IIIT-H

Web Mining
Lecture 8: Social Recommender Systems (Part 2)

Manish Gupta

22nd Aug 2013

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

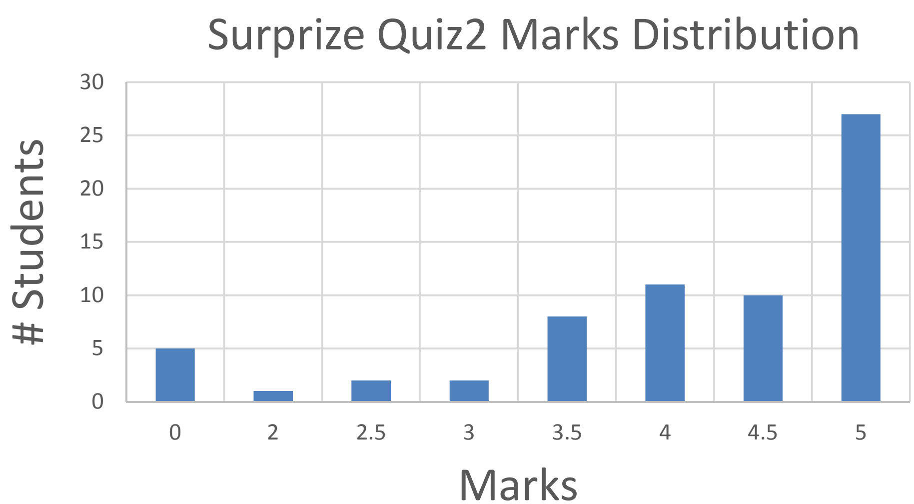
Recap of Lecture 7: Social Recommender Systems (Part 1)

- Introduction to Recommender Systems
- Fundamental Recommendation Approaches
- Content Recommendation
- Tags Recommendation
- People Recommendation
- Communities Recommendation

Announcements

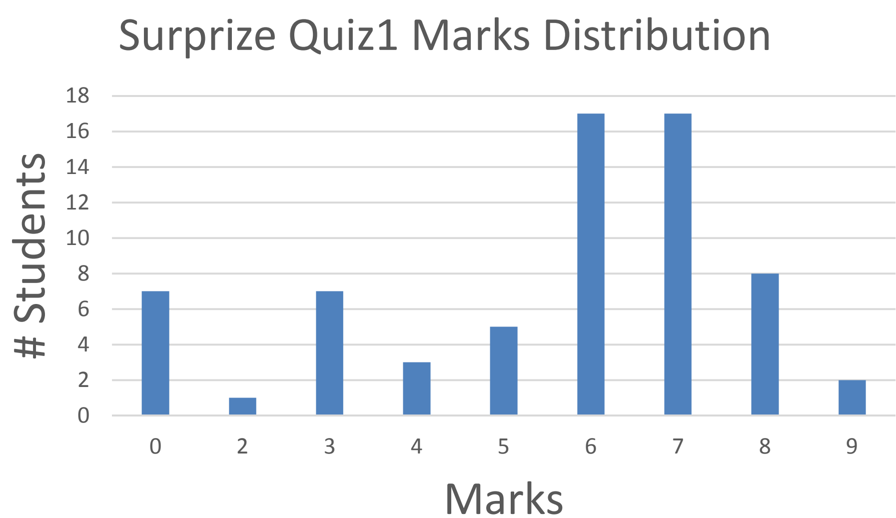
- Assignment 2 will be put up on Aug 28.
- Rescheduling of lectures
  - Makeup class for Aug 28 lecture will be on Sep 2 6-7:30pm
- Surprise Quiz 2 Analysis
  - Q1: Correctly answered by majority (~90%) along with the reasons.
  - Q2: Answered correctly by majority (~80%) but correct reasons by relatively less (~50%) people.
  - Q3: Correctly answered by majority. (~90%)
  - Q4: Correctly answered by less (~40%) people; students are unclear about batch processing and on-line processing.
  - Q5: Correctly answered by majority (~90%) along with the reasons.

Surprise Quiz1 versus Quiz2



Averages (out of 5)

- PG – 4.093
- PGSSP – 2.25
- UG – 4.136
- PhD – 4.5
- Overall – 4.015



Averages (out of 10)

- PG – 6.15
- PGSSP – 0.75
- UG – 5.47
- PhD – 5
- Overall – 5.45

Today's Agenda

- Recommendation for Groups
- The Cold Start Problem
- Trust
- Temporal Aspects in Social Recommendation
- Evaluation Methods

Today's Agenda

- **Recommendation for Groups**
- The Cold Start Problem
- Trust
- Temporal Aspects in Social Recommendation
- Evaluation Methods

Recommendation for a Group rather than Individuals

- Friends collaborating with a recommender system to design the perfect vacation
- Family selecting a movie or TV show to watch together
- Group of colleagues choosing a restaurant for an evening out
- Or looking for a recipe for a joint meal

Issues in Recommendation for Groups

Phase of recommendation	Differences from individuals	General issues
Members specify their preferences	Members can examine each other	Benefits/drawback for the group/system
System generates recommendations	Aggregating preferences/results must be applied	Aggregation methods
Members decide which recommendation (if any) to accept	Negotiation may be required.	How to support the process of arriving at a final decision

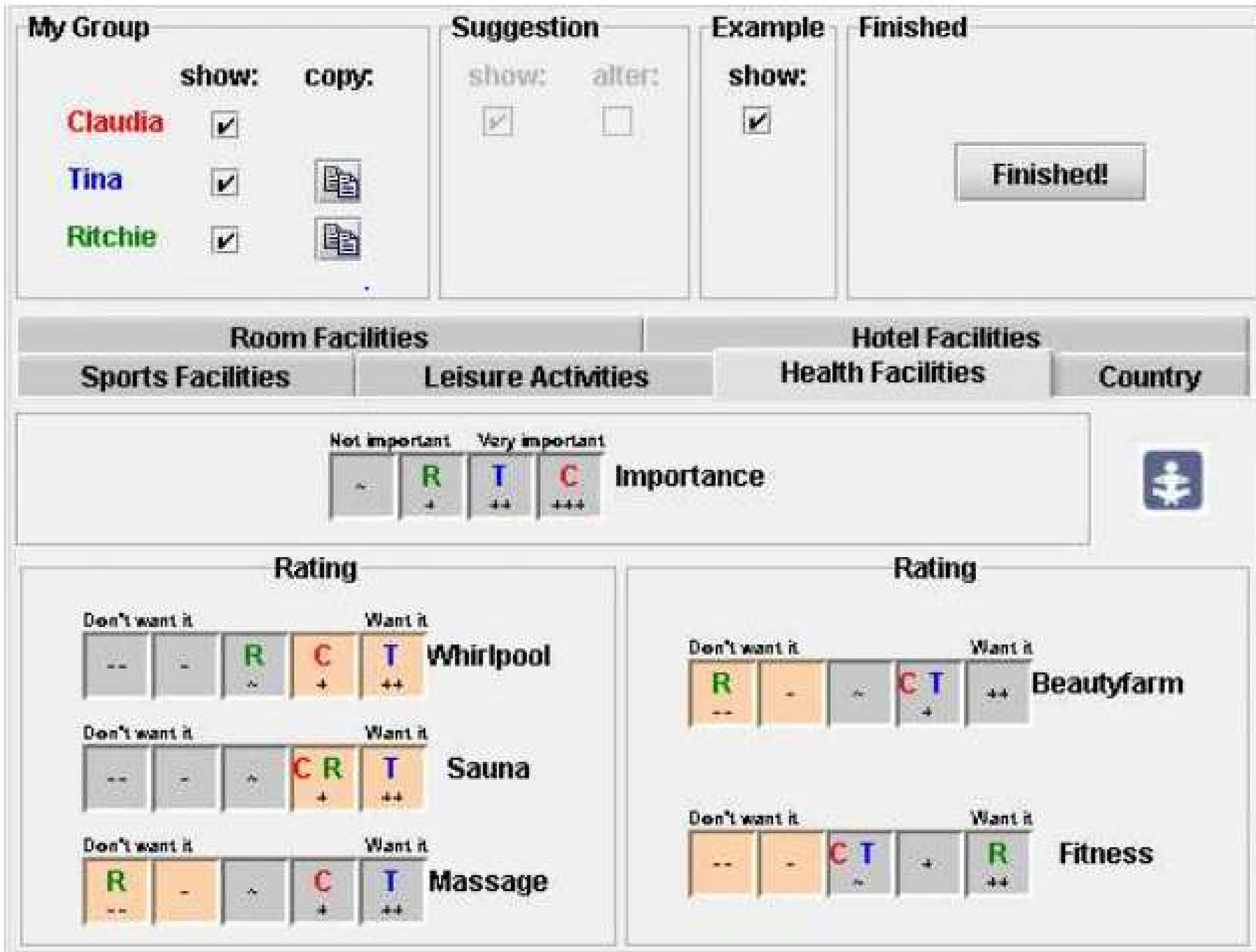
Explicit Specification of Preferences

- MusicFX (McCarthy, CSCW 2000)
  - A group preference arbitration system that allows the members of a fitness center to influence the selection of music in a fitness center

Rating	Interpretation
+2	I *love* this music
+1	I *like* this music
0	I *don't mind / don't care about* this music
-1	I *dislike* this music
-2	I *hate* this music

i	*Genre* *Person*	A	B	C	D	E
1	Alternative Rock	2	2	0	2	2
2	Hottest Hits	1	1	2	0	-2
3	New Music	1	1	1	0	0
4	Hot Country	2	0	0	0	-2
5	Dance	2	-1	1	-1	-1
6	World Beat	0	1	-1	1	-2
7	Traditional Country	1	0	0	-2	-2
8	50's Oldies	0	0	0	-1	-1
9	Heavy Metal	-1	-1	-1	-1	-2
10	Polka	-1	-1	-2	-2	-2

Collaborative Specification of Preferences in the Travel Decision Forum [Jameson 04]



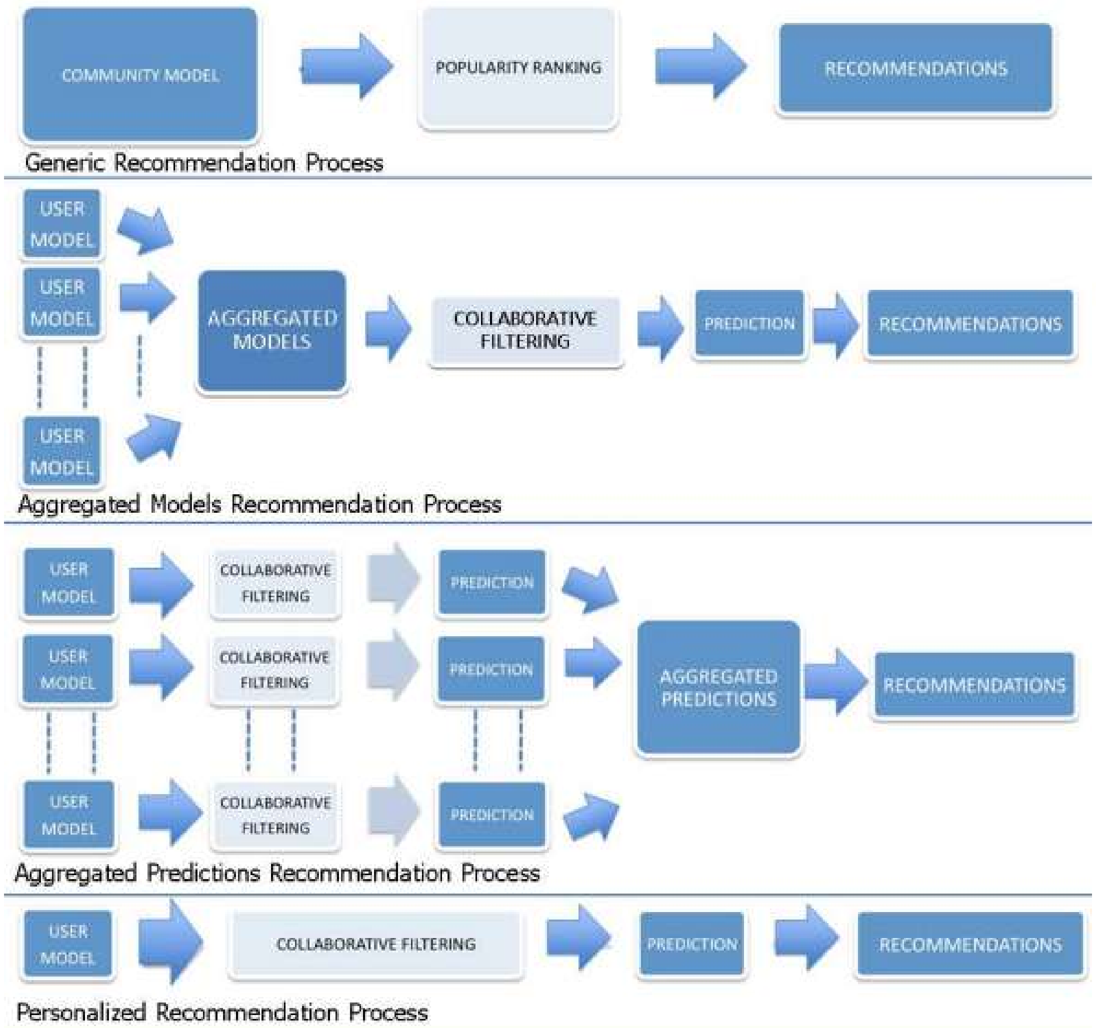
Collaborative Specification - Advantages

- Persuade other members to specify a similar preference, perhaps by giving them information that they previously lacked
- Explain and justify a member's preference
  - (e.g., .I can't go hiking, because of an injury)
- Taking into account attitudes and anticipated behavior of other members
- Encouraging assimilation to facilitate the reaching of agreement

Aggregating Preferences

- Least misery
  - Recommendations are based on the lowest predicted rating
  - Rate item i by R_i =min $\{r_{i_1} \ldots r_{i_k}\}$ (for group of users {1..k})
- Average
  - R_i = avg $\{r_{i_1} \ldots r_{i_k}\}$
- Fairness
  - R_i = avg $\{r_{i_1} \ldots r_{i_k}\} - w \times$ std $\{r_{i_1} \ldots r_{i_k}\}$
- Fusion
  - Aggregate item ranking created for the individuals (e.g., Borda count)
    - The Borda count determines the winner of an election by giving each candidate a certain number of points corresponding to the position in which he or she is ranked by each voter.
- Group modeling
  - Compute an aggregate preference model that represents the preferences of the group as a whole
    - e.g. the system computes a model of the group by forming a linear combination of the individual models

Aggregation Methods [Berkovsky, RecSys '10]



$$pred(item_i) = \frac{\sum_{x \in U} rat(u_x, item_i)}{n}$$

$$rat(f_a, item_i) = \frac{\sum_{x \in f_a} \omega(u_x, f_a)\, rat(u_x, item_i)}{\sum_{x \in f_a} \omega(u_x, f_a)}$$

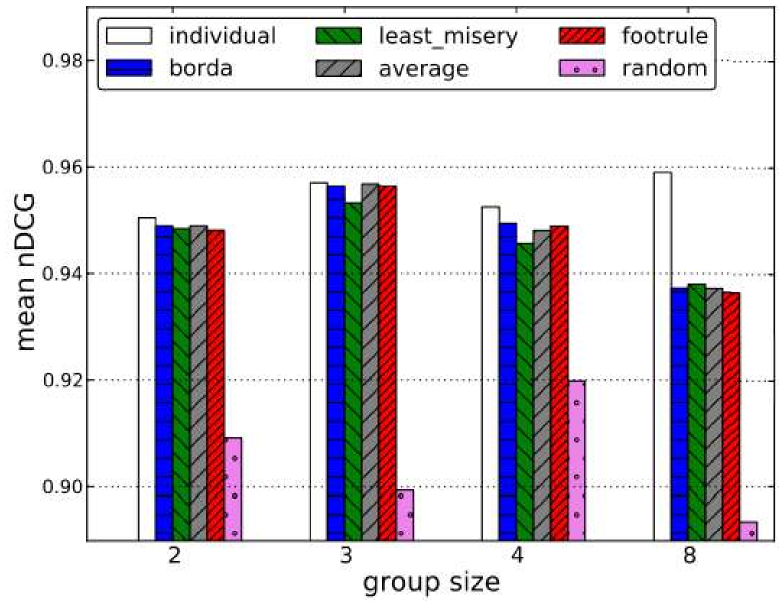
$$pred(f_a, item_i) = \frac{\sum_{f_b \in F} sim(f_a, f_b)\, rat(f_b, p_i)}{\sum_{f_b \in F} sim(f_a, f_b)}$$

$$pred(u_x, item_i) = \frac{\sum_{y \in U} sim(u_x, u_y)\, rat(u_y, item_i)}{\sum_{y \in U} sim(u_x, u_y)}$$

$$pred(f_a, item_i) = \frac{\sum_{x \in f_a} \omega(u_x, f_a)\, pred(u_x, item_i)}{\sum_{x \in f_a} \omega(u_x, f_a)}$$

Group recommendations [Baltrunas et al., RecSys2010]

- Takes as input the individual ranked lists of items' recommendations for each user in the group
- Returns a ranked list of recommendations for the whole group
- Aggregation methods
  - Let $\sigma_u(j)$ denote position of item j in user u's preference list
  - Spearman footrule aggregation aims at minimizing the average Spearman footrule distance to the input rankings
    - Given 2 rankings σ_u and σ_v, Spearman footrule distance is $F(\sigma_u, \sigma_v) = \sum_{i \in I} |\sigma_u(i) - \sigma_v(i)|$



Today's Agenda

- Recommendation for Groups
- **The Cold Start Problem**
- Trust
- Temporal Aspects in Social Recommendation
- Evaluation Methods

The Cold Start Problem

- The Cold Start problem concerns the issue where the RS cannot draw inferences for users or items for which it has not yet gathered sufficient information
- New items
  - e.g., a newly created document w/o tags or bookmarks
  - e.g., a newly created community w/o members
- New users
  - e.g., a user that has just signed up to a new site
  - e.g., a new member or employee
- Typically addressed by applying a hybrid approach

The Cold Start Problem of New Items

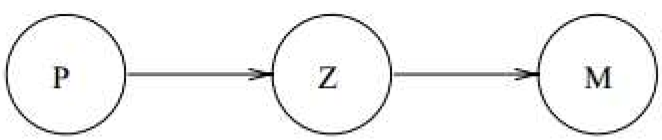
- Traditional CF systems are based on item ratings
  - Until rated by a substantial number of users, the system will not be able to recommend the item
- a.k.a the “early rater” problem – first person to rate an item gets little benefit
- Same for implicit feedback over items – clicks, searches, comments, tags
- Even more acute for activity streams, where items quickly come and go
- Typically addressed by integrating CB similarity measurements
  - Recommendation based on the data of older similar items

The Cold Start Problem of New Items

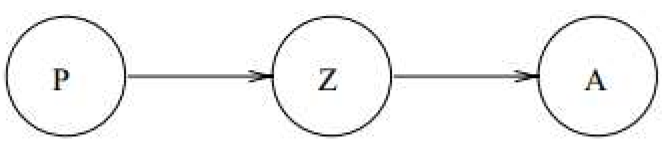
- Methods and metrics for cold-start recommendations [Schein et al., SIGIR '02]
- Extend hybrid RS to average content data in a model-based fashion
- Recommend movies to a user based on how similar the cast is to movies the user has already rated.

Random Variable	Object	Interpretation
P	p	person
M	m	movie
A	a	actor
Z	z	latent class

a)



b)



$$\mathsf{P}(p,m) = \sum_z \mathsf{P}(p)\mathsf{P}(z|p)\mathsf{P}(m|z)$$

$$\mathsf{P}(p,a) = \sum_z \mathsf{P}(p)\mathsf{P}(z|p)\mathsf{P}(a|z)$$

E-Step:

$$\mathsf{P}(z|a,m) \propto \mathsf{P}(a|z)\mathsf{P}(z|m)$$

M-Step:

$$\mathsf{P}(z|m) \propto \sum_a n(a,m)\mathsf{P}(z|a,m)$$

Recommendations are made using:

$$\mathsf{P}(p|m) = \sum_z \mathsf{P}(p|z)\mathsf{P}(z|m)$$

The Cold Start Problem for New Users

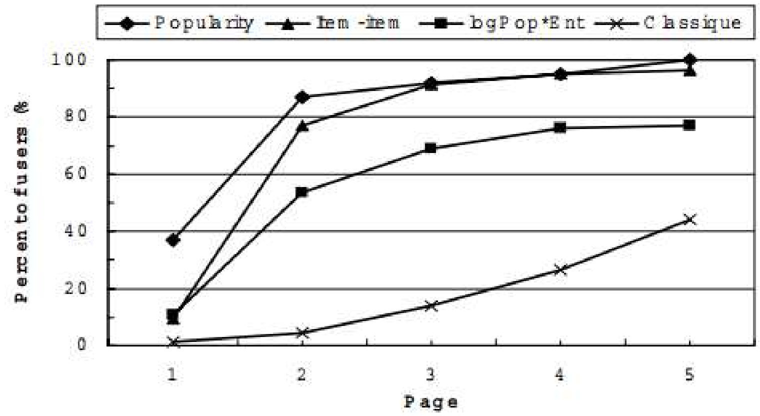
- Sometimes also referred to as the "New User Problem"
- User needs to rate sufficient items for a CB recommender to really understand the user's preferences
- Mitigated by CF – similar users who rated more items can yield more recommendations
- Traditional CF still faces an issue if the user did not provide any explicit feedback (or very small amount of feedback)
- Typically resolved through building a user profile by integrating other user activity (implicit feedback)
  - Browsing history, click-through data, searches
- Social media introduces new ways to learn about the user from external sources
  - Friends ("social filtering"), tags, communities, …
  - More public information which is less sensitive to privacy issues

The Cold Start Problem for New Users

- Increasing engagement through early recommender intervention [Freyne et al., RecSys '09]
- New users of an enterprise SNS
- Use aggregation approach – extract social network data from external sources (project and patent DB, org chart, wikis and blogs)
- A brand new employee – still has org chart and basic profile data (location, division, project, etc.)

The Cold Start Problem for New Users

- Getting to know you: learning new user preferences in recommender systems [Rashid et al., IUI ’02]
- Experimentation with MovieLens (offline and online)
- Focus on minimizing user effort, while maximizing accuracy: When a new user joins MovieLens, the system presents pages of ten movies until the user rates ten or more movies
- Different techniques that CF can use to learn about new users
  - Random
  - Classique: For each page of movies presented, select one movie randomly from an ad hoc manual list of popular movies and the rest randomly from all movies
  - Popular (#ratings), add less information
  - Entropy (diverse ratings), add a lot of information
  - Log Pop*Ent – product of popularity and entropy
  - Item-Item personalized – once the user has rated one movie (before – random from top 200 of Pop*Ent)



Strategy	User Effort	Accuracy
Random/Classique	★	★★
Popularity	★★★★★	★★★★
(log) Pop*Ent	★★★	★★★★★
Item-Item	★★★★★	★★

Cold Start for Tag-based Recommenders

- Tag-based recommenders are sensitive to both new item and new user cold start problems
  - New items are "tagless"
  - New users are not associated with tags
- Possible Solutions
  - Hybridize with a CB recommender
  - Use CB automatic tag extraction
  - Apply CF for tags to enrich tags
    - Will not work for brand new users or items

Today's Agenda

- Recommendation for Groups
- The Cold Start Problem
- **Trust**
- Temporal Aspects in Social Recommendation
- Evaluation Methods

Social Relations based Recommendation

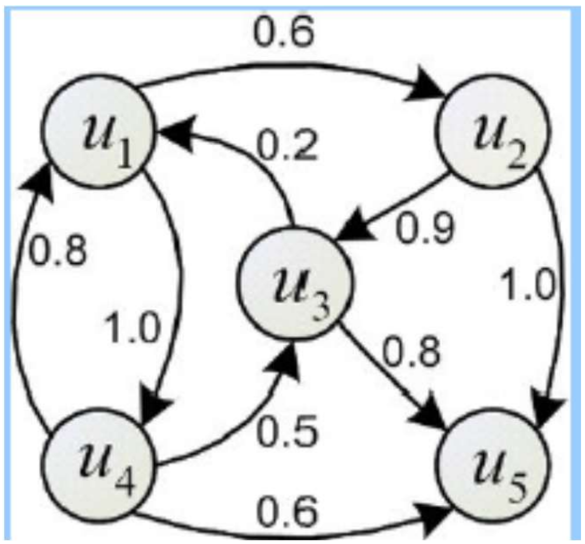
- Traditional recommender systems ignore the social connections between users
- To improve the recommendation accuracy users' social network should be taken into consideration
  - People who are socially connected are more likely to share the same or similar interests
  - Users can be easily influenced by the friends they trust, and prefer their friends' recommendations.

Trust Enhanced Recommendation

- The question of whom to trust and what information to trust has become both more important and more difficult to answer
- Social trust relationships, derived from social networks, are uniquely suited to speak to the quality of online information
- Merging social networks based trust and recommender systems can improve the accuracy of recommendations and improve the user's experience

Problem Definition

Social trust graph



User-item rating matrix

	v_1	v_2	v_3	v_4	v_5	v_6
u_1		5	2		3	
u_2	4			3		4
u_3			2			2
u_4	5			3		
u_5		5	5			3

TidalTrust: Accumulate Recommendation from Trusted People only (Golbeck 06)

- Trust-based predicted rating: $p_{ai} = \frac{\sum_{u \in R_a} t_{au} r_{ui}}{\sum_{u \in R_a} t_{au}}$
- With CF: $p_{ai} = \overline{v_a} + \frac{\sum_{u \in R_a} t_{au}\,(v_{ui} - \overline{v_u}) + \sum_{u \in R - R_a} w_{au}\,(v_{ui} - \overline{v_u})}{\sum_{u \in R_a} t_{au} + \sum_{u \in R - R_a} w_{au}}$
- Items recommended by trusted neighbors will be boosted
- Note that we still can get recommendations from similar non trusted neighbors (w(a,u) << t(a,v) if v is trusted and u is not)
- Trust is highly correlated with similarity
  - people usually have more trust in similar people
- However, trust captures nuances that similarity cannot assess

Inferring Trust

- When two individuals know each other, they can assess the trustworthiness of one another.
- Goal: Select two individuals – source and sink and then recommend to the sink how much to trust the source.
- If the source does not know the sink, the source asks all of its friends how much to trust the sink, and computes a trust value by a weighted average
- Neighbors repeat the process if they do not have a direct rating for the sink
- Set a minimum trust threshold and only consider paths where all edges have trust ratings at or above the threshold
- $t_{is} = \frac{\sum_{j \in adj(i) | t_{ij} \geq max} t_{ij} t_{js}}{\sum_{j \in adj(i) | t_{ij} \geq max} t_{ij}}$

- Max = largest trust value that can be used as a minimum threshold such that a path can be found from source to sink

Reputation

- We can measure the reputation ("global trust") of the user using network analysis
  - e.g. PageRank
- These reputation values can be used to bias recommendations from highly trusted (reliable) people

What about Distrust? (Victor 2009)

- Use the distrust set to filter out "unwanted" individuals from collaborative recommendation processes
- Distrust as an indicator to reverse deviations
  - Consider distrust scores as negative weights
    - Reduce the recommendation score of items recommended by distrusted neighbors
- Using distrust for recommendation is still an open challenge

Trust in Recommendation (by Explanations)

- MoviExplain: A Recommender System with Explanations (Symeonidis 09)
- Good explanations could help inspire user trust and loyalty, increase satisfaction, make it quicker and easier for users to find what they want, and persuade them to try or purchase a recommended item

Our Justified Recommendations

[Movie id]	[Movie title]	[The reason is]	[because you rated]
1526	Witness (1985)	Ford, Harrison (I)	21 movies with this feature
1273	Color of Night (1994)	Willis, Bruce	7 movies with this feature
1004	Geronimo: An American Legend (1993)	Hackman, Gene	7 movies with this feature
1442	Scarlet Letter, The (1995)	Oldman, Gary	7 movies with this feature
1044	Paper, The (1994)	Close, Glenn	7 movies with this feature
693	Casino (1995)	De Niro, Robert	6 movies with this feature
274	Sabrina (1995)	Pollack, Sydney	6 movies with this feature
1092	Dear God (1996)	Kinnear, Greg	5 movies with this feature

Explanation Aims

- Transparency
  - Explain how the system works
- Curability
  - Allow users to tell the system it is wrong
- Trust
  - Increase users' confidence in the system
- Effectiveness
  - Help users make good decisions
- Persuasiveness
  - Convince users to try or buy
- Efficiency
  - Help users make decisions faster
- Satisfaction
  - Increase the ease of usability or enjoyment

Explanation Types

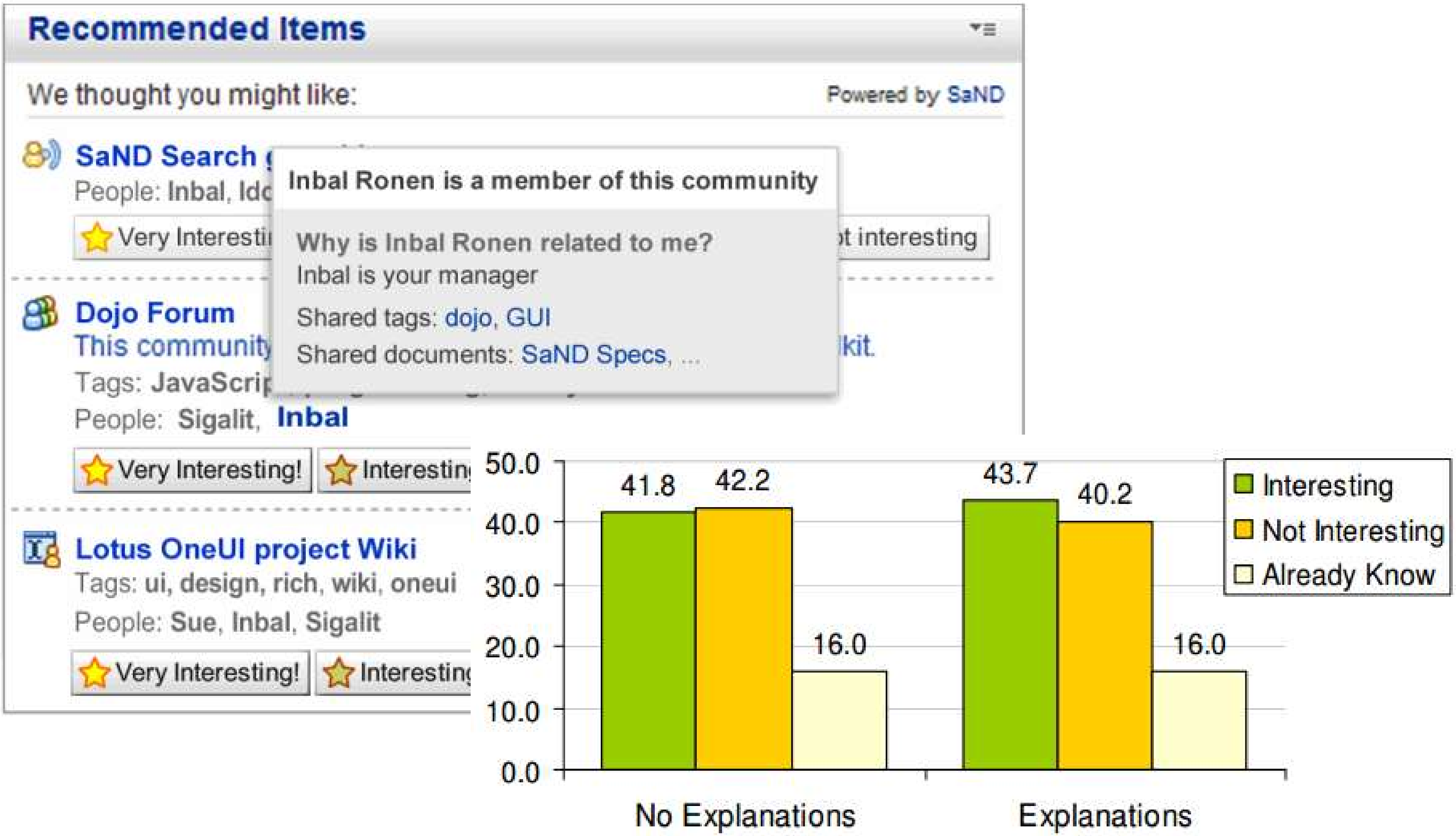
- Nearest neighbor explanation
  – Customers who bought item X also bought items Y, Z
  – Item Y is recommended because you rated related item X
- Content based explanation
  – This story deals with topics X, Y which belong to your topic of interest
- Social based explanation
  – People leverage their social network to reach information and make use of trust relationships to filter information
    - Your friend X wrote that blog
    - 50% of your friends liked this item (while only 5% disliked it)

TagSplanations: Explaining Recommendations using Tags (Vig 09)

- Components of Tagsplanations
  - Tag relevance: depends on tag popularity and preference correlation
    - Preference correlation is the correlation between users' preference for the tag and their preference for the movie
  - Tag preference: the user's sentiment toward a tag computed using user's movie ratings
  - Tag Filtering
    - Tag Quality
    - Tag Redundancy
    - Usefulness for explanations

Rushmore

Social-based Explanation (Guy 09)



Today's Agenda

- Recommendation for Groups
- The Cold Start Problem
- Trust
- **Temporal Aspects in Social Recommendation**
- Evaluation Methods

The Time Factor in RS

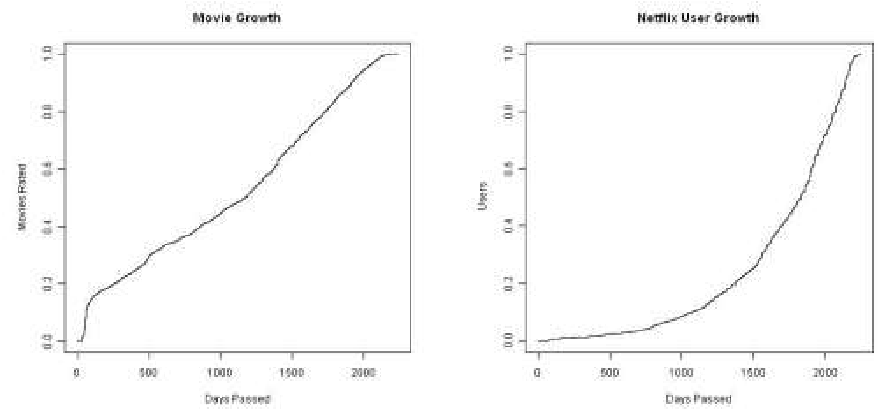
- Most RS works ignore temporal factors
- In practice, time yields many changes
  - Changes in user tastes and needs
    - e.g., a scientist moving to a new domain
  - Appearance of new items and users, disappearance of others
    - e.g., a new action movie about aliens
  - Changes in items and their features
    - e.g, a restaurant changing its menu
- The pace of changes is greater in the social media world, where the masses are involved in constantly creating new content, communities, comments, tags, etc.
- Real-time web mediums like Twitter even further magnify the importance of time

Temporal CF

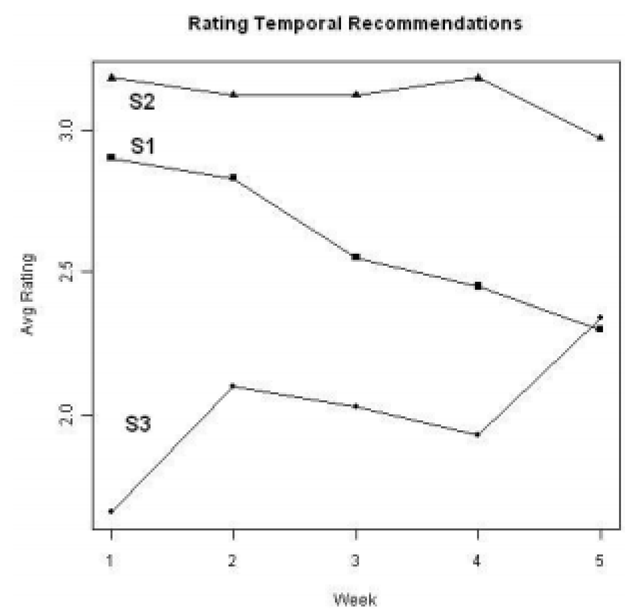
- Collaborative filtering with temporal dynamics [Koren, KDD'09]
- Tracking customer preferences for products over time
- Both users and products go through a distinct series of changes in their characteristics
- A mere decay model loses too much signal
- Separate transient factors from lasting ones
  - Matrix factorization – model both user and product change along time, to distill longer term trends
  - Item-item neighborhood – learning how influence between two items rated by a user decays over time, to reveal the fundamental item-item relations
- Leading to improved results over Netflix in both models

Temporal Diversity

- Temporal diversity in recommender systems [Lathia et al., SIGIR '10]
- Over time, the same items might be recommended again and again
  - Users may lose interest in interacting with the RS
- Temporal diversity is therefore important
  - Netflix data changes over time: #movie grows, #users grows, #ratings grows
- User survey – simulated 5 "weeks"
  - S1 – top-10 most popular movies (no diversity)
  - S2 – ~7 popular movies replaces each week
  - S3 – random out of the Netflix dataset

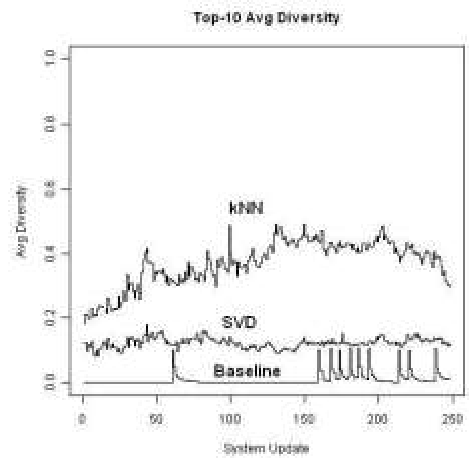


(a) Movie Growth (b) User Growth

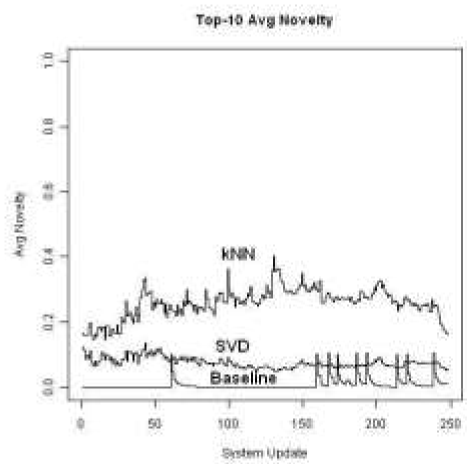


Temporal Diversity

- Comparing diversity of 3 CF algorithms over time
  - Baseline – item's mean rating
  - Item-item based kNN
  - Matrix factorization based on SVD
- Given 2 recommendation lists L_1 and L_2
  - diversity(L_1, L_2, N)=$\frac{|L_2 \backslash L_1|}{N}$
  - Novelty(L_1, N)=$\frac{|L_1 \backslash A_t|}{N}$
    - A_t is the list of all items recommended to the user till now
- Growing Netflix dataset 249 system updates each after 7 days



Top-10 Diversity



Top-10 Novelty

Temporal User Profile

- Adaptive web search based on user profile constructed without any effort from users [Sugiyama et al., WWW'04]
  - Personalized search based on browsing history
  - Given a query, search results are adapted based on the user's information needs
  - User profile evolves over time, combining
    - Persistent user behavior computed over a fixed time decay window
    - Ephemeral aspects captured from the current session (day)
- Personalizing search via automated analysis of interests and activities [Teevan et al., SIGIR '05]
  - Wide range of implicit user activities over short and long time periods using a relevance feedback framework
    - Search queries, visited pages – "implicit" short-term relevance feedback
    - documents and email – longer term interests

Future Directions

- Enhance user modeling approaches to consider time in a smarter way
  - e.g., someone you “friended” 5 years ago – a very good friend or a one-time encounter?
    - Measure interaction over time
  - e.g., tag usage as a signal to changes of interest over time
    - Distinguish transient vs. lasting interests
- Learn from user feedback to compensate for decay in interest
  - Exploitations vs. exploration
- Different recommendation for new users, old users, heavy users, etc.
- Impose diversity and avoid recommending the same items again and again
- More live user studies over time

Today's Agenda

- Recommendation for Groups
- The Cold Start Problem
- Trust
- Temporal Aspects in Social Recommendation
- **Evaluation Methods**

Evaluation Goals

- An application designer who wishes to add a recommendation system to her application must make a decision about the most appropriate algorithm for her goals
- Most evaluation methods rank systems based on
  - Prediction power — the ability to accurately predict the user's choices
  - Classification accuracy – the ability to differentiate good items from bad ones.
  - Novelty and Exploration ability -discovering new items, and exploring diverse items
  - Other features
    - Preserving the user privacy
    - Fast response
    - Ease of interaction with the recommendation engine

Offline Evaluation

- Based on a pre-collected data set of users choosing or rating items
  - Usually done by recording historical user data, and then hiding some of these interactions in order to compare the user predicted rating with her actual rating
- No interaction with real users, thus allow comparing a wide range of candidate algorithms at a low cost
- Mostly useful for evaluating the prediction power of the system and for system tuning
- For example: the Netflix challenge

The Netflix Challenge (Bernett 07)

- Open competition for the best collaborative filtering algorithm to predict user ratings for films
  - based on previous movie ratings (1-5 scale)
- Netflix provided a training data set of
  - 100,480,507 ratings
  - by 480,189 users
  - for 17,770 movies
- Contestants were judged according to their predictions for 3 million withheld ratings
- $1 million prize was given to a team who achieved 10% improvement over the accuracy of the Netflix movie recommendation system

Evaluating Prediction Accuracy the User Opinions/Ratings over Items

- Root Mean Squared Error (RMSE) the most popular metric used in evaluating accuracy of predicted ratings
  - $RMSE = \sqrt{\frac{1}{N}\sum_{(u,i)\in N}(r_{ui} - \hat{r}_{ui})^2}$
- A popular alternative: Mean Absolute Error (MAE)
  - $MAE = \sqrt{\frac{1}{N}\sum_{(u,i)\in N}|r_{ui} - \hat{r}_{ui}|}$
- Both measure the average error of the system predictions
- RMSE disproportionably penalizes large errors, compared to MAE

Measuring Ranking Accuracy

- Evaluating how many recommended items were purchased by the user
  - Recall -how many of acquired items were recommended
  - Precision – how many recommended items were acquired
    - Prec@N – how many top-N recommended items were acquired
  - A trade off is expected
    - long recommendation lists typically improve recall while reduce precision
- For a ranked list of recommended items use NDCG (Normalized discount Cumulative Gain)

Evaluating Top-N Recommendation (Cremonesi, RecSys2010)

- For each item i rated by user u
  - Randomly select 1000 additional items unrated by user u.
    - We may assume that most of them will not be of interest to user u.
  - Predict the ratings for the test item i and for the additional 1000 items.
  - Form a ranked list by ordering all the 1001 items according to their predicted ratings.
    - The best result corresponds to the case where item i precedes all the random items
  - Form a top-N recommendation list by picking the N top ranked items from the list.
    - If rank(i) $\leq$N we have a hit (i.e., the test item i is recommended to the user).
    - Otherwise we have a miss
- Recall/Precision are defined by averaging over all rated items T
  - recall@N = #hits / |T|
  - precision@N = #hits/ (|T|*N)

Off-line evaluation of a Tag Recommendation System using Social Bookmarks (Carmel, CIKM'09)

- Given a dataset of bookmarks {(u,d,t)}
- For each (u,d) pair:
  - Hide all bookmarks related to this pair from the bookmark data (all (u,d,t') triplets)
  - Recommend tags for this pair
  - Score the recommendation lists by Mean Average Precision (MAP), given the actual tags used by u to bookmark d

User Studies

- A user study is conducted by recruiting a set of users, and asking them to perform several interaction tasks with the recommendation system.
  - While the subjects perform the tasks, we observe and record their behavior
    - the portion of the task completed
    - the accuracy of the task results
    - the time taken to perform the task
    - general impression
    - etc.

Subjective Opinions

- I understand why the products were returned through the explanations in the interface
- This interface gave me some really good recommendations
- I felt in control of specifying and changing my preferences in this interface
- I find this interface easy to use
- This interface is competent to help me effectively find products I really like.
- I find this interface is useful to improve my “shopping” performance.
- I found my visit to this interface enjoyable
- I am confident that the product I just “purchased” is really the best choice for me
- I easily found the information I was looking for
- I feel that this interface is trustworthy
- I trust the recommended products since they were consistent with my preferences.
- My overall satisfaction with the interface is high
- I would purchase the product I just chose if given the opportunity
- If I had to search for a product online in the future and an interface like this was available, I would be very likely to use it
- If I had a chance to use this interface again, I would likely make my choice more quickly.

User studies – Pros and Cons

- Enable online test of the user behavior when interacting with the recommendation system
- Are very expensive to conduct
  - typically restricted to a small set of subjects and a relatively small set of tasks, and cannot test all possible scenarios
  - The test subjects must represent the population of users of the real system
    - as closely as possible

Online Evaluation

- Evaluate the system by real users that perform real tasks
  - Provides the strongest evidence for the true value of the system to its users
  - The real effect of the recommendation system depends on a variety of user's dependent factors that are changed dynamically
    - The user current intent
    - The user's current context
- Feedback from the users is collected by observing their feedback to the system's recommendation
  - Systems are evaluated according to the acquired vs. non-acquired ratio
- Such a live user experiment may be controlled
  - Randomly assign users to different conditions
    - e.g. test a new version of your system on a test set of users
  - A/B testing: split users to test groups and measure effectiveness of different conditions/algorithms on the groups
- On-line evaluation studies are done on a regular basis by commercial Recommendation Systems

Take-away Messages

- Recommendation for Groups involves challenges like how to input group preferences, how to aggregate preferences and how to generate a single acceptable recommendation for all
- The Cold Start Problem for new items and new users is addressed by using hybrid approaches
- Trust in recommendation systems can be enhanced by using social relations and providing explanations
- Products and users and their liking change with time and so recommenders need to consider the time dimension
- We looked at various Evaluation Methods for recommendation systems

Further Reading

- Gediminas Adomavicius and Alexander Tuzhilin. 2005. Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. IEEE Trans. on Knowl. and Data Eng. 17, 6 (June 2005), 734-749. DOI=10.1109/TKDE.2005.99 http://dx.doi.org/10.1109/TKDE.2005.99
- Yading Song, Simon Dixon, and Marcus Pearce. A Survey of Music Recommendation Systems and Future Perspectives. 9th International Symposium on Computer Music Modelling and Retrieval (CMMR 2012). Pages 395-410. http://www.eecs.qmul.ac.uk/~simond/pub/2012/Song-Dixon-Pearce-CMMR-2012.pdf
- Joseph A. Konstan, John Riedl. Recommender systems: from algorithms to user experience. User Model User-Adap Inter (2012) 22:101–123. http://www.grouplens.org/system/files/algorithmstouserexperience.pdf

Preview of Lecture 9: Social Networks

- Introduction to social networks, power laws
- Diameter, hop-plots, bow-tie, preferential attachment
- User behavior analysis

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: Group Recommendations

- MusicFX: An Arbiter of Group Preferences for Computer Supported Collaborative Workouts, McCarthy et al. CSCW 98
- Two Methods for Enhancing Mutual Awareness in a Group Recommender System, Jameson et al., AVI 2004
- Recommendation to groups Jameson and Smyth, 2007
- Group recommendations with rank aggregation and collaborative filtering (Baltrunas et a.l, RecSys2010)
- Group-Based Recipe Recommendations: Analysis of Data Aggregation Strategies. Berkovsky et al, RecSys2010

References: Cold Start Problem

- Freyne J., Jacovi M., Guy I., and Geyer, W. Increasing engagement through early recommender intervention. Proc. RecSys '09, 85-92.
- Schein A.I., Popescul A., Ungar L.H, & Pennock D.M. Methods and metrics for cold-start recommendations. Proc. SIGIR ’02, 253-260.
- Rashid A.M., Albert I., Cosley D., Lam S.K., McNee S.M., & Konstan, J.A. Getting to know you: learning new user preferences in recommender systems. Proc. IUI ’02, 127-134.

References: Trust

- Trust and Distrust Based Recommendations for Controversial Reviews. Victor et al. IEEE Intelligent Systems
- Inferring binary trust relationships in Web-based social networks. Golbeck et al. TOIT 2006
- MoviExplain: a recommender system with explanations. Symeonidis et al. RecSys 2009
- Tagsplanations: explaining recommendations using tags. Vig aet al. IUI 2009
- Personalized recommendation of social software items based on social relations. Guy et al. RecSys 2009

References: Social Recommender Systems in the Enterprise

- Freyne J., Jacovi, M., Guy I., & Geyer W. 2009. Increasing engagement through early recommender intervention. Proc RecSys '09, 85-92.
- Geyer, W., Dugan, C., Millen, D., Muller, M., & Freyne, J. Recommending topics for self-descriptions in online user profiles. Proc. RecSys '08, 59-66.
- Blog Muse
- Guy I., Ur, S., Ronen, I., Perer, A., & Jacovi, M. Do you want to know? recommending strangers in the enterprise. Proc. CSCW '11.

References: Temporal Aspects in Social Recommendation

- Koren Y. Collaborative filtering with temporal dynamics. Proc. KDD ’09, 447-456.
- Lathia N., Hailes, S., Capra L., & Amatriain X. 2010. Temporal diversity in recommender systems. Proc. SIGIR ’10, 210-217.
- Sugiyama K., Hatano K., & Yoshikawa M. Adaptive web search based on user profile constructed without any effort from users. Proc. WWW '04, 675-684.
- Teevan, J., Dumais, S. T., & Horvitz, E. Personalizing search via automated analysis of interests and activities. Proc. SIGIR ’05, 449-456.

References: Social Recommendation over Activity Streams

- Bernstein M.S., Suh B., Hong L., Chen, J., Kairam S., & Chi E.H. Eddi: interactive topic-based browsing of social status streams. Proc. UIST ’10, 303-312.
- Chen J., Nairn R., Nelson L., Bernstein, M., & Chi E. Short and tweet: experiments on recommending content from information streams. Proc. CHI ’10, 1185-1194.
- Garcia Esparza, S., O'Mahony, M. P., & Smyth, B. 2010. On the real-time web as a source of recommendation knowledge. Proc. RecSys '10, 305-308.
- Phelan O., McCarthy K., & Smyth B. 2009. Using twitter to recommend real-time topical news. Proc. RecSys '09, 385-388.

References: Evaluation Methods

- The Netflix Prize, Bernett et al. KDD CUP 2007
- Evaluating collaborative filtering recommender systems. Herlocker et al . TOIS 2004
- Personalized social search based on the user's social network. Carmel et al., CIKM 2009
- User Evaluation Framework of recommender Systems, Chen et al. (SRS 2010)
- Performance of recommender algorithms on top-n recommendation tasks, Cremonesi et al. RecSys 2010

Social Recommender Systems in the Enterprise

Social Media in the Enterprise

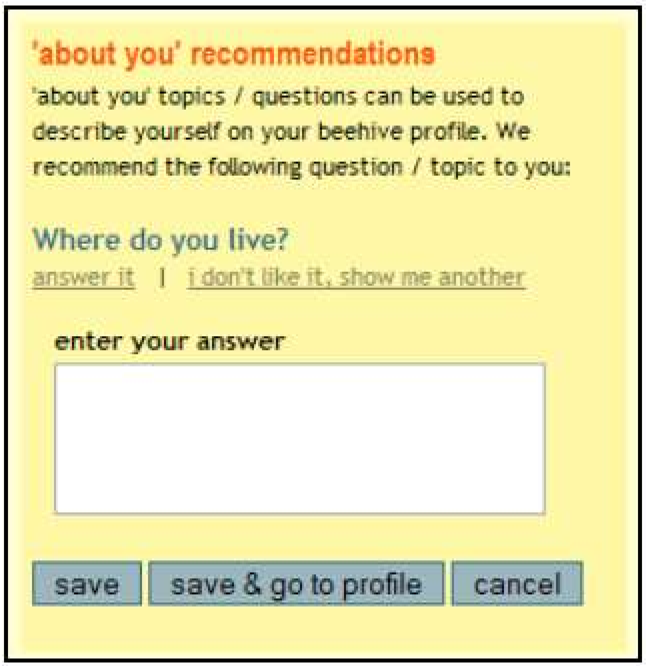
- Following their success on the web, social media application have emerged in large enterprises
  - Corporate blogs and wikis (Blog Central)
  - Social bookmarking (Dogear)
  - Social network sites (Fringe, Beehive/SocialBlue, Town Square)
  - People tagging
  - Social file sharing
- Several differences observed from use outside the firewall
  - e.g., people seek more strongly to connect with strangers
- Information overload analogously created within the enterprise
- Enterprise social media products in market: Microsoft's Yammer, IBM Connections, Jive

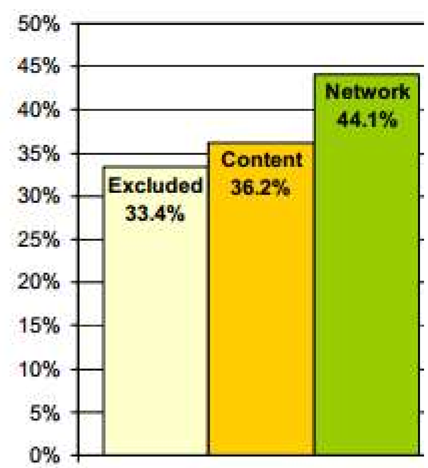
Social Recommenders in the Enterprise

- Same user identity in all systems
  - Facilitates aggregation
  - Higher accountability
- Lower scales, but also lower density
- Cold start for new employees
  - Mitigated with directory data (location, division, role, etc.)

Recommending Content to Create

- Recommending topics for self-descriptions in online user profiles [Geyer et al., RecSys ’08]
- “About You” Q&A pairs in SocialBlue SNS
- Content algorithm – user profile based on all SocialBlue content
- Network algorithm – if one or more friends have created this ‘About You’
- Network algorithm performed best





Stranger Recommendation

- Do you want to know? Recommending strangers in the enterprise [Guy et al., CSCW ’11]
- Recommendation of people who are unknown yet interesting in the organization
- Maybe useful to
  - Get help or advice
  - Reach new opportunities
  - Discover new routes for career development
  - Learn about new assets that can be leveraged
  - Connect with influencers
  - Cultivate organizational social capital
  - Grow own reputation and influence within the organization
- Complements recommendation of people to connect with, as those are quickly exhausted over time

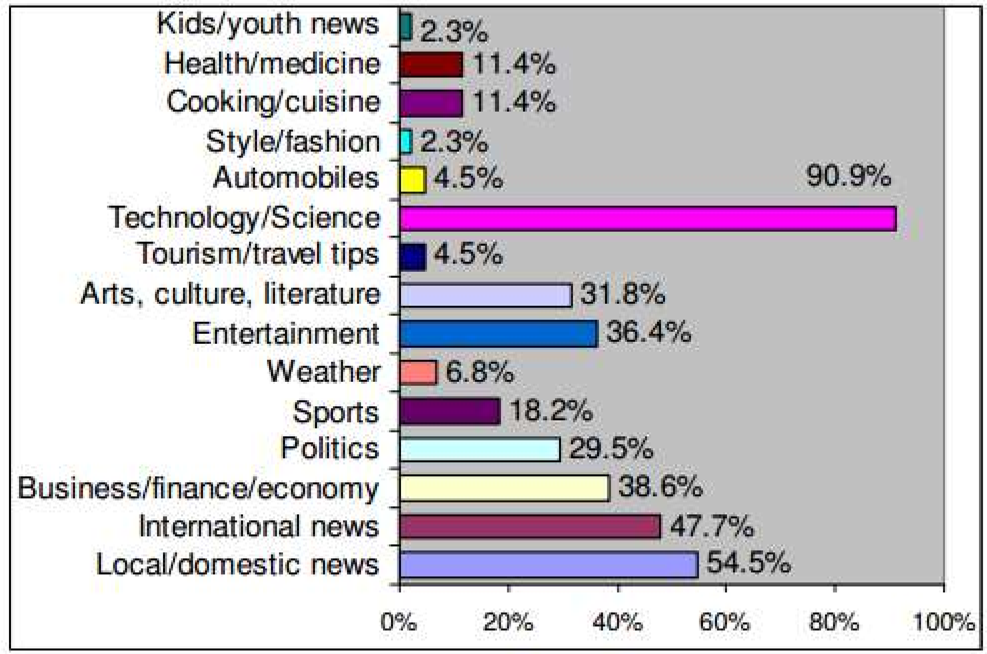
Stranger Recommendation

- Method: subtracting the familiarity network from the similarity network
- Similarity: common things and places: tags, communities, wikis
- Score based on Jaccard's index
- Presentation with evidence
- Two-thirds of the recommendations are strangers
- Significantly more interesting than a random person
- Out of 9 recommendations, 67% got at least one stranger rated 3 or above
- Exploratory recommendation
  - Low accuracy, high serendipity

Social Recommendation over Activity Streams

Utilizing the Stream for Recommendation

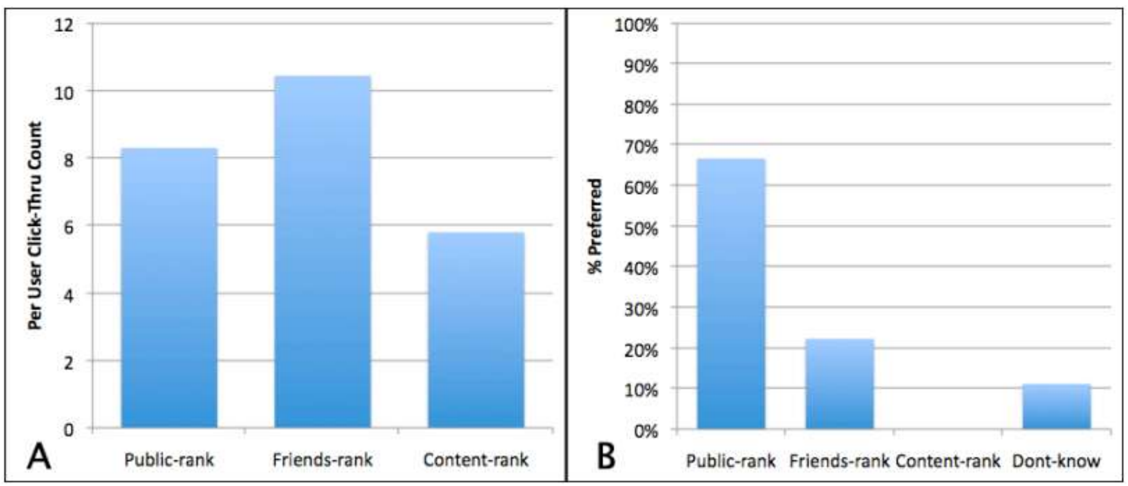
- Activity Streams: Twitter, Facebook, FriendFeed
- Stream data is intensive, fresh, and represents the "crowds"
- Yet, it is also noisy and sparse in content and meta-data
- Using the stream smartly can enhance current SRS
  - Especially for modeling recent interests and trends
- Stream data requires special techniques
  - Frequency is high, so is noise
  - Recency plays special role
  - Content is sparse, no tags (apart from #tags)



Percentages of Field Study Subjects who Use Twitter to Track Different Types of News

Enhancing News Recommendation

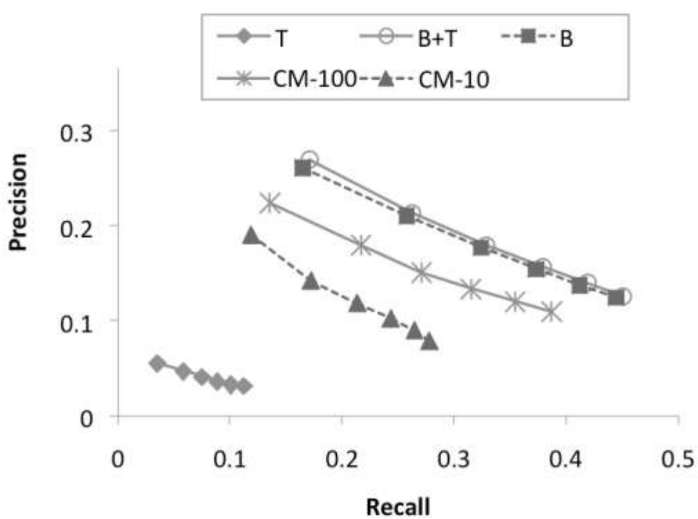
- Using twitter to recommend real-time topical news. [Phelan el al., RecSys '09]
- CB approach based on co-occurrence of popular terms within the user's RSS and Twitter items
- 3 Recommendation strategies
  - Most recent public tweets (PublicRank)
  - Tweets from the user's followees (FriendsRank)
  - Non-tweet, based on top 100 RSS terms (ContentRank)
- First indication that personalized Twitter-based profile can substantially enhance recommendations

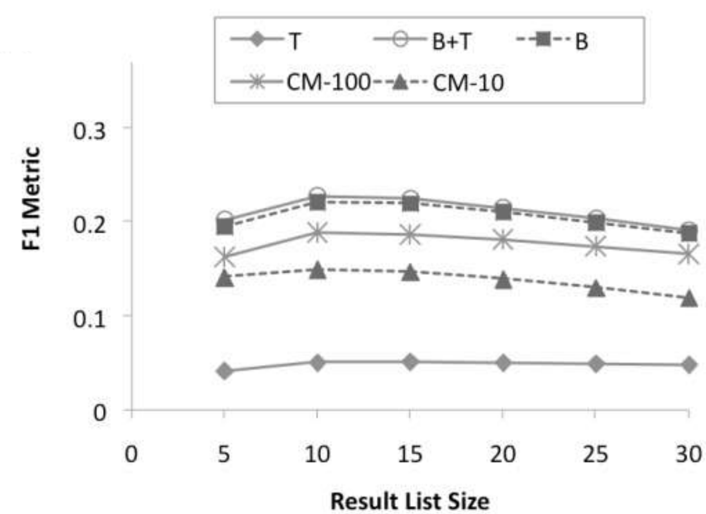


A) Average per user click-through for different recommendation strategies. B) Preferred recommendation strategies

Enhancing Movie Recommendations

- On the real-time web as a source of recommendation knowledge [Esparza et al., RecSys '10]
- Blippr – a Twitter-like short textual movie review service, also
- Users are represented based on their blips (and tags)
- Initial results show superiority over CF
- Adding tags did not improve the performance in this case supporting tags
- CM-x: Community-based CF algorithm with x neighbors



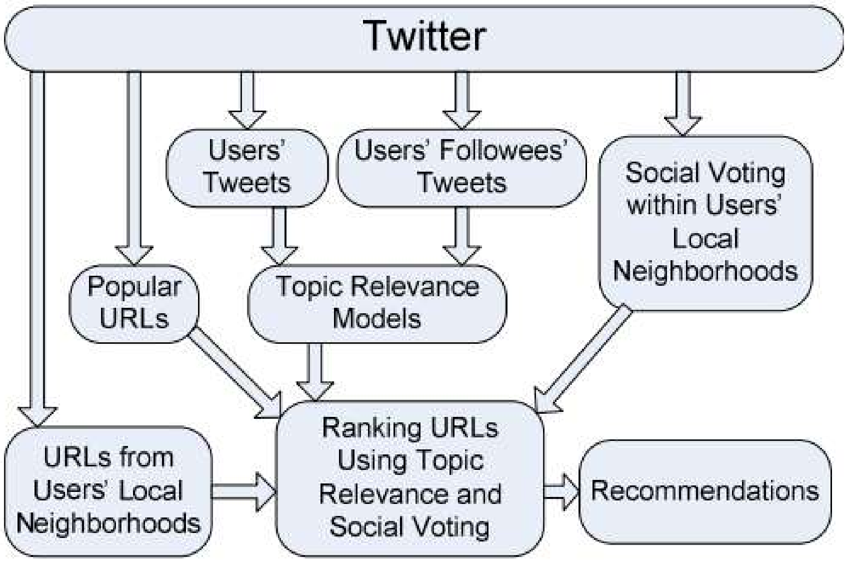


Personalized Filtering of the Stream

- Current stream filtering is done based on the network of "followees" or friends, which is insufficient
- On the one hand, users often receive hundreds of news items per day
  - Much beyond what users have time to process
  - Need to filter the stream to those items that are indeed of interest
- On the other hand, there may be other useful news outside the circle of friends or followees
- Challenges:
  - Frequency – huge flood of news items,
  - Recency plays big role – news items are interesting only when very fresh
  - News items are sparse in content and unstructured (sparse in metadata)
  - Cold start problem of new items

Recommending Twitter URLs

- Short and tweet: experiments on recommending content from information streams [Chen et al., CHI '10]
- Directly recommend content through URLs
- Candidate selection
  - Followees and followees-of-followees
  - Popular tweets
- Topic Relevance
  - Cosine similarity between user and URL
  - Both based on TF-IDF
  - User profile based on self tweets and followees' tweets
  - URL based on tweets (independent of page content)
- Social process
  - Based on "votes" by followees of followees (fof)
  - A vote increases as an fof is followed by more of the user's followees
  - A vote increases as an fof tweets less frequently



Recommending Twitter URLs

- Comparing 12 algorithms: (2 candidate) * (3 topic relevance) * (2 social process)
- Field study, 44 subjects
- At least 20 followees and 50 tweets
- Rating top-5 URLs from each algorithm
  - Interesting or not interesting
- Each URL is shown with up to 3 tweets
  - From the user's fof, if available
- Social process beats topic relevance
- FoF beats popularity
- Self beats followee