A/B testing
  - CTR, long CTR (only count clicks where >X% video was watched), session length, time until first long watch, recommendation coverage (#users with recommended videso)
  - 60% of all homepage video clicks are recommendations

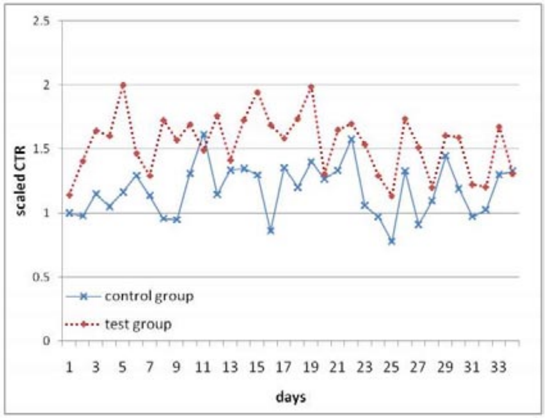


Content Recommendation: News

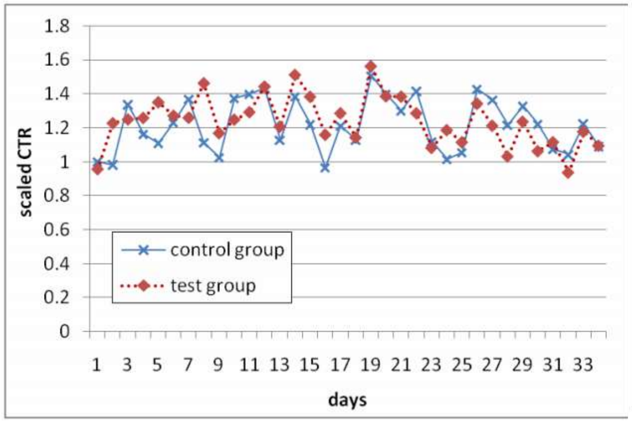
- Personalized news recommendation based on click behavior [Liu et al., IUI 2010]
- News Recommendation on the Google News website
- Combined CB-CF approach: Rec(article) = CR(article) x CF(article)
  - CR=Content-based Recommendation Score
  - CF=Collaborative Filtering based Recommendation Score
- CR is based on the topic of the article and two main factors
  - User's own past clicks (reflecting the user's genuine news interests) for that category
  - General news trends based on click behavior from the general public
- CF is based on clustering dynamic datasets
  - MinHash – fuzzy clustering based on the proportional overlap between the set of items they clicked
  - Shown to scale and retain quality

Content Recommendation: News

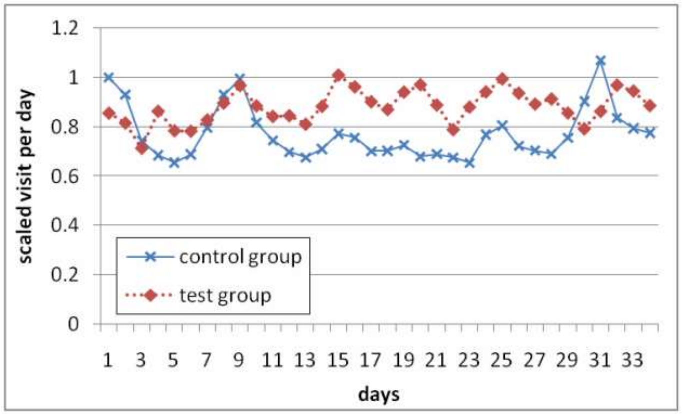
- Evaluation based on live trial
- Hybrid method shown to perform best
  - 31% better than CF method
- Noticeable effect on frequency of visits to Google news website (after a week of getting used to the feature)
- No effect on overall stories read on the News homepage (maybe people only have fixed amount on time)



CTR of the recommended news section



CTR of the Google News homepage



Frequency of website visit per day

Content Recommendation: Digg Stories

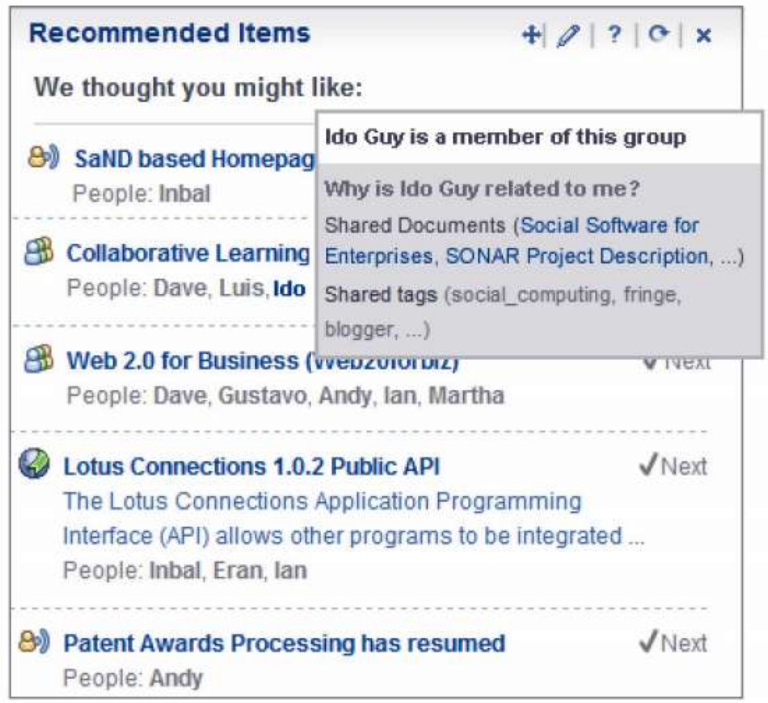
- Social network and social information filtering on Digg [Lerman, ICWSM ’07]
- Digg – social news aggregator, allowing users to submit links to, vote, and discuss news stories
- Recommendations by Digg Friends
- Live trial by tracking Digg users over time
  - Users tend to like stories submitted by friends
  - Users tend to like stories their friends read and liked
- Need to control for user-diversity to avoid “tyranny of the minority”, where lion’s share of front page stories comes from most active users
- In practice, also use the concept of “diggers like me”

Content Recommendation: Blogs

- Document Representation and query expansion models for blog recommendation [Arguello et al., ICWSM ’08]
- Personalized recommendation of blogs in response to a query
- Blog is a collection of documents (blog entries)
- The query represents an interest in a topic
- Two document models
  - Large document model – based on the blog as a whole, a virtual concatenation of its respective entries
  - Smoothed small document model – each entry is a document, aggregation at the ranking level
- Query expansion using Wikipedia
- Evaluation using the TrecBlog06 collection
  - Two document models equally perform, hybridization further improves
  - Query expansion shown to improve recommendation results

Content Recommendation: Social Software Items

- Personalized Recommendation of Social Software Items based on Social Relations [Guy et al., RecSys ’09]
- Social network-based recommendations of blogs, bookmarks, and communities
- Key distinction
  - Familiarity: co-authorship, org chart, direct connection or tagging, etc.
  - Similarity – co-usage of tags, co-bookmarking, co-membership, co-commenting
- Explanations – showing the “implicit recommender” and her relationship to the user and item

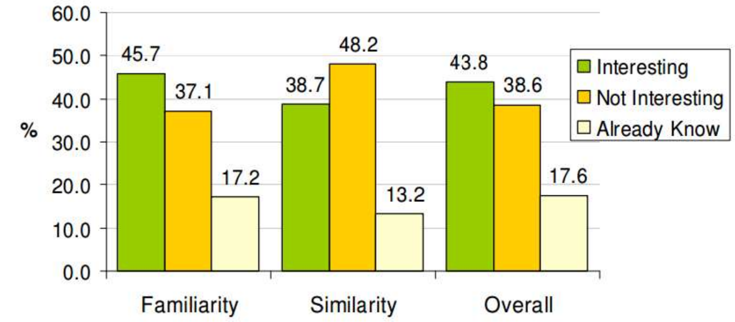


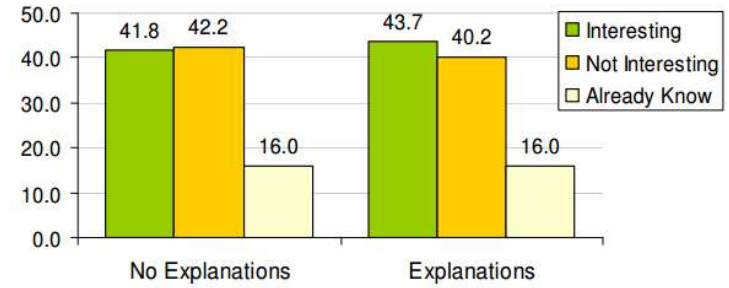
Content Recommendation: Social Software Items

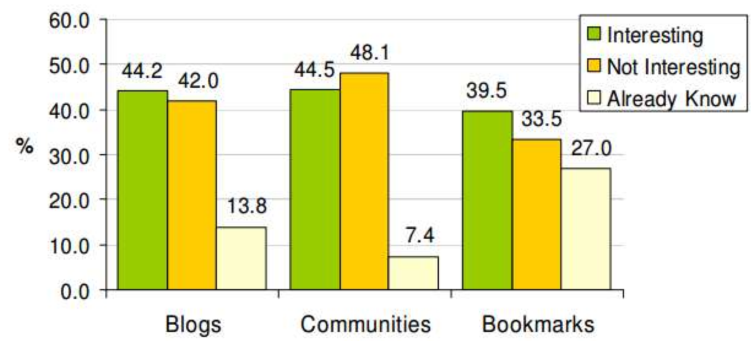
- $RecScore(u,i) = e^{-\alpha t(i)} \sum_{v \in N^T(u)} S^T[u,v] \sum_{r \in R(v,i)} W(r)$
- $t(i)$ is number of days passed since the creation date of item i
- α is a decay factor
- $N^T(u)$ is set of users within u's network of type T
- $T \in \{familiarity, similarity, overall\}$
- $S^T[u,v]$ is relationship score between users u and v based on a network of type T
- $R(v,i)$ is set of all relationship types between user v and item i (e.g., authorship, membership, etc)
- $W(r)$ is weight for user-item relationship type between user v and item i

Content Recommendation: Social Software Items

- Recommendations from familiar people are significantly more accurate than recommendations from similar people
- Similar people yield more diverse, less expected items
- Explanations have an instant effect increasing interest in recommended items
- Bookmarks have a very high percentage of known items - 27%, while communities have the lowest one – only 7.4%

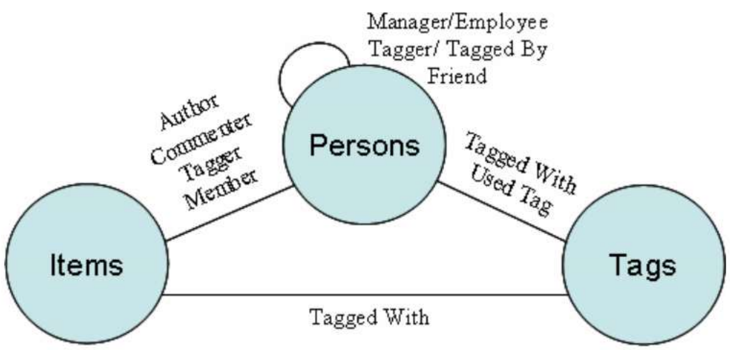


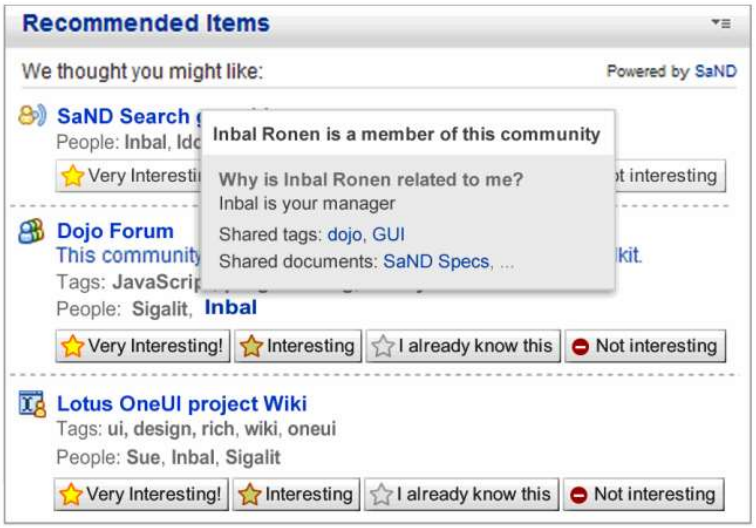




Content Recommendation: Social Software Items

- Social media recommendation based on people and tags [Guy et al., SIGIR ’10]
- 5 item types: blogs, bookmarks, communities, wikis, files
- Comparison between people-based and tag-based recommenders
- 3 types of tags
  - used tags — direct relation based on tags the user has used
  - incoming tags — direct relation based on tags applied on the user by others
  - indirect tags — indirect relation based on tags applied on items related to the user





Content Recommendation: Social Software Items

- User profile P(u) =related people N(u) +related tags T(u)
- $RecScore(u,i) = e^{-\alpha t(i)}\left[\beta \sum_{v \in N(u)} w(u,v).w(v,i) + (1-\beta)\sum_{t \in T(u)} w(u,t).w(t,i)\ \right]$
- $t(i)$ is number of days passed since the creation date of item i
- α is a decay factor
- W denotes the relationship strength weights

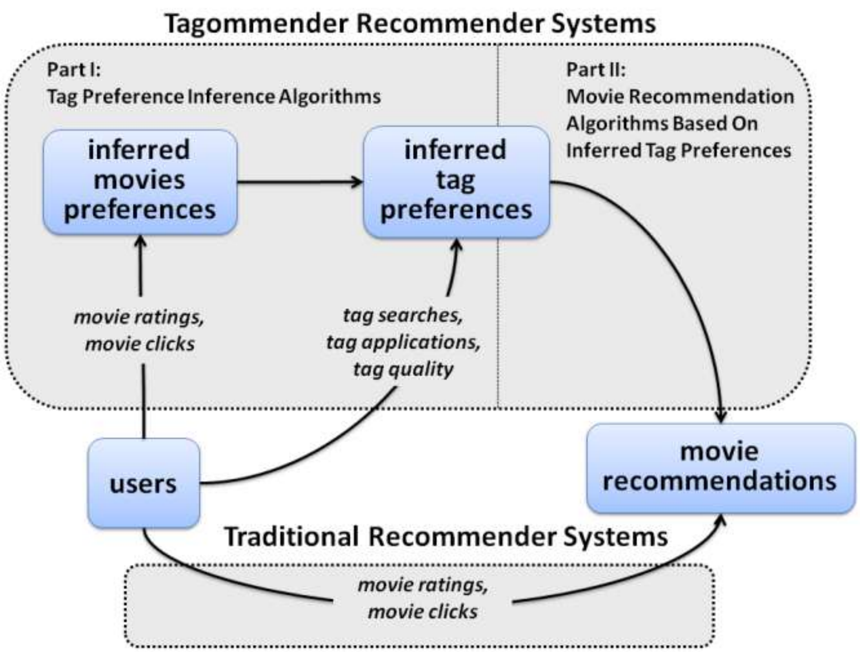
Content Recommendation: Social Software Items

%	Not Interested	Interested	Highly Interested
used	16.84	38.25	44.91
incoming	15.48	31.75	52.78
direct	**7.46**	22.81	**69.74**
indirect	35.38	45.38	19.23

- Hybrid tags (used+incoming=direct) most accurate, Indirect are least effective
- Tag-based method significantly outperforms people-based recommendation method in terms of accuracy
- Yet has less diversity, more expected results, and less effective explanations
- Hybrid combines the good of both worlds

Content Recommendation: Movies

- Tagommenders: connecting users to items through tags [Sen et al., WWW '09]
- Inspecting various ways to recommend items based on tags
  - Movie ratings
  - Movie clicks
  - Tag applications
  - Tag Searches
  - Tag Quality – based on #users who apply this tag/search using this tag
- Evaluation based on MovieLens
  - Tag preference inference
    - Algorithms based on movie ratings performed best followed by those based on tag signals
    - Algorithm based on combination of all signals, performed best.
  - Tag-based algorithms outperformed state-of-the-art CF for movie recommendations



Summary of Key Points

- Social networks play an important role in CF for social media
  - Enhance regular CF in various manners
- "Tagommenders" are highly effective for recommendation
  - Outperform regular CF
- As in Traditional RS, hybrid approaches (e.g., tags+social networks, short+long term interests) typically further improve
- Many users => strong evaluation on live systems
- Accuracy vs. Serendipity tradeoff
  - Accuracy alone is not enough – serendipity and diversity also play a key role [Mcnee et al., CHI '06]

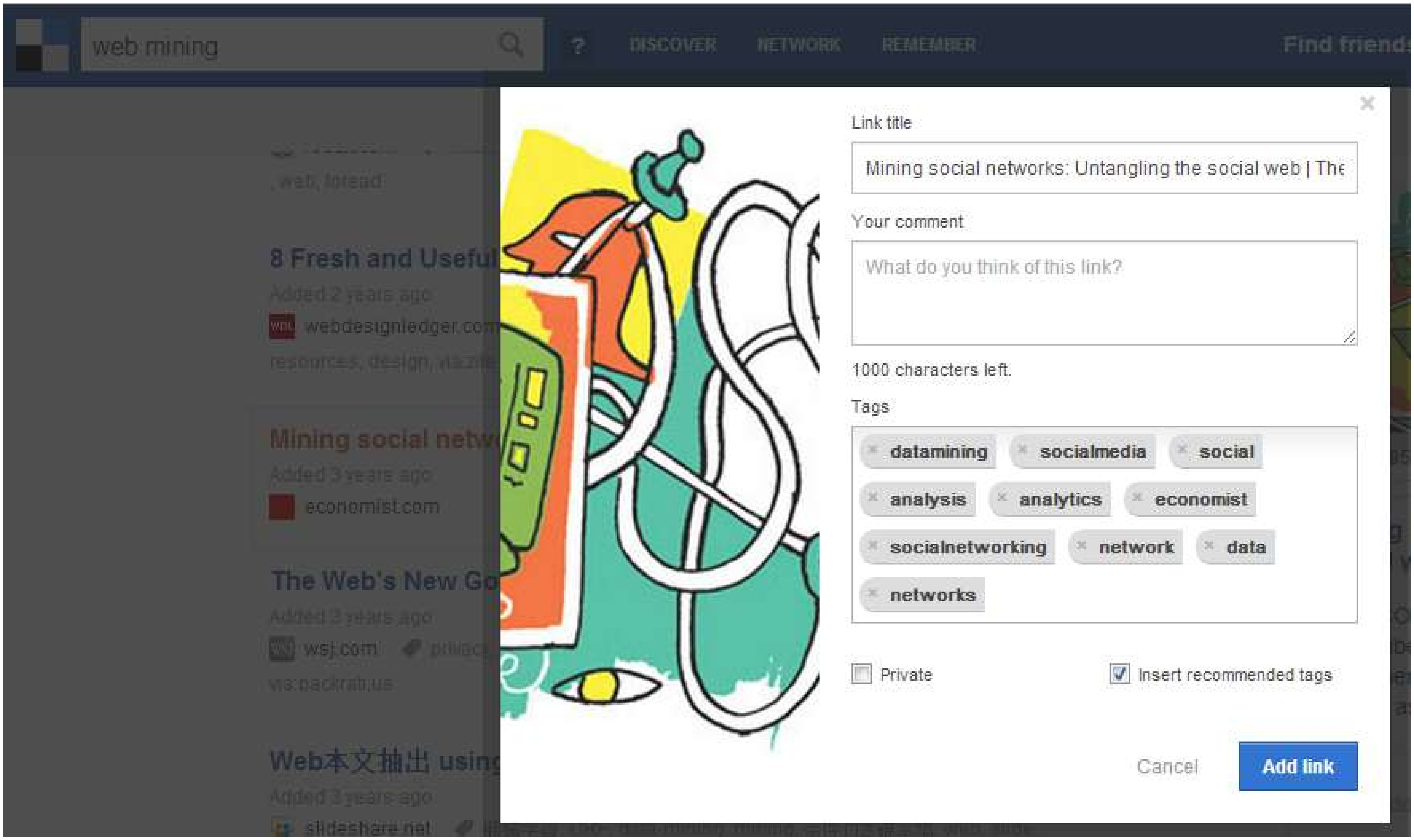
Today's Agenda

- Introduction to Recommender Systems
- Fundamental Recommendation Approaches
- Content Recommendation
- **Tag Recommendation**
- People Recommendation
- Community Recommendation

Tag Recommendation

- Adding terms (tags) to objects by the public provides additional contextual and semantic information to various resources
  - Web pages (e.g. Delicious)
  - Academic publications (e.g. CiteULike)
  - Multimedia objects (e.g. Flickr, Last.Fm, YouTube)
- External tags are useful for many applications
  - Search/browse, classification, tag-cloud representation, query expansion
- Tag Recommendation: recommend appropriate tags to be applied by the user per specific item annotation
  - Assist the user in the tagging phase
  - Reduce undesired noise in the aggregated folksonomy

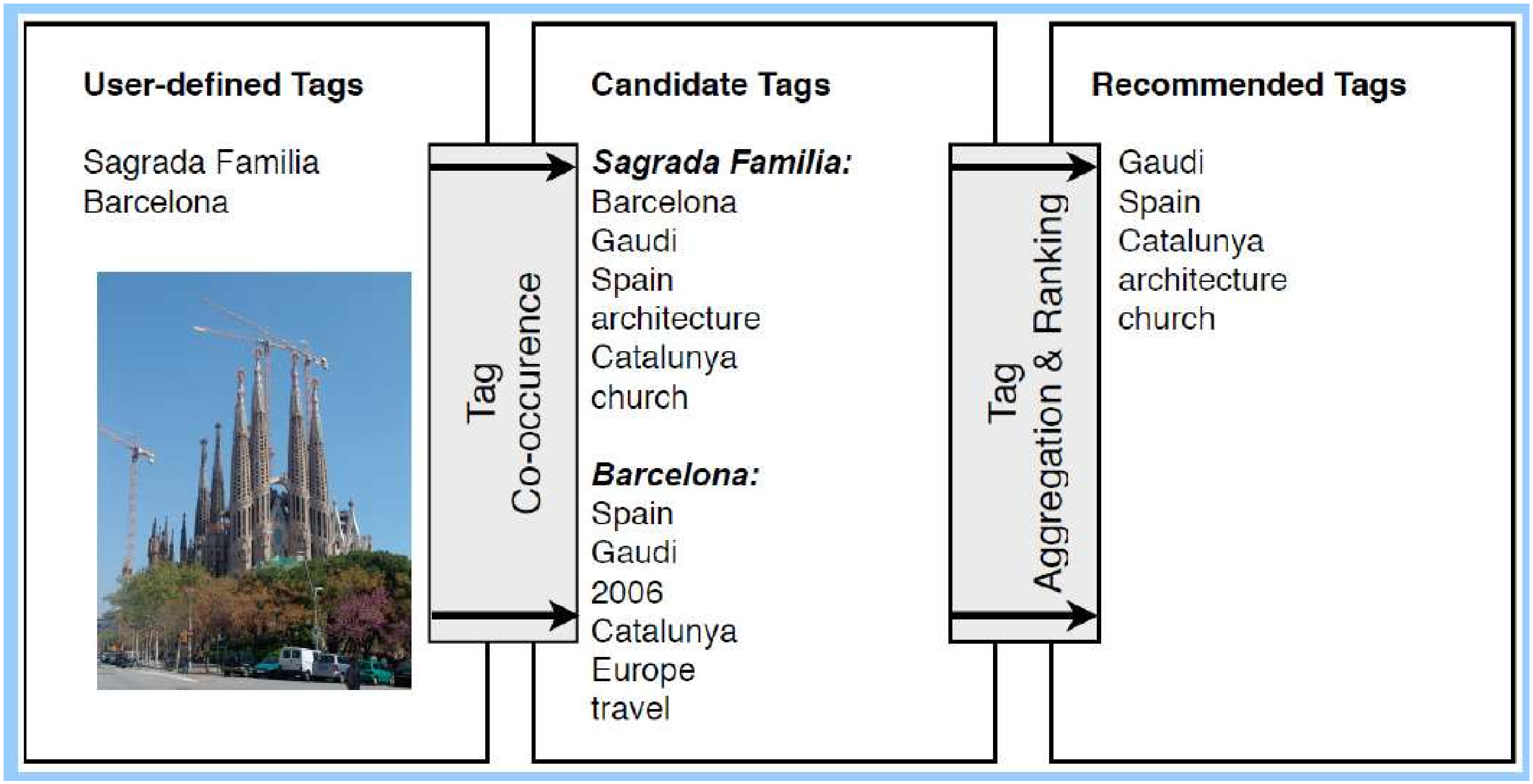
Delicious Tag Recommendation Example



Tag Recommendation Approaches

- Popular (Recommend the most popular tags to the user)
  - Popular tags already assigned for the target item (Golder 2005)
  - Frequent tags previously used by the user
  - Tags co-occurred with already assigned tags (Sigurbjornsson 2008)
- Collaborative Filtering
  - Recommend tags associated with “similar” items
  - Recommend tags given by “similar” users
- Hybrid
  - Recommend tags given by similar users to similar items (Symeonidis 08, Rendle 10, Carmel 10)

Flickr's Tag Recommender (Sigurbjornsson WWW2008)



Content-based Tag Recommendation

- Recommend keywords/phrases from the item's associated text (content, anchor-text, meta data, etc.)
  - e.g. terms with highest tf-idf score
- Analyze mutual relationship between content and tags
  - Recommend tags that have the highest co-occurrence with important keywords
  - Language modeling approach (Givon 2010)
    - Estimate the joint tag and keyword probability distribution.
    - This provides an estimate that a given item will be annotated with certain tags, given a background collection of annotated items

Graph-based Tag Recommendation

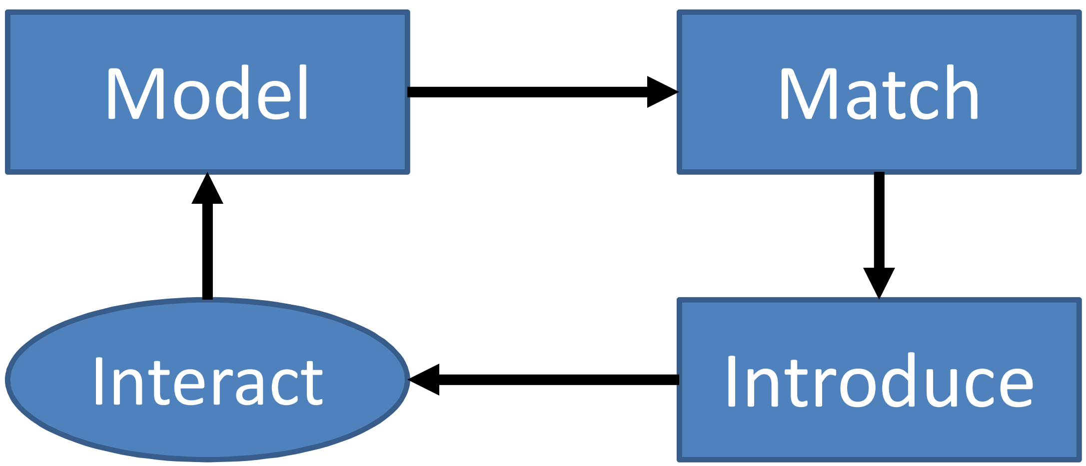
- The FolkRank algorithm (Hotho 2006)
  - A resource which is tagged with important tags by important users becomes important
  - The same holds, symmetrically, for tags and users
- We have a graph of connected vertices (resources, users, tags) which are mutually reinforcing each other by spreading their weights
- Graph nodes are scored by random walk techniques
- $w = dAw + (1 - d)p$
  - w: a weight vector over nodes
  - A: a row-stochastic matrix of the graph
  - p: preference vector over the nodes
- For tag recommendation, return the top ranked tags, while setting p to bias the desired pair of user and resource
- Evaluation
  - For each user we pick one of his posts randomly
  - The task of the different recommenders is to predict the user tags of this post, based on the rest of the Folksonomy
  - We measure how many of those tags are covered by the top-k recommended tags
  - FolkRank works better than any of the following: CF, popular tags, popular tags by resource

Today’s Agenda

- Introduction to Recommender Systems
- Fundamental Recommendation Approaches
- Content Recommendation
- Tag Recommendation
- **People Recommendation**
- Community Recommendation

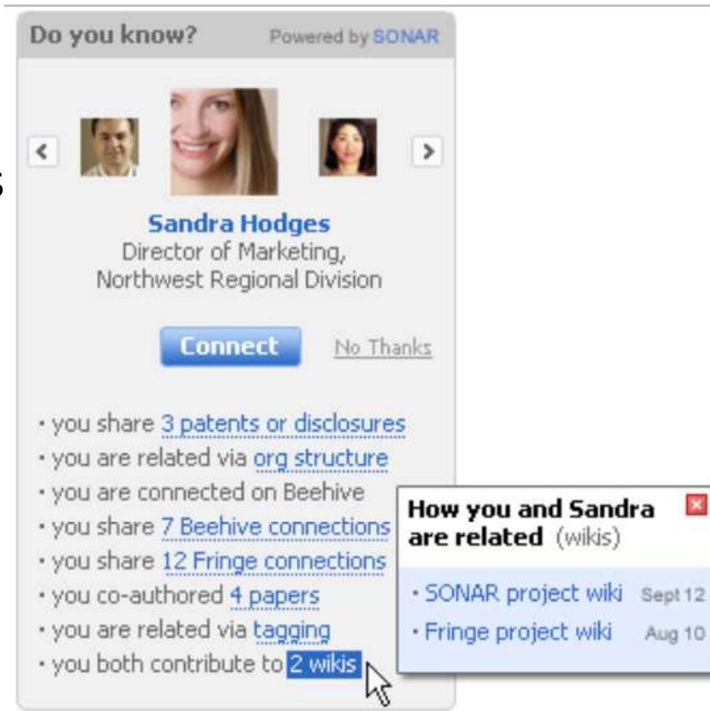
People Recommendation: Social Matching

- Social matching systems = recommender systems that recommend people to each other
  - Must reveal some amount of personal information
  - Privacy, trust, reputation, interpersonal attraction have greater importance
  - Interaction overload vs. information overload



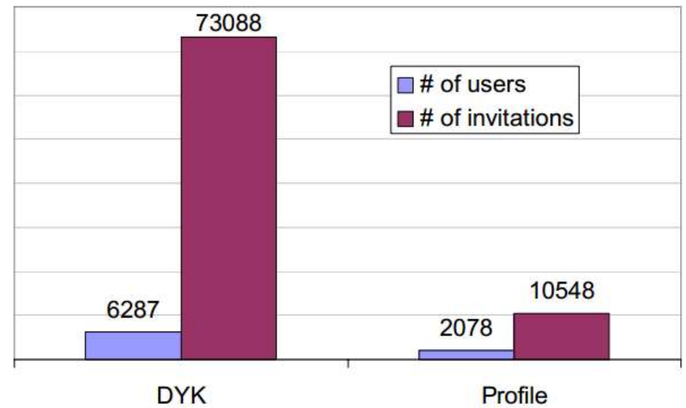
People Recommendation: Recommending People to Connect with

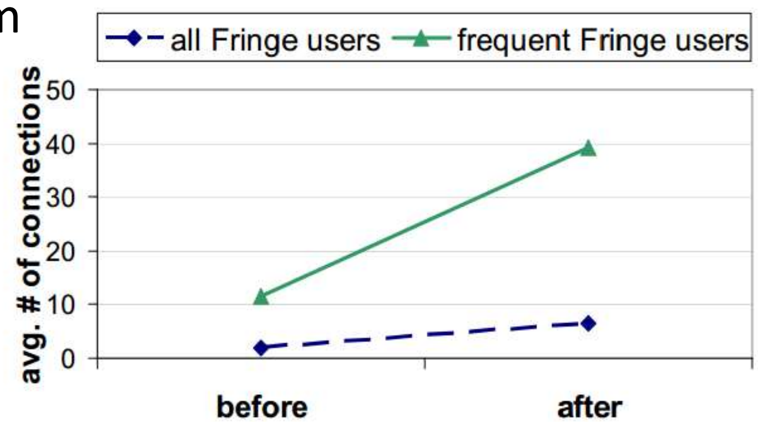
- Do You Know? Recommending People to Invite into Your Social Network [Guy et al., IUI '09]
- Recommendation in the enterprise based on the following signals
  - Org chart relationships
  - Paper and patent co-authorship
  - Project co-membership
  - Commenting on each others' blogs
  - Tagging each other
  - Mutual connections
  - Connection in another SNS
  - Wiki co-editing
  - File sharing
- Rich and detailed "evidence"



People Recommendation: Recommending People to Connect with

- Evaluation of DYK (Did You Know) feature based on the Fringe enterprise SNS
- Dramatic increase in the number of invitations sent and users accepting invitations
  - "I must say I am a lazy social networker, but Fringe was the first application motivating me to go ahead and send out some invitations to others to connect"
- Evidence increases users' trust in the system and makes them feel more comfortable
  - "If I see more direct connections I'm more likely to add them […] I know they are not recommended by accident"
- Substantial increase in friends per user
- Sharp decay in usage over time
  - Excitement drops, connections exhausted



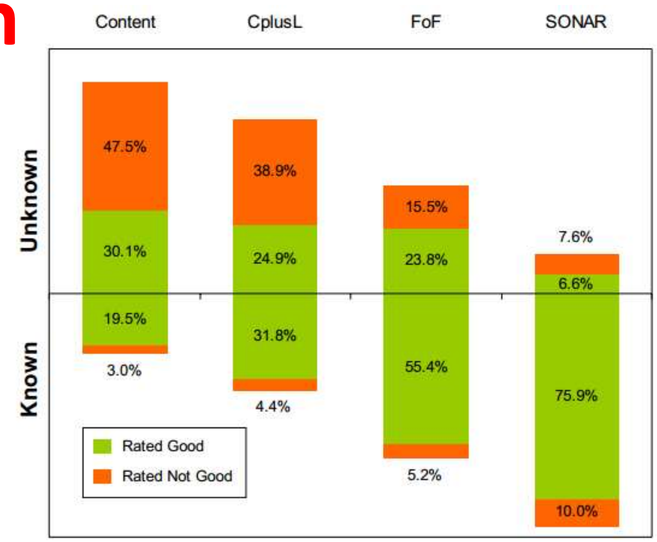


People Recommendation: Recommending People to Connect with

- Make new friends, but keep the old: recommending people on social networking sites [Chen et al., CHI '09]
- Content Matching (CM)
  - Profile entries, status messages, photo text, shared lists, job title, location, description, tags
  - Strength of user u's interest in word w_i is $v_u(w_i) = TF_u(w_i) . IDF_u(w_i)$
    - $IDF_u(w_i) = \log[(\#all\ users)/(\#users\ using\ w_i\ at\ least\ once)]$
  - Cosine similarity of both users' word vector
  - Latent semantic analysis did not perform better
    - And does not yield intuitive explanations
- Content-plus-Link (CplusL)
  - Hybrid CM + social link
  - Social link: a sequence of 3 or 4 users
    - a connects to b, a comments on b, b connects to a
- Friend-of-Friend (FoF)
  - Based on number of mutual friends
  - One or more recommendations for 57.2% of the users
- Aggregated Relationships (SONAR)
  - Similar to the "Do you know?" algorithm
  - One or more recommendations for 87.7% of the users

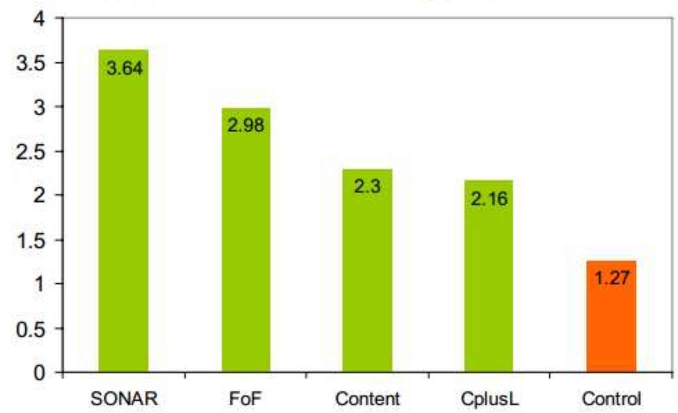
People Recommendation: Recommending People to Connect with

- Evaluation based on the SocialBlue Enterprise SNS (“Beehive”)
- Survey with 258 participants
  - CM and CplusL yield mostly unknown people, while FoF and SONAR yield mostly known
  - Content similarity vs. relationships algorithms
    - The latter are more accurate overall
    - The former are better at discovering new friends
- Controlled field study with 3,000 users
  - SONAR yields most effective results
  - Combine relationships (at first) and content similarity (when the network grows)?



SONAR	FoF	CplusL	Content
59.7%	47.7%	40.0%	30.5%

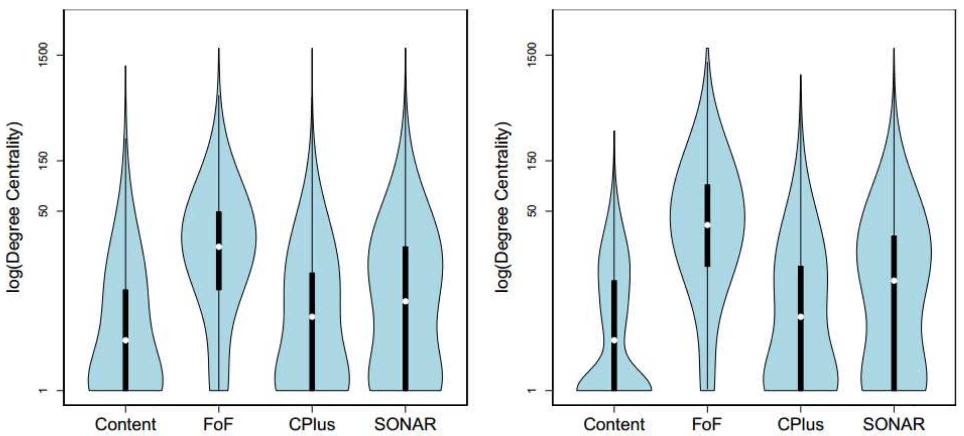
Recommendations resulting in connect actions.



Increase in number of friends.

People Recommendation: Recommending People to Connect with

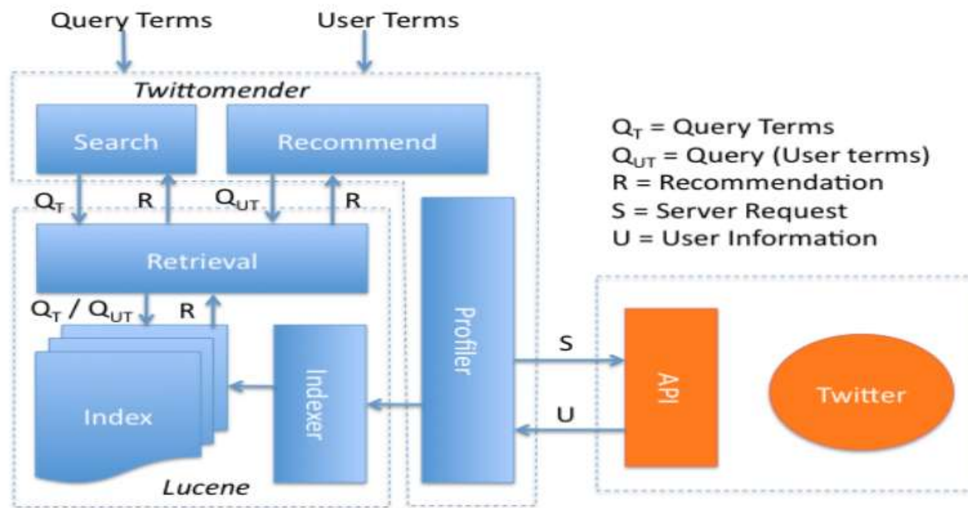
- The network effects of recommending social connections [Daly et al., RecSys '10]
  - IBM's SocialBlue social network site
- FoF is highly biased towards well-connected users, leading to high rec. frequency of the same users
- CM is most diverse and often recommends users with few connections only
- CM and SONAR affect betweenness centrality most significantly
- CM is most biased for same country but least biased for same division
- SONAR substantially increases cross-country and intra-division connections



(a) All (b) Accepted

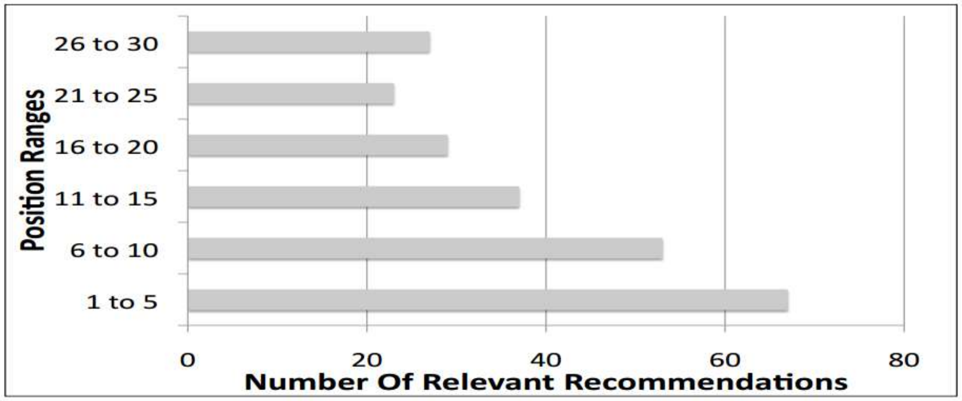
People Recommendation: Recommending People to Follow

- Recommending twitter users to follow using content and collaborative filtering approaches [Hannon et el., RecSys'10]
- CB, CF, and Hybrid approaches
- User profiles based on
  - Own tweets
  - Followers' tweets
  - Followees' tweets
  - Followers
  - Followees
- Using Lucene to index users by their profile, after applying TF-IDF to boost distinctive terms/users within the profile



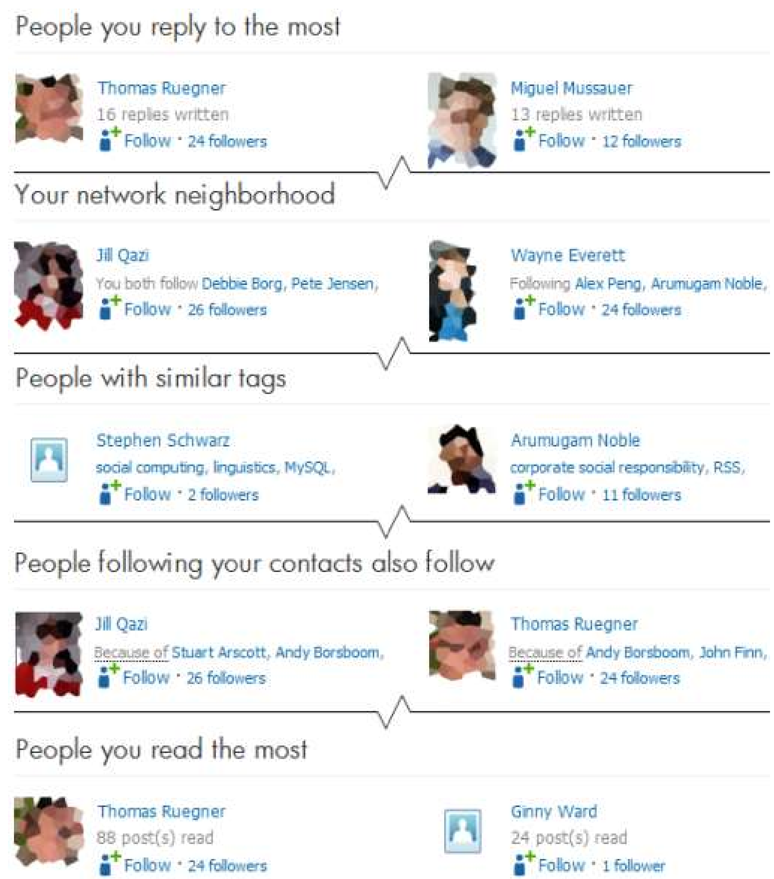
People Recommendation: Recommending People to Follow

- Offline Evaluation, 20K users
  - 19,000 training set Twitter users
  - 1,000 test users
  - Create index per profile and predict followees
  - Measure by precision and position
  - Slight advantage to followers and tweets of followers
  - Hybrid (own, followers' and followees' tweets) improves results (precision close to 0.3)
- Live User Trial, 34 participants
  - Hybrid approach combining all types
  - 30 recommended Twitter users
  - On average, 6.9 out of 30



People Recommendation: Recommending People to Follow

- Who should I follow? Recommending people in directed social networks [Brzozowski & Romero, 2010]
- Experiments with the WaterCooler enterprise SNS
- 110 users followed 774 new people during 24-day trial period
- Strongest pattern A<-X->B
  - Sharing an audience with someone is a surprisingly compelling reason to follow them
  - Besides it is not easy for A and B to find each other
- Similarity (and most read) are not so strong indicators
- Most replied is a good indicator



Today's Agenda

- Introduction to Recommender Systems
- Fundamental Recommendation Approaches
- Content Recommendation
- Tag Recommendation
- People Recommendation
- **Community Recommendation**

Community Recommendation: Recommending Similar Communities

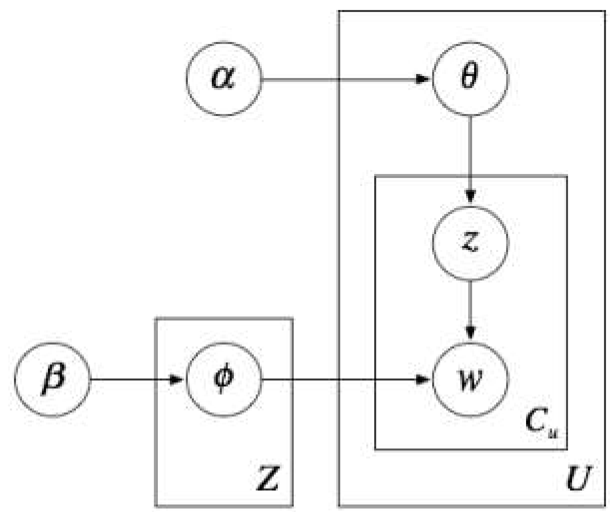
- Evaluating similarity measures: a large-scale study in the Orkut social network [Spertus et al., KDD '05]
- Orkut – SNS by Google, used to be largest in Brazil and India
  - 20K communities with over 20 members
  - 180K distinct members
  - Over 2M memberships
- 6 community similarity measures, based on community membership
  - How appropriate is R (recommended community) as a recommendation for B (base community)
  - L1 norm: $L1(B,R) = \frac{|B \cap R|}{|B||R|}$
  - L2 norm: $L2(B,R) = \frac{|B \cap R|}{\sqrt{|B|.|R|}}$
  - Log-Odds: $LogOdds(B,R) = \log \frac{P(R|B)}{P(\bar{R}|B)}$
  - Salton (IDF): $IDF(B,R) = \frac{|B \cap R|}{|B|}.(-\log \frac{|R|}{|U|})$
  - Pointwise Mutual-Info: Pos. Correlations $MI(b,r) = P(R,B).\log \frac{P(R,B)}{P(R).P(B)}$
  - Pointwise Mutual-Info: Pos. and Neg. Correlations $MI2(b,r) = P(R,B).\log \frac{P(R,B)}{P(R).P(B)} + P(\bar{R},\bar{B}).\log \frac{P(\bar{R},\bar{B})}{P(\bar{R}).P(\bar{B})}$

Community Recommendation: Recommending Similar Communities

- Live trial on the Orkut SNS
- Click-through to measure user acceptance
- L2 shown as the best similarity measure
- Followed by MI1, MI2, IDF,L1, and Log-Odds
- Conversion rate - % of non-members who clicked-through and then joined
  – 46% for base members, 17% for base nonmembers

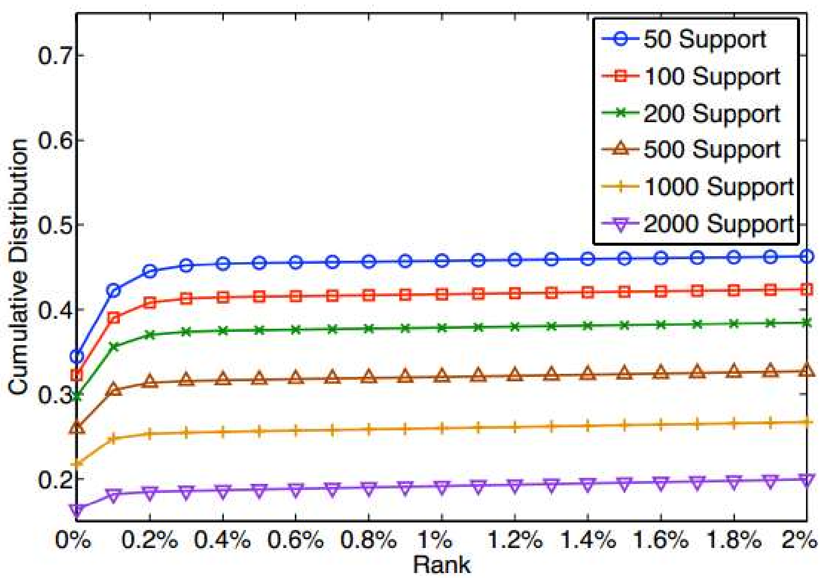
Community Recommendation: Personalized Community Recommendation

- Collaborative filtering for Orkut communities: discovery of user latent behavior [Chen et al., WWW '09]
- Personalized community recommendation using CF of two types
  - Association rule mining (ARM) – association between communities shared between many users: users who join X typically join Y
  - Latent Dirichlet Allocation (LDA) – user-community co-occurrences using latent aspects (topics): x is related to y through a semantic feature, e.g., "baseball"
    - Users=docs, communities=words, membership=co-occurrence
    - Per-topic distribution of users and communities

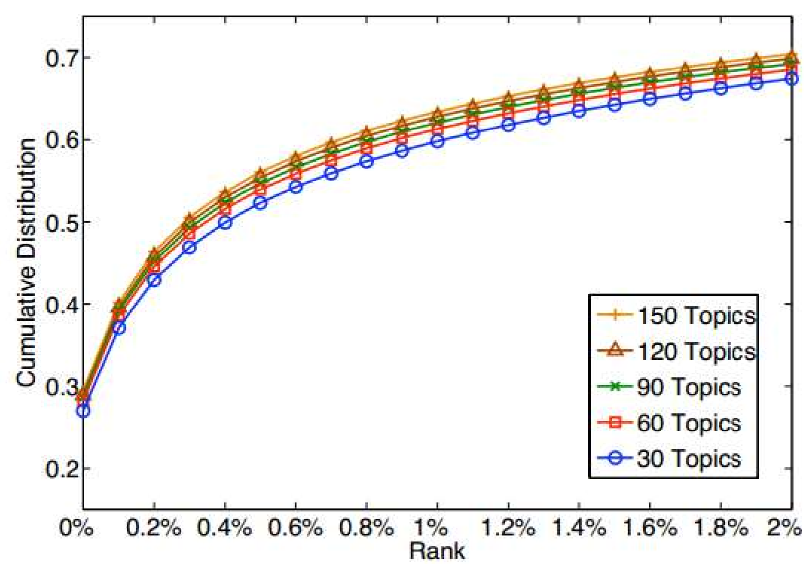


Community Recommendation: Personalized Community Recommendation

- Orkut membership data: 492K users, 118K communities
- Top-k recommendation: withhold 1 community the user has joined with k-1 random communities, obtain rank (k=1001)
- ARM is better when recommending lists of up to 3 communities
- LDA is consistently better when recommending a list of 4 or more
- In general, LDA ranks communities better than ARM
- LDA is parallelized to improve efficiency



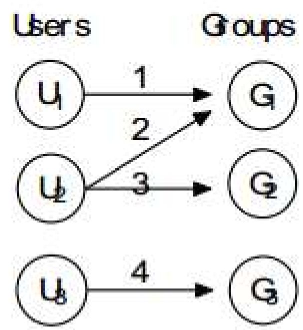
(a) ARM: micro view of top-k performance

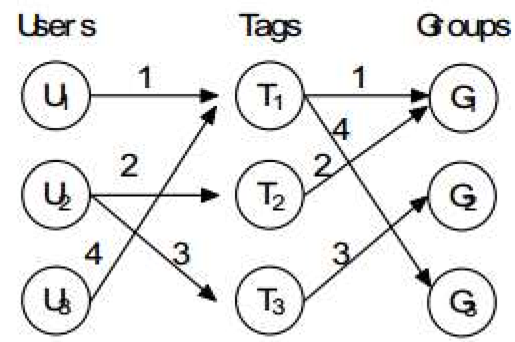


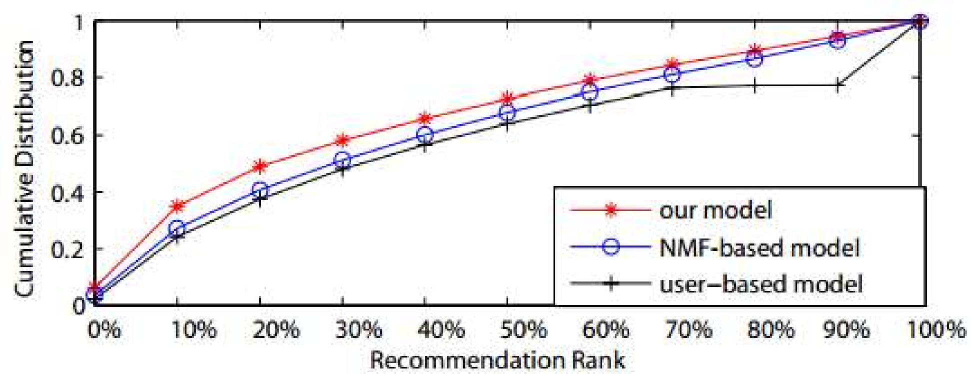
(b) LDA: micro view of top-k performance

Community Recommendation: Personalized Community Recommendation

- Flickr group recommendation based on tensor decomposition [Zheng et al., SIGIR ’10]
- Latent association between users and groups through tags
- Model through a three-mode tensor
- Experiments using 197x5328x4064 tensor
  - Statistically significant superiority over user-based and non-negative matrix factorization approaches

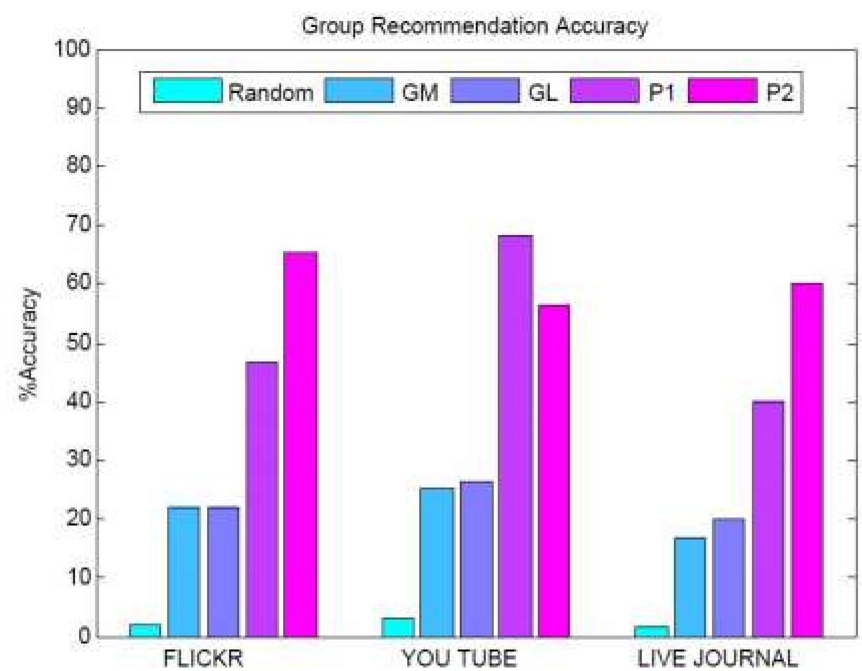






Community Recommendation: Personalized Community Recommendation

- Group Proximity Measure for Recommending Groups in Online Social Networks (Saha & Getoor, SNA-KDD '08)
- Group proximity based on escape probability: prob that random walk starting from node i will visit node j before returning to node i.
  - Identify core and outlier nodes; shrink graph to community-community graph
  - Random walks starting in G1 visits G2 before any other node in G1
- Method 1: recommend the closest groups to those the user is a member of
- Method 2: boost groups that are closer to many of the user's groups
- GM and GL-two SVM classifiers trained over group membership data (binary/frequency data of group membership wrt user)
- Accuracy = predicted membership in top 5



Community Recommendation: Personalized Community Recommendation

- Group Recommendation System for Facebook [Baatarjav et al., OTM ’08]
  - Matching user profiles with group identities
  - Combining hierarchical clustering and decision tree
  - Facebook data for University of North Texax (1580 users), focus on 17 groups
  - 15 profile features: age, gender, timezone, relationship status, political view, interests, movies, affiliations, …
  - Characterize groups by the majority of their members
  - Removing noise by removing members far from the group’s center
  - Reported average accuracy 73%
- From LinkedIn Blog (“groups you may like”)
  - Building a virtual profile per group by selecting the most representative features of group members using Information Theory techniques like Mutual Information and KL Divergence.
  - Mapping user’s attributes to group’s virtual profile
  - Adding more recommendations based on CF

Take-away Messages

- Social Recommender Systems are aimed at solving the information overload and the interaction overload problems
- Collaborative filtering based methods and content based methods are the two most popular approaches for social recommendation systems
- We looked at methods to recommend content to users: Videos, news, blogs, Digg stories, social software items
- We also studied methods for tag, people and community recommendations

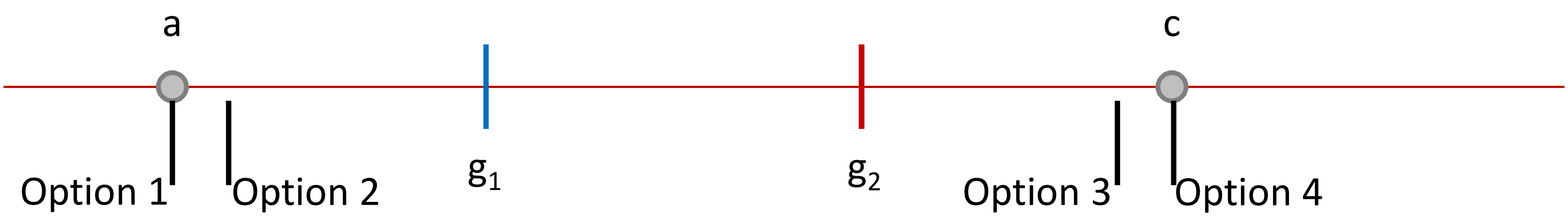
Further Reading

- J. Bobadilla, F. Ortega, A. Hernando, A. Gutiérrez, Recommender systems survey, Knowledge-Based Systems, Volume 46, July 2013, Pages 109-132, ISSN 0950-7051, http://dx.doi.org/10.1016/j.knosys.2013.03.012.
- Stefan Siersdorfer, Sergej Sizov. Social Recommender Systems for Web 2.0 Folksonomies. Pages 261-269. HT 2009. http://www.l3s.de/~siersdorfer/sources/2009/p261-siersdorfer.pdf
- Su Mon Kywe, Ee-Peng Lim, Feida Zhu. A Survey of Recommender Systems in Twitter. Social Informatics Lecture Notes in Computer Science Volume 7710, 2012, pp 420-433. http://www.mysmu.edu/faculty/fdzhu/paper/SocInfo'12_57.pdf
- Jie Bao, Yu Zheng, David Wilkie, and Mohamed F. Mokbel. A Survey on Recommendations in Location-based Social Networks. ACM Transaction on Intelligent Systems and Technology. 2013. http://research.microsoft.com/apps/pubs/?id=191797

Preview of Lecture 8: Social Recommender Systems (Part 2)

- Recommendation for Groups
- The Cold Start Problem
- Trust
- Social Recommender Systems in the Enterprise
- Temporal Aspects in Social Recommendation
- Social Recommendation over Activity Streams
- Evaluation Methods
- Summary of Social Recommender Systems

Surprise Quiz 2 (5 Marks)

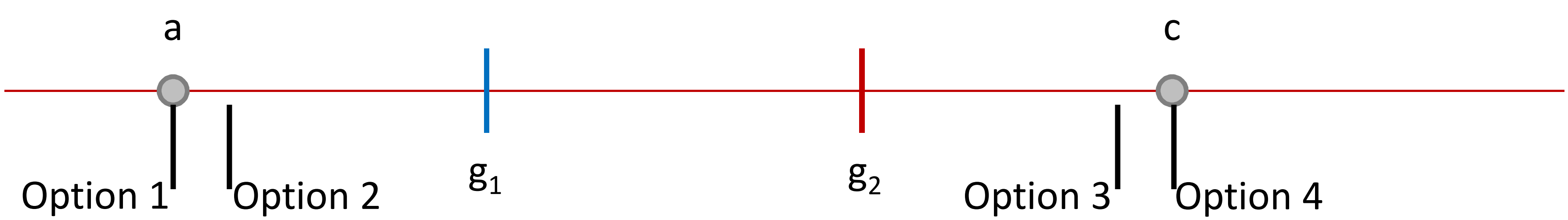


a and c are two data points, g1 and g2 are initial cluster centroids.
Answer in 1-2 sentences. Each question carries 1 point. Only 0.5 point per question if "why" is not answered.

Q-1. At the end of K-Means, where will cluster center g1 end up – Option 1 or Option 2? Why?
Q-2. At the end of EM, where will cluster center g1 end up – Option 1 or Option 2? Why?

Q-3. Is EM for Gaussian Mixture Models supervised or unsupervised? Why?
Q-4. Is EM for Gaussian Mixture Models an online algorithm or a batch algorithm? Why?
Q-5. Is EM for Gaussian Mixture Models closed-form or iterative? Why?

Answers for Surprise Quiz 2



At the end of K-Means, where will cluster center g1 end up – Option 1 or Option 2?

Option 1: K-Means puts the "mean" at the center of all points in the cluster, and point a will be the only point in g1's cluster.

At the end of EM, where will cluster center g1 end up – Option 1 or Option 2?

Option 2: EM puts the "mean" at the center of all points in the dataset, where each point is weighted by how likely it is according to the Gaussian. Point a and Point b will both have some likelihood, but Point a's likelihood will be much higher. So the "mean" for g1 will be very close to Point a, but not all the way at Point a.

Answers for Surprise Quiz 2

Is EM for GMMs

Supervised or Unsupervised?

- Unsupervised

Online or batch?

- batch: if you add a new data point, you need to revisit all the training data to recompute the locally-optimal model

Closed-form or iterative?

-iterative: training requires many passes through the data

Disclaimers

- This course represents opinions of the instructor only. It does not reflect views of Microsoft or any other entity (except of authors from whom the slides have been borrowed).
- Algorithms, techniques, features, etc mentioned here might or might not be in use by Microsoft or any other company.
- Lot of material covered in this course is borrowed from slides across many universities and conference tutorials. These are gratefully acknowledged.

Thanks!

References: Fundamental Recommendation Approaches

- Recommender Systems: An Introduction, Jannach et al. 2011.
- Recommender Systems Handbook, Ricci et al. 2010
- Hybrid web recommender systems, : Survey and Experiments. Burke, User Modeling and User-Adapted Interaction. 2002
- Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. Adomavicius et al., IEEE Transactions on Knowledge and Data Engineering. 2005

References: Content Recommendation

- Arguello, J., Elsas, J., Callan, J., & Carbonell J. Document representation and query expansion models for blog recommendation. Proc. ICWSM'08.
- Davidson, J. Liebal, B., Junning L., et al. The YouTube video recommendation system. Proc. RecSys '10, 293-296.
- Groh, G., & Ehmig, C. Recommendations in taste related domains: collaborative filtering vs. social filtering. Proc. GROUP '07, 127-136.
- Golbeck J. Generating predictive movie recommendations from trust in social networks. Proc. 4th Int. Conf. on Trust Management. Pisa, Italy
- Guy, I., Zwerdling, N., Carmel, D., et al. Personalized recommendation of social software items based on social relations. Proc. RecSys '09, 53-60.
- Guy, I., Zwerdling N., Ronen, I, et al. Social media recommendation based on people and tags. Proc SIGIR '10, 194-201.

References: Content Recommendation

- Lerman, K. Social networks and social information filtering on Digg. Proc. ICWSM ’07.
- Liu, J., Dolan, P. & Pederson E.B. Personalized news recommendation based on click behavior. Proc. IUI ’10, 31-40.
- McNee M.S., Riedl, J., & Konstan J.A. 2006. Being accurate is not enough: how accuracy metrics have hurt recommender systems. Proc CHI '06, 1097-1101.
- Sen, S., Vig, J., & Riedl, J.Tagommenders: connecting users to items through tags. Proc. WWW '09, 671-680.
- Sinha, R. & Swearingen, K. Comparing recommendations made by online systems and friends. 2001 DELOS-NSF Workshop on Personalization and Recommender Systems in Digital Libraries.

References: Tag Recommendation

- The Structure of Collaborative Tagging System. Golder et al. Journal of Information Science, 2005
- Flickr tag recommendation based on collective knowledge. Sigurbjörnsson et al. WWW 2008
- Tag recommendations based on tensor dimensionality reduction. Symeonidis. RecSys 2008
- Pairwise interaction tensor factorization for personalized tag recommendation Rendle et al. WSDM 2010
- Social bookmark weighting for search and recommendation, Carmel et al. VLDB Journal 2010
- Large Scale Book Annotation with Social Tags. Givon et al. ICWSM 2009
- FolkRank: A Ranking Algorithm for Folksonomies. Hotho et al. FGIR 2006
- Tag Recommendations in Folksonomies, Jaeschke et al, PKDD 2007

References: People Recommendation

- Brzozowski, M.J. & Romero, D.M. Who should I follow? Recommending people in directed social networks.
- Chen, J., Geyer, W. Dugan, C., Muller, M., & Guy, I. 2009. Make new friends, but keep the old: recommending people on social networking sites. Proc. CHI ’09, 201-210.
- Daly E.M., Geyer W., & Millen D.R. The network effects of recommending social connections. Proc. RecSys '10, 301-304.
- Guy I., Ronen I., & Wilcox E. Do you know? recommending people to invite into your social network. Proc. IUI’09, 77-86.
- Hannon, J., Bennett, M., & Smyth, B. Recommending twitter users to follow using content and collaborative filtering approaches. Proc. RecSys '10, 199-206.
- McDonald D.W & Ackerman M.S. 2000. Expertise recommender: a flexible recommendation system and architecture. Proc. CSCW '00, 231-240.
- Quercia, D., & Capra, L. 2009. FriendSensing: recommending friends using mobile phones. Proc. RecSys ’09, 273-276.
- Terveen, L. and McDonald, D. W. Social matching: A framework and research agenda. ACM TOCHI 12, 3, (2007), 401-434.

References: Community Recommendation

- Baatarjav, E.A., Phithakkitnukoon S., & Dantu R. Group recommendation system for Facebook. Proc. OTM ’08, 211-219.
- Chen W.Y., Chu J.C., Luan J., Bai H., Wang, Y., & Chang E.Y. Collaborative filtering for Orkut communities: discovery of user latent behavior. Proc. WWW '09, 681-690.
- Official LinkedInBlog - The engineering behind LinkedIn products “you may like”: http://blog.linkedin.com/2011/03/02/linkedin-products-you-may-like/
- Saha & Getoor. Group Proximity Measure for Recommending Groups in Online Social Networks. Proc. SNA-KDD ’08.
- Spertus, E., Sahami, M., & Buyukkokten O. Evaluating similarity measures: a large-scale study in the Orkut social network. Proc. KDD '05, 678-684.
- Zheng N., Li Q., Liao S., & Zhang L. Flickr group recommendation based on tensor decomposition. Proc. SIGIR '10, 737-738.