

Recommender Systems

Alban Galland

INRIA-Saclay

18 March 2010

- What is this lecture about?
 - ▶ What is the purpose of a recommender system?
 - ▶ What are the key features?
 - ▶ How does it work?
 - ▶ What are the main challenges?
 - ▶ When to use it?
 - ▶ How to design it?

Content

- 1 Who uses a recommender system?
- 2 What tasks and data correspond to a recommendation problem?
- 3 How to do it?
 - Content-filtering algorithms
 - Collaborative-filtering algorithms
 - Not personalized
 - User-based
 - Item-based
 - Hybrid methods
- 4 To go further
 - Interesting issues
 - Bibliography

Content

- 1 Who uses a recommender system?
- 2 What tasks and data correspond to a recommendation problem?
- 3 How to do it?
 - Content-filtering algorithms
 - Collaborative-filtering algorithms
 - Not personalized
 - User-based
 - Item-based
 - Hybrid methods
- 4 To go further
 - Interesting issues
 - Bibliography

Content site

- Examples: AlloCine, Zagat, LibraryThing, Last.fm, Pandora, StumbleUpon
- Task: predict ratings of items by a given user or find a list of interesting items
- Data: precise content description, explicit rating for some user

1. **The Art of Computer Programming: Volume 3 - Sorting and Searching** by **Donald E. Knuth**
448 copies. Average rating 4.54.
No thanks! | Why? (close why) Recommendation based on:
Introduction to Algorithms by Thomas H. Cormen
Compilers: Principles, Techniques, and Tools by Alfred V. Aho
Computers and Intractability: A Guide to the Theory of NP-Completeness by Michael R. Garey
2. **Introduction to Automata Theory, Languages, and Computation** by **John E. Hopcroft**
206 copies. 2 reviews. Average rating 3.98.
No thanks! | Why? (close why) Recommendation based on:
Computers and Intractability: A Guide to the Theory of NP-Completeness by Michael R. Garey
Computational complexity by Christos H. Papadimitriou

Recommendation on LibraryThing

eCommerce site

- Example: Amazon, Netflix
- Task: build group of products for bundle sales or more generally find a list of products that the user is likely to buy
- Data: list of purchases and browsing history for all users

What Do Customers Buy After Viewing This Item?

77% buy the item you viewed | 11% buy this alternative | 5% buy this alternative

Cell: A Novel Hardcover by Stephen King \$26.95 \$19.14 | Duma Key, A Novel Paperback by Stephen King \$9.99 | Just After Sunset: Stories Hardcover by Stephen King \$28.00 \$18.48

[Find similar items](#)

Frequently Bought Together

When customers buy | 12% buy it with | 10% buy it with

Kiss of a Demon King Kindle Edition by Kresley Cole | Dark Desires After Dusk Kindle Edition by Kresley Cole | Dream Warrior Kindle Edition by Sherrilyn Kenyon

[Find similar items](#)

Recommendation on Amazon

eCommerce site

- The Netflix challenge
 - ▶ \$1M prize competition
 - ▶ Input: huge training dataset
 - ▶ Goal: improve root mean square prediction error rate of 10% compare to Netflix algorithm
 - ▶ 40000+ teams from 186 countries (5000+ teams with valid submissions)
 - ▶ Begins October 2006, winners in June 2009

Advertisement

- Example: Google AdSense, DoubleClick
- Task: find a list of advertisements optimized according to expected income
- Data: browsing history for all users

Sponsored Links

New Car
Search Cheap **Cars** & Save Money
View **New** & Used Local Listings Now!
[GoodCheapCars.com](#)

Drive a New Free Car
Or Get Paid to Drive Your Own **Car**.
No Catch, No Hidden Costs! Aff.
[www.thefreecar.com](#)

Recommendation on Google

Content

- 1 Who uses a recommender system?
- 2 What tasks and data correspond to a recommendation problem?
- 3 How to do it?
 - Content-filtering algorithms
 - Collaborative-filtering algorithms
 - Not personalized
 - User-based
 - Item-based
 - Hybrid methods
- 4 To go further
 - Interesting issues
 - Bibliography

Task(1)

- General purpose
 - ▶ Top-k filtering: list of “best” items (main usage) or anti-spam
 - ▶ Items correlation: find similar items
 - ▶ Prediction of rating: predict affinity between any pair of an user and an item (more general)

What to do with data?

- Two kinds of problem with data:
 - ▶ Information retrieval (IR): static content, dynamic query \Rightarrow modeling content (organized with index)
 - ▶ Information filtering (IF): dynamic content, static query \Rightarrow modeling query (organized as filters)
- Recommendation is between IR and IF since the content varies slowly and the queries depend of few parameters. Methods of both IR and IF are then used to reduce computation at query time.

Task(2)

- Degree of personalization
 - ▶ Generic: everyone receives same recommendations
 - ▶ Demographic: everyone in the same category receives same recommendations
 - ▶ Contextual: recommendation depends only on current activity
 - ▶ Persistent: recommendation depends on long-term interests

Data (1)

- Context of the current page (current request, item currently explored and structured content about this context)
- History of the current user on the system (explicit or implicit ratings)
- History of all users on the system
- History of the current user on multiple systems, the whole web or even on its computer
- History of all users on multiple systems, the whole web or even their computer

Explicit ratings

- Numeric ratings:
 - ▶ Numeric scale, usually between 2 (thumb up/thumb down) and 15 (between A+ and E-) levels.
 - ▶ The more levels you have, the much data you get but the much variance you have on these data.
 - ▶ Numeric ratings should be normalized.
- Partial order: comparison between two items
- Semantic information: tags, labels

Data (2)

- In general, three matrix as input:
 - ▶ Users attributes
 - ▶ Items attributes
 - ▶ Rating matrix

Implicit ratings

- Based on interaction and time
 - ▶ purchase
 - ▶ clicks
 - ▶ browsing (page view time)
 - ▶ cursor on the page
- Used to generate an implicit numeric rating

Content

- 1 Who uses a recommender system?
- 2 What tasks and data correspond to a recommendation problem?
- 3 How to do it?
 - Content-filtering algorithms
 - Collaborative-filtering algorithms
 - Not personalized
 - User-based
 - Item-based
 - Hybrid methods
- 4 To go further
 - Interesting issues
 - Bibliography

Content-filtering algorithms

- Usually, content-filtering algorithms means an algorithm based on the attributes of the items and the ratings of the targeted user
- Interpretation of the preferences of users as a function of the attributes
- Two main methods:
 - ▶ Heuristic-based: Use common techniques of information retrieval presented earlier in the course : TF/IDF, cosine, clustering...
 - ▶ Model-based: Use a probabilistic model to learn prediction of users from attributes

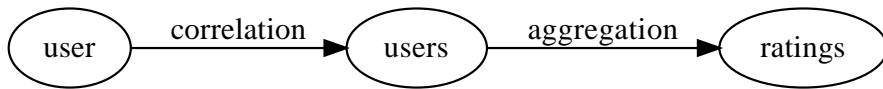
General scope

- Purely editorial (still used for some advertisement)
- Content filtering: depending on attributes of items
- Collaborative filtering: depending on ratings of all users
- Hybrid

Direct aggregation

- Usually, collaborative filtering algorithm means an algorithm based on the rating matrix.
- The recommender system displays some statistics summary
 - ▶ the average rating of the users
 - ▶ the average rating of professional reviewers.
 - ▶ a set of reviews of the users or of professional reviewer
- Some basic techniques such as explicit voting or date are used to rank reviews.

User-based collaborative filtering



- For each user u_i , compute correlation with others users
- For each item i_k , aggregate the ratings of i_k by the users highly correlated with u_i
- Problem: sparsity of data (little information about each user) \Rightarrow bad correlation, easy to attack (cf. cold start and attacks issues)

Some aggregations methods

- Let $\hat{r}(u_i, i_k)$ the rating prediction of user u_i and item i_k
- Let $S_t(u_i) = \{u_j, sim(u_i, u_j) > t\}$ the users highly correlated with u_i for a threshold t
 - ▶ Means on the best users

$$\hat{r}(u_i, i_k) = \frac{1}{|S_t(u_i)|} \sum_{S_t(u_i)} r(u_j, i_k)$$

- ▶ Weighted average on the bests users

$$\hat{r}(u_i, i_k) = \frac{\sum_{S_t(u_i)} sim(u_i, u_j) r(u_j, i_k)}{\sum_{S_t(u_i)} sim(u_i, u_j)}$$

- Usually, choice of $S_t(u_i)$ is sensitive since it is a trade-off between sparsity and noise.

Some correlation methods

- Let U_i be the vector of ratings of user u_i (see as a line).

- ▶ Scalar product similarity:

$$sim(u_i, u_j) = U_i^t U_j$$

- ▶ Cosine similarity:

$$sim(u_i, u_j) = \frac{U_i^t U_j}{\|U_i\| \|U_j\|}$$

- ▶ Another one:

$$sim(u_i, u_j) = \frac{U_i^t U_j}{\|U_i\|^2}$$

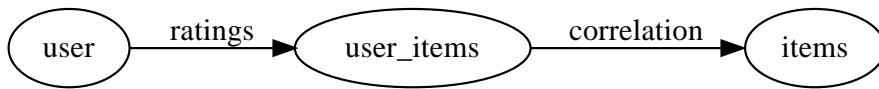
- Usually, U_i has to be normalized to get meaningful results

Application on an example

- What would you predict for user1 on item5, item6 and item7?

user	item1	item2	item3	item4	item5	item6	item7
user1	5	3	4	1	?	?	?
user2	5	3	4	1	5	2	5
user3	5	?	4	1	5	3	?
user4	1	3	2	5	1	4	2
user5	4	?	4	4	4	?	4

Item-based collaborative filtering



- For each item i_k , compute correlation with others items
- For each user u_i , aggregate her ratings of the items highly correlated with i_k
- For items, sparsity of data is less important \Rightarrow less problems with cold start and attacks

Matrix representation

$$R' = R^t R R \quad (1)$$

- R' is the normalized predicted rating matrix.
- R is the normalized rating matrix where unknown values have been set to 0
- This computation correspond to both user-based and item-based collaborative filtering with scalar product correlation using all intermediate seeds.
- One could use classic matrix dimensionality reduction such as singular value decomposition to decrease the computational cost and improve results.

Application on an example

- What would you predict for user1 on item5, item6 and item7?

user	item1	item2	item3	item4	item5	item6	item7
user1	5	3	4	1	?	?	?
user2	5	3	4	1	5	2	5
user3	5	?	4	1	5	3	?
user4	1	3	2	5	1	4	2
user5	4	?	4	4	4	?	4

Model-based techniques (out of the scope of this course)

- Learn the ratings using a probabilistic model of generation (e.g. Latent-class generative model) and estimation of the parameters (e.g. using Expectation Maximization)

Hybrid methods

- Usually, hybrid algorithms use both items attributes and the ratings of all users
- Generals methods
 - ▶ Heuristic combination of content-filtering and collaborative-filtering methods
 - ▶ For the model-based techniques, modification of the model to take into account both kinds of data. (E.g. Hierarchical Bayesian model of users and items heterogeneity and estimation via Markov Chain Monte Carlo)

Data Quality

- How to manage the cold start problem (new user, new item) or more generally data sparsity?
 - ▶ The system must have a special behavior for user with few ratings (eg. not personalized recommendation)
 - ▶ The system may use bot-users to rate new items according to the content

Content

- 1 Who uses a recommender system?
- 2 What tasks and data correspond to a recommendation problem?
- 3 How to do it?
 - Content-filtering algorithms
 - Collaborative-filtering algorithms
 - Not personalized
 - User-based
 - Item-based
 - Hybrid methods
- 4 To go further
 - Interesting issues
 - Bibliography

Confidence and display(1)

- How to improve the confidence in the recommender system?
 - ▶ By providing good recommendations!
 - ▶ By providing information about each recommendation (eg. ratings, explanation)

Confidence and display(2)

- How to display recommendations?
 - ▶ The item recommended must be easy to identify and evaluate by the user
 - ▶ Ratings must be easy to understand and meaningful
 - ▶ Explanations must provide a quick way for the user to evaluate the recommendation

Interaction and time(2)

- How to manage scalability
 - ▶ Applications usually need real-time prediction computation
 - ▶ The computation time has to scale with number of users and items
- How to manage temporal changes?
 - ▶ You can not run your algorithms each time a modification occurs
 - ▶ The off-line computation must be robust to small modification and scheduled accordingly
 - ▶ The on-line computation must benefit from modifications
 - ▶ The computation must be done incrementally when possible
 - ▶ The system may “forget” older information

Interaction and time(1)

- How to interact with the user?
 - ▶ You may ask the user to correct a prediction
 - ▶ You must update your rating matrix with this prediction and update your recommendation accordingly
 - ▶ You may want to learn the key parameters of your algorithm using the feedback
 - ▶ You may ask the user to provide feedback on the explanation
 - ▶ You may ask the user to provide more context for the current task (eg. by using categories)

Data and security(1)

- How to insure privacy?
 - ▶ If the profile is public, there is no privacy issues
 - ▶ If the profile is private, the system should avoid to give too much information using anonymity techniques.
 - ▶ This problem is even worse in cross-systems

Data and security(2)

- How to design algorithms that are robust against manipulation?
 - ▶ Attacks are characterized by number of false users and knowledge on the system.
 - ▶ The attacker want to modify the distribution of the ratings without being easy to detect
 - ▶ There is a lot of known attacks such as sampling attack, random attack, average attack, bandwagon attack...
 - ▶ Lot of techniques to detect attack : find profiles which are unlikely according to the global distribution of profiles, find profiles updates which are unlikely according to the global distribution of updates...

Improving recommendation(1)

- How to manage diversity?
 - ▶ Recommending very close items could be counter-productive (since they may be substitute) \Rightarrow Systems can use correlation between items (eg. base on content) to filter items
 - ▶ Recommending what everybody like and what the user already know is not really interesting \Rightarrow Systems can try more risky prediction (eg. high score with low confidence)

Improving recommendation(2)

- How to use social networks to improve recommendations?
 - ▶ Users are likely to like what their friends like.
 - ▶ Exploring the social graph is a direct way to do recommendation
 - ▶ Correlation between user could be biased by the social graph
 - ▶ Potential friends could be suggested using recommendation techniques.



Improving recommendation(3)

- How to recommend for a group?
 - ▶ The recommendation for the group could be an aggregation of the recommendation for the members.
 - ▶ The group could be seen as a user (with aggregation functions to reconcile ratings)

Improving recommendation(4)

- How to evaluate recommendation?
 - ▶ There is a lot of noise on the data, which could be the main source of errors
 - ▶ It is more difficult to evaluate when there is no rating.
 - ▶ It is even more difficult if you want to improve recommendation by adding constraints like diversity

Bibliography

-  Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., and Riedl, J. GroupLens: An Open Architecture for Collaborative Filtering of Netnews
CSCW, 1994
-  Adomavicius, G. and Tuzhilin, A. Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions
IEEE Transactions on knowledge and data engineering, 2005