Introduction to the Social Web

Recommendation and Mining

Sihem Amer-Yahia

CNRS/LIG

Nov 9th, 2016

Nov 15th, 2016

Instructor: Sihem Amer-Yahia

- Ph.D. in CS, 1999, Univ. of Paris-Orsay & INRIA, France
- Research Scientist, at&t labs: 1999-2006
- Senior Research Scientist, Yahoo! Research: 2006-2011
- Principal Research Scientist, QCRI: 2011-12
- Since Dec 2011: DR1 CNRS@LIG
  - Big Data Management and Query Processing for Search and Recommendation and their application to Social Computing, Large-scale information exploration algorithms
  - Head of the SLIDE team (ScaLable Information Discovery and Exploitation) at LIG

Social Content Sites

- **Web destinations that let users:**
  - Consume and produce content
    - Videos / photos / articles /…
    - tags / ratings / reviews /…
  - Engage in social activities with
    - friends / family / colleagues / acquaintances /…
    - people with similar interests / located in the same area /…
- **Two major driving factors:**
  - Social activities improve the attractiveness of traditional content sites
    - the "similar traveler" feature improves user engagement
  - Content is critical to the value of social networking sites
    - a significant amount of user time is spent browsing other people's photos, posts, etc.

Social Content Sites

- **Users engage the system**
  - Contribute content
  - Disclose information about themselves
  - Need help navigating the ever-growing cyber-city maze
- **Ultimate goal**
  - Personalize search and information discovery
  - Predict what a user's interests will be in the future
  - Understand user behavior
- **Many social content sites, collaborative tagging sites are one particular kind**
  - *Flickr*, *YouTube*, *Delicious*, photo tagging in *Facebook*

Course Outline

- Nov 9th, 2016: Recommendation
- Nov 15th, 2016: Social data mining

Recommendation Outline

- Recommender Systems
  - **What are recommender systems** and how do they work?
  - Example application: Hotlist Recommendation on Delicious
  - How are recommender systems evaluated?

- Recommendation challenges
  - Well-known challenges
  - Recommendation diversity
  - Group recommendation

Recommender Systems

GetGlue

digg

YouTube

ebay

amazon.co

NETFLIX

last.fm

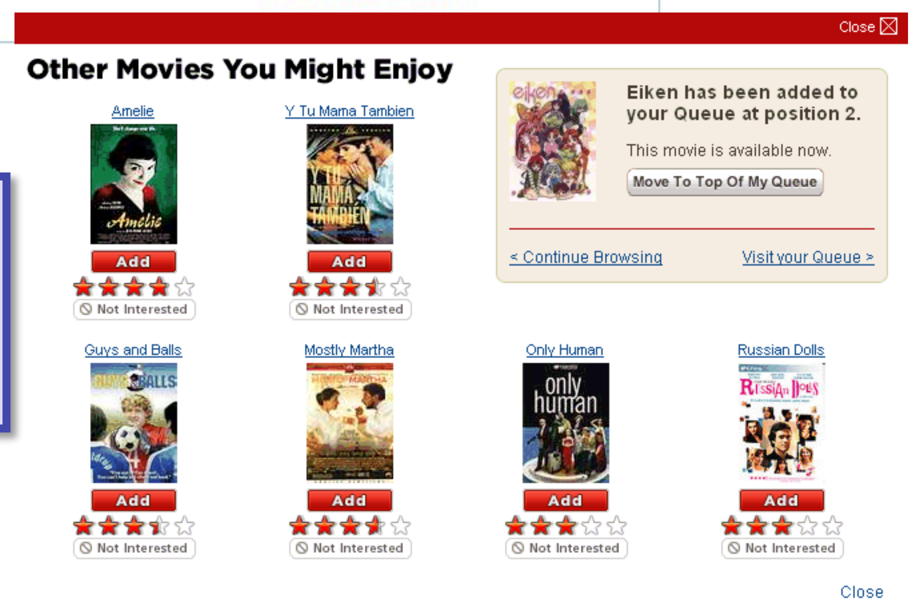
news

hulu

BARNES&NOBLE

All are social content sites that thrive on
User Generated Content (UGC)!

Recommender System



- *Predict ratings for unrated items*
- *Recommend top-k items*

Motivation

- from http://blog.kiwitobes.com/?p=58
- Amazon makes 20-30% of its sales from recommendations. Only 16% of people go to Amazon with explicit intent to buy something
- Collected data matters more than the algorithm.
  – Amazon's algorithm is essentially a large product-product correlation matrix for the past hour, but it works for them because they collect so much data through user actions
- A lot of types of data can be used: votes, ratings, clicks, page-view time, purchases, tagging…

Academia: An Overview

- **Early days: 3 papers by HCI researchers (1995)**
- **Today: over 1000 papers**
  - ACM RecSys09
    - 203 submissions, thereof 140 long and 63 short papers
    - acceptance rate for long papers of 17% and of 34% overall
  - Fields: CS/IS, marketing, DM/statistics, MS/OR
- **Netflix $1M Prize Competition**
  - Data: ≈18K movies, ≈500K customers, 100M ratings
  - $1M Prize: improve Netflix RMSE rates by 10%
  - ≈ 40K contestants from 179 countries
  - Winners in June 2009: a coalition of four: BellKor's Pragmatic Chaos with statisticians, machine learning experts and computer engineers from America, Austria, Canada and Israel — declared that it had produced a program that improves the accuracy of the predictions by 10.05 percent.
- **2nd Netflix Workshop was at KDD in August 2008.**

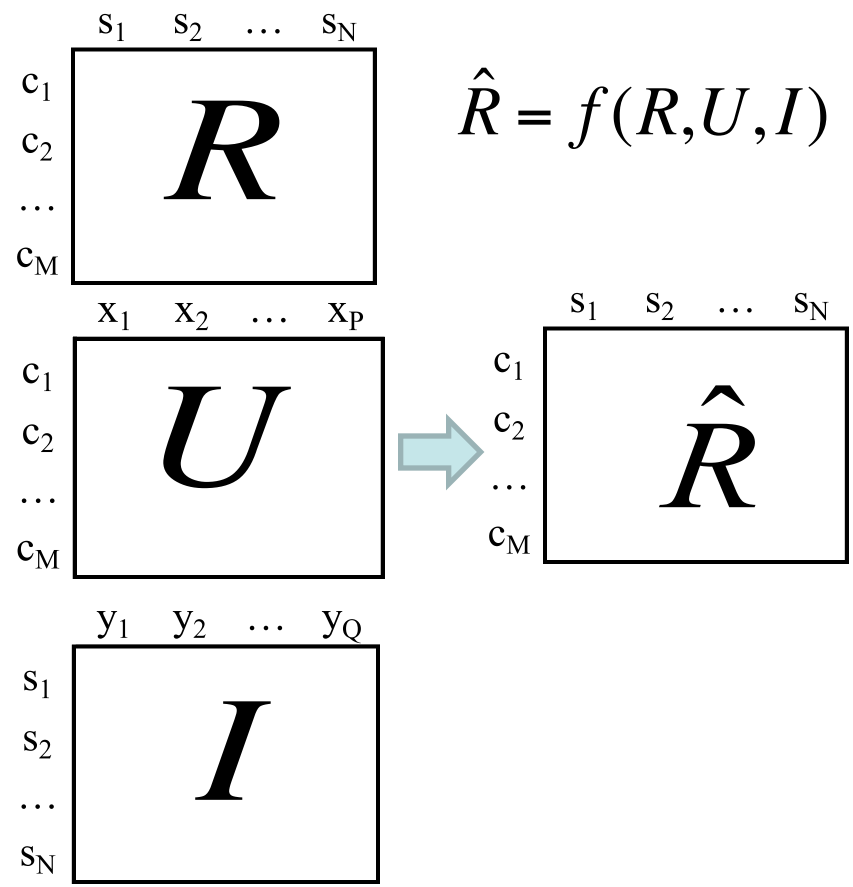
Recommendation Outline

- Recommender Systems
  - What are recommender systems and **how do they work**?
  - Example application: Hotlist Recommendation on Delicious
  - How are recommender systems evaluated?

- Recommendation challenges
  - Well-known challenges
  - Recommendation diversity
  - Group recommendation

Recommendation Model

- **Input**
  - Rating matrix *R:* r_{ij} – rating user c_i assigns to item s_j
  - User attribute matrix *U:* x_{ij} – *attribute* x_j *of user* c_i
  - Item attribute matrix *I:* y_{ij} – *attribute* y_j *of item* s_i
- **Output**
  - Predicted new matrix $\hat{R}$



$$\hat{R} = f(R, U, I)$$

Types of Recommendations

- **Content-based**
  - How similar is an item *i* to items *u* has liked in the past?
  - Uses metadata for measuring similarity
  - Works even when no ratings are available on items
  - Requires metadata!

- **Collaborative filtering**
  - Treat items and users as vectors, compute vector distance

Taxonomy of Traditional Recommendation Methods

- Recommendation approach [Balabanovic & Shoham 1997]
  - Content-based, collaborative filtering
- Nature of the prediction technique
  - Heuristic-based (uses matrix as is), model-based
- Support for rating/transaction data
  - Both, rating-only [R], transaction-only [T]

	Heuristic-based	Model-based
Content-based		
Collaborative filtering		

Content-based, Heuristic-based

- Item similarity methods [Lang 1995; Pazzani & Billsus, 1997; Zhang et al. 2002]
  - Information Retrieval (IR) Techniques
  - Treat each item as a document
  - Item similarity computed as document similarity

- Instance-based learning [Schwab et al. 2000]
- Case-based reasoning [Smyth 2007]

	Heuristic-based	Model-based
Content-based		
Collaborative filtering		

Term Frequency

Variants of TF weight

weighting scheme	TF weight
binary	$0, 1$
raw frequency	$f_{t,d}$
log normalization	$1 + \log(f_{t,d})$
double normalization 0.5	$0.5 + 0.5 \cdot \frac{f_{t,d}}{\max_{\{t' \in d\}} f_{t',d}}$
double normalization K	$K + (1 - K) \frac{f_{t,d}}{\max_{\{t' \in d\}} f_{t',d}}$

Inverse Document Frequency

Variants of IDF weight

weighting scheme	IDF weight ($n_t = \|\{d \in D : t \in d\}\|$)
unary	1
inverse document frequency	$\log \frac{N}{n_t}$
inverse document frequency smooth	$\log(1 + \frac{N}{n_t})$
inverse document frequency max	$\log\left(1 + \frac{\max_{\{t' \in d\}} n_{t'}}{n_t}\right)$
probabilistic inverse document frequency	$\log \frac{N - n_t}{n_t}$

Item Similarity based on IR

- Item attributes are word occurrences in each document

$$y_{ij} = TF_{ij} \cdot IDF_j$$

- TF_{ij} – term frequency: frequency of word y_j occurring in the description of item s_i;
- IDF_j – inverse document frequency: inverse of the frequency of word y_j occurring in descriptions of all items

- Each item becomes a vector of y_{ij}

Item Similarity

- Content-based profile v_i of user c_i constructed by aggregating profiles of items c_i has experienced

$$\hat{r}_{ij} = score(\mathbf{v}_i, \mathbf{y}_j)$$

$$\hat{r}_{ij} = \cos(\mathbf{v}_i, \mathbf{y}_j) = \frac{\mathbf{v}_i \bullet \mathbf{y}_j}{\| \mathbf{v}_i \|_2 \cdot \| \mathbf{y}_j \|_2}$$

Content-based, Model-based

- Classification models [Pazzani & Billsus 1997; Mooney & Roy 1998]
- One-class Naïve Bayes classifier [Schwab et al. 2000]
- Latent-class generative models [Zhang et al. 2002]

	Heuristic-based	Model-based
Content-based		
Collaborative filtering		

Collaborative Filtering Algorithms

- Non-Personalized Summary Statistics
- K-Nearest Neighbor
- Dimensionality Reduction
- Content + Collaborative Filtering
- Graph Techniques
- Clustering
- Classifier Learning

	Heuristic-based	Model-based
Content-based		
Collaborative filtering		
Hybrid		

Collaborative Filtering, Heuristic-based

- **Neighborhood methods**
  - User-based algorithm [Breese et al. 1998; Resnick et al. 1994; Sarwar et al. 1998]
  - Item-based algorithm [Deshpande & Karypis 2004; Linden et al. 2003; Sarwar et al. 2001]
  - Similarity fusion [Wang et al. 2006]
  - Weighted-majority [Delgado and Ishii 1999]
  - Matrix reduction methods (SVD, PCA processing) [Goldberg et al. 2001; Sarwar et al. 2000]
- **Association rule mining [Lin et al. 2002]**
- **Graph-based methods [Aggarwal et al. 1999; Huang et al. 2004, 2007]**

	Heuristic-based	Model-based
Content-based		
Collaborative filtering		
Hybrid		

Collaborative Filtering, Heuristic-based

	HP1	HP2	HP3	TW	SW1	SW2	SW3
A	4			5	1		
B	5	5	4				
C				2	4	5	
D		3					3

Jaccard

	HP1	HP2	HP3	TW	SW1	SW2	SW3
A	4			5	1		
B	5	5	4				
C				2	4	5	
D		3					3

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

Jaccard(A,B) = 1/5 < 2/4 = Jaccard(A,C)

Cosine

	HP1	HP2	HP3	TW	SW1	SW2	SW3
A	4			5	1		
B	5	5	4				
C				2	4	5	
D		3					3

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}}$$

, where A_i and B_i are components of vector A and B respectively.

cos(A,B) = 0.380 > 0.322 = cos(A,C)

Rounding the data

	HP1	HP2	HP3	TW	SW1	SW2	SW3
A	4			5	1		
B	5	5	4				
C				2	4	5	
D		3					3

Replace ratings 3, 4, 5, with 1
And ratings 1, 2, with 0

Compute Jaccard and Cosine

Shows that C is further from A than B is

Normalizing ratings

	HP1	HP2	HP3	TW	SW1	SW2	SW3
A	2/3			5/3	-7/3		
B	1/3	1/3	-2/3				
C				-5/3	1/3	4/3	
D		0					0

Replace each rating with its difference with the mean (average) for that user
Low ratings become negative
High ratings are positive

Cosine: users with opposite views on common movies will have vectors in opposite directions and users with similar opinions aboutmovies rated in common will have a small angle.

cos(A,B) = 0.092 > -0.559 = cos(A,C)

Collaborative Filtering, Model-based

- Matrix reduction methods [Takacs et al. 2008; Toscher et al. 2008]
- Latent-class generative model [Hofmann 2004; Kumar et al. 2001; Jin et al. 2006]
- User-profile generative model [Pennock et al. 2000; Yu et al. 2004]
- User-based classifiers [Billsus & Pazzani 1999; Pazzani & Billsus 1997]
- Item dependency (Bayesian) networks [Breese et al. 1998; Heckerman et al. 2000]

	Heuristic-based	Model-based
Content-based		
Collaborative filtering		
Hybrid		

Recommendation Outline

- Recommender Systems
  - What are recommender systems and how do they work?
  - **Example application: Hotlist Recommendation on Delicious**
  - How are recommender systems evaluated?

- Recommendation challenges
  - Well-known challenges
  - Recommendation diversity
  - Group recommendation

Recommendation Outline

- Recommender Systems
  - What are recommender systems and how do they work?
  - Example application: Hotlist Recommendation on Delicious
  - **How are recommender systems evaluated?**

- Recommendation challenges
  - Well-known challenges
  - Recommendation diversity
  - Group recommendation

Recommendation Outline

- Recommender Systems
  - What are recommender systems and how do they work?
  - Example application: Hotlist Recommendation on Delicious
  - How are recommender systems evaluated?

- **(Some) Recommendation challenges**
  - Well-known challenges
  - Recommendation diversity
  - Group recommendation

Well-Known Challenges

- The new user problem
- The recurring startup problem
- The sparse rating problem
- The scaling problem

The New User Problem

- To be able to make accurate predictions, the system must first learn the user’s preferences from the input the user provides (e.g., movie ratings, URL tagging).
- If the system does not show quick progress, a user may lose patience and stop using the system

The Recurring Startup Problem

- New items are added regularly to recommender systems.
- A system that relies solely on users' preferences to make predictions would not be able to make accurate predictions on these items.
- This problem is particularly severe with systems that receive new items regularly, such as an online news article recommendation system.

The Sparse Rating Problem

- In any recommender system, the number of ratings already obtained is very small compared to the number of ratings that need to be predicted.
- Effective generalization from a small number of examples is thus important.
- This problem is particularly severe during the startup phase of the system when the number of users is small.

The Scaling Problem

- Recommender systems are normally implemented as a centralized algorithm and may be used by a very large number of users.
- Sometimes, predictions need to be made in real time and many predictions may potentially be requested at the same time.
- The computational complexity of the algorithms needs to scale well with the number of users and items in the system.

Recommendation Outline

- Recommender Systems
  - What are recommender systems and how do they work?
  - Example application: Hotlist Recommendation on Delicious
  - How are recommender systems evaluated?

- Recommendation challenges
  - Well-known challenges
  - **Recommendation diversity**
  - Group recommendation

Diversification

From the pool of relevant items, identify a list of items that are dissimilar to each other and maintain a high cumulative relevance, i.e., strike a good balance between relevance and diversity.

Existing Solutions

- **Attribute-based diversification in 3 steps:**
  - pair-wise item-to-item distance function on item attributes
  - Perform Diversification:
    - Optimize an overall score as a weighted combination of relevance and distance
    - Constrain either relevance or distance, maximizing the other
  - Overhead of retrieving item attributes

- **Explanation-Based Diversification**

Recommendation Strategy

- **Estimate the rating of an unrated item (*i*) by the user (*u*) based on its similarity to items already rated and how *u* rated those items.**

$$\texttt{relevance}(u, i) = \Sigma_{i' \in \mathcal{I}} \texttt{ItemSim}(i, i') \times \texttt{rating}(u, i')$$

- **Similarly, one could define a user-based strategy**

$$\texttt{relevance}(u, i) = \Sigma_{u' \in \mathcal{U}} \texttt{UserSim}(u, u') \times \texttt{rating}(u', i)$$

Explanation

- **Basic Notion**
  - The set of objects because of which a particular item is recommended to the user

- **Explanation for Item-Based Strategies**

$$\mathtt{Expl}(u,i) = \{i' \in \mathcal{I} \mid \mathtt{ItemSim}(i,i') > 0 \; \& \; i' \in \mathtt{Items}(u)\}$$

- **Explanation for User-Based Strategies**

$$\mathtt{Expl}(u,i) = \{u' \in \mathcal{U} \mid \mathtt{UserSim}(u,u') > 0 \; \& \; i \in \mathtt{Items}(u')\}$$

Explanation-Based Diversity

- **Pair-wise diversity distance between two recommended items**
  - Standard similarity measures like *Jaccard similarity* and *cosine similarity*
  - E.g. (Distance based on Jaccard similarity)

$$DD_u^J(i, i') = 1 - \frac{|\mathtt{Expl}(u,i) \cap \mathtt{Expl}(u,i')|}{|\mathtt{Expl}(u,i) \cup \mathtt{Expl}(u,i')|}.$$

- **Diversity for the set of recommended items (*S*)**

$$DD_u(S) = avg\{DD_u(i, i') \mid i, i' \in S\}$$

Diverse Recommendation Problem

Top-K Recommendation with Diversification

Given a user u, find a subset S from the set of candidate items, such that |S| = k and the overall relevance of items in S and the diversity of S are balanced.

Cong Yu, Laks V. S. Lakshmanan, Sihem Amer-Yahia: Recommendation Diversification Using Explanations. ICDE 2009: 1299-1302

Recommendation Outline

- Recommender Systems
  - What are recommender systems and how do they work?
  - Example application: Hotlist Recommendation on Delicious
  - How are recommender systems evaluated?

- Recommendation challenges
  - Well-known challenges
  - Recommendation diversity
  - **Group recommendation**

Group Recommendation (motivation)

- How do you decide where to go to dinner with friends?
  - email/text/phone
  - not optimal for reaching consensus
- What if there was a system that knew each user's preferred list?
- What is the best way to model consensus?
- How to *evaluate* that?
- How to *efficiently* compute *group recommendations*?

Group Recommendation by Example

- **Task: recommend a movie to group G ={u1, u2 ,u3}**
  - predictedRating(u1,"God Father") = 5
  - predictedRating(u2, "God Father") = 1
  - predictedRating(u3, "God Father") = 1

  - predictedRating(u1, "Roman Holiday") = 3
  - predictedRating(u2, "Roman Holiday") = 3
  - predictedRating(u3, "Roman Holiday") = 1

- ***Average Rating*** **and** ***Least Misery*** **fail to distinguish between "God Father" and "Roman Holiday"**

Group Reco Problem Definition

Consensus function *combines* ***relevance*** *(average or least misery) and* ***disagreement*** *(average pair-wise or variance) in the score of a group recommendation*

$\mathcal{F}(\mathcal{G}, i) = w_1 \times \texttt{rel}(\mathcal{G}, i) + w_2 \times (1 - \texttt{dis}(\mathcal{G}, i))$, *where* $w_1 + w_2 = 1.0$ *and each specifies the relative importance of relevance and disagreement in the overall recommendation score.*

Problem: Given a user group G (formed on-the-fly) and a consensus function F, find the k best items according to F, such that each item is new to all users in G

S. Amer-Yahia, S. B. Roy, A. Chawla, G. Das, C. Yu: Group Recommendation: Semantics and Efficiency. VLDB 2009.

In practice

- Choose your similarity measure wisely, you will have to try more than one
- Define your goal early with the domain expert to determine how to evaluate your approach
- Build a prototype ASAP
- Use existing tools whenever possible

Main references

- Overview of Recommendation Systems

http://web.stanford.edu/class/ee378b/papers/adomavicius-recsys.pdf

- Collaborative Filtering: Chapter 9 of Mining Massive Datasets book

http://infolab.stanford.edu/~ullman/mmds/book.pdf

- Delicious recommendations

J. Stoyanovich, S. Amer-Yahia, C. Yu, C. Marlow: Leveraging Tagging Behavior to Model Users' Interest in del.icio.us (AAAI Workshop on Social Information Processing 2008)

- Diverse recommendations

Cong Yu, Laks V. S. Lakshmanan, Sihem Amer-Yahia: Recommendation Diversification Using Explanations. ICDE 2009: 1299-1302

- Group recommendations

S. Amer-Yahia, S. B. Roy, A. Chawla, G. Das, C. Yu: Group Recommendation: Semantics and Efficiency. ***VLDB 2009.***

- Evaluating recommender systems

http://essay.utwente.nl/59711/1/MA_thesis_J_de_Wit.pdf