Recommender Systems and Graph Theory Applications

By Vela Dimitrova Mineva

"The Web, they say is leaving the era of search and entering one of discovery. What's the difference? Search is what you do when you are looking for something. Discovery is when something wonderful that you didn't know existed, or didn't know how to ask for, finds you."

M. O'Brien, "The Race to Create a 'Smart' Google"

Outline

1. What is a recommendation system (RS)?
2. History overview
3. Preliminaries
4. Common approaches in RS
   - Collaborative filtering (CF)
   - Content-based filtering
   - Hybrid Approaches
5. Graph Theory Applications in the RS domain
   - Solution to the cold-start & sparsity problems
   - Graph-based RS at eBay
6. Implementing a graph-based RS
   - GraphLab
   - Neo4j & Gremlin
7. Conclusion

Recommendation Systems

<u>WHAT?</u>

- Information filtering technique, which provides users with information, which they might be interested in
- Recommender Systems vs. Search Engines

<u>WHY?</u>

- A necessity for the users in the current age of information overload
- Greater revenues for vendors

Popular Recommender Systems

History Overview

- The concept of RS grows out of the idea of information reuse

- 1992 – Tapestry (manual, CF)
- 1994 – Grouplens (rating data)
- 1997 – MovieLens (movie RS)

After 2000 – Everywhere!

- 2006 – Netflix Prize (best CF algorithm)

Preliminaries

Required components:

- User profile
- Utility matrix (user-item pairs → value)

	Item 1	Item 2	Item 3	Item 4	Item 5	Item 6	Item 7
Alice	4			5	1		
Lily	5	5	4				
Chris				2	4	5	
Yordan		3					3

Goal: Fill in the blanks

Techniques: Collaborative Filtering (CF) - 1

Formula:

collect feedback → analyze → recommend

User-based CF:

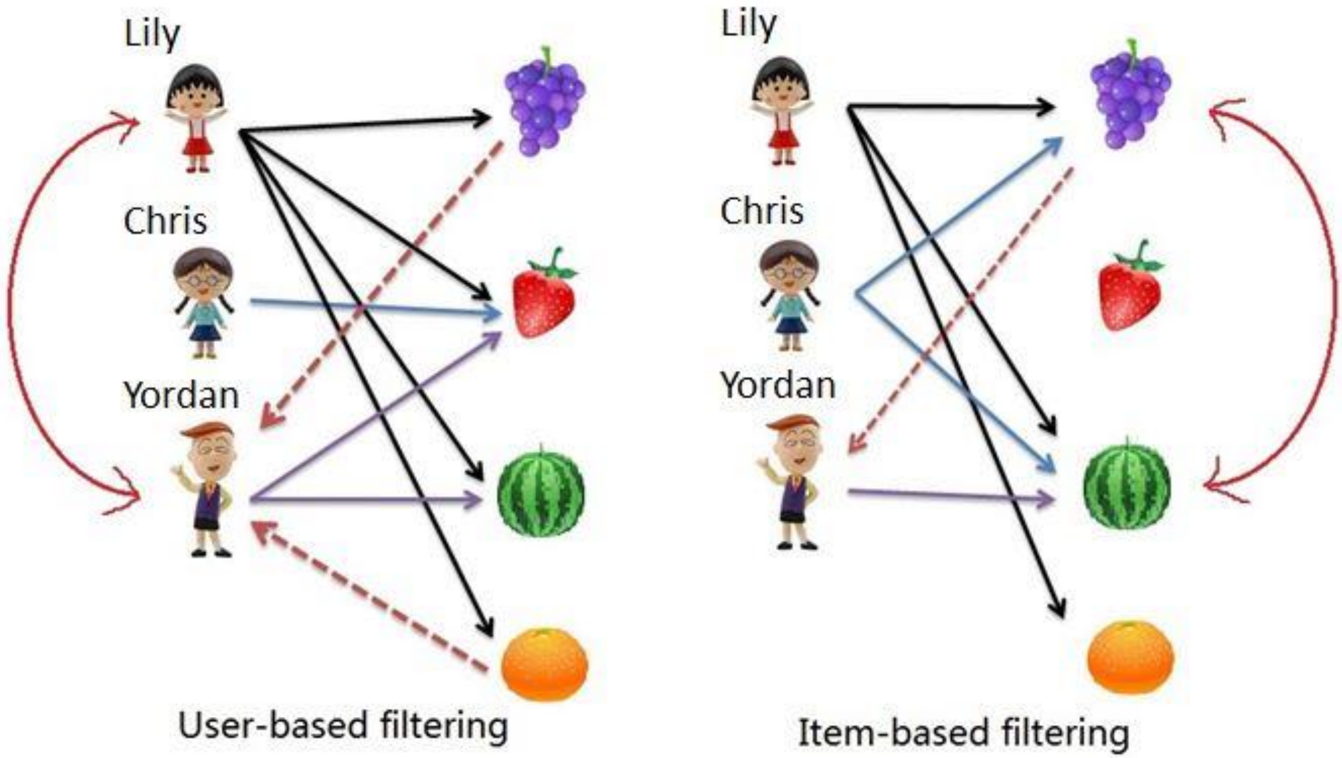
- Use user-item ratings matrix
- Make user-to-user correlations
- Find strongly correlated users
- Recommend items preferred by those users

Techniques: CF (2)

Item-based CF:

- Use user-item ratings matrix
- Make item-to-item correlations
- Find items that are strongly correlated with preferred ones
- Recommend items with strongest correlations

Techniques: CF (3)



Techniques: CF (4)

<u>Advantages:</u>

- Information about the items is not required

<u>Disadvantages:</u>

- Cold-start problem
- Sparsity problem

Techniques: Content-based Filtering

- Use information about the items
- Find items similar to the items preferred in the past
- Recommend items with highest similarity

<u>Advantages:</u>

- Large group of users – NOT required

<u>Disadvantages:</u>

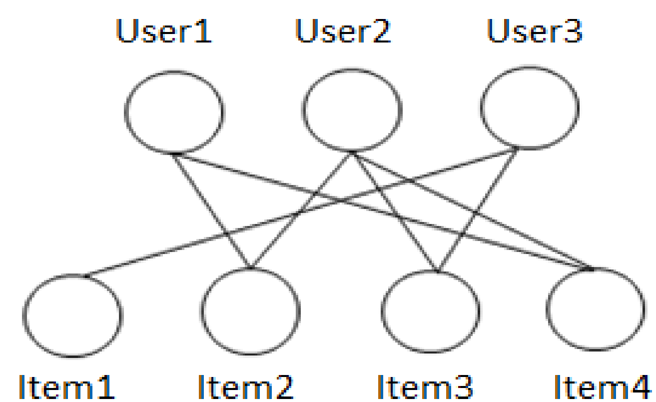
- Information about items – required
- Overspecialization

Cold-Start & Sparsity Problem (1)

<u>Graph-based solution:</u>

- Augment the matrix with new data from "transitivity" of customer tastes
- Spreading activation graph-exploration algorithm

	Item 1	Item 2	Item 3	Item 4
User1	0	1	0	1
User2	0	1	1	1
User3	1	0	1	0



Cold-Start & Sparsity Problem (2)

Pseudocode:

Initialize all activation values $a[i]$ to 0

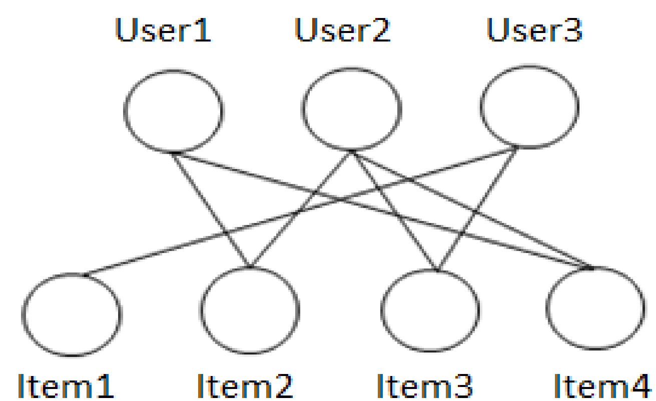
Set one or more origin nodes to an initial activation value greater than F

For each unfired node a_i with $a[i]$> F

  For each edge $a_i a_j$ adjust $a[j] = a[j] + (a[i] * w_{ij} * D)$
  Mark $a[i]$ as a fired node

Example:

- User1 & User2 – nearest neighbors
- Item3 recommended to User1



Graph-based RS at eBay (1)

Thesis:

There is correlation between items users like and dislike.

Architecture of RS:

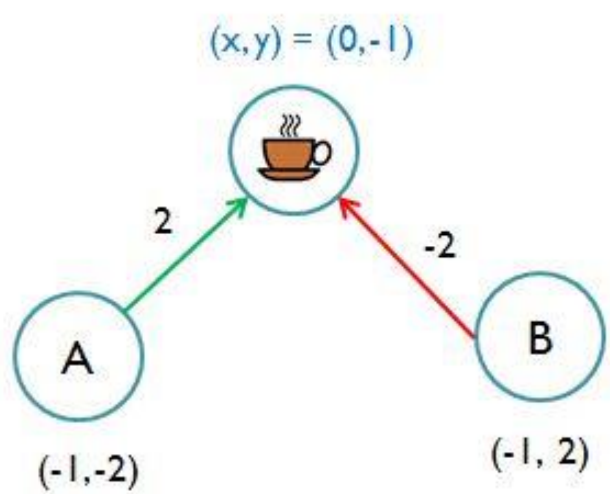
- High-dimensional taste space
- Bipartite directed weighted graph
- Positive weights – likes; negative – dislikes

…

Graph-based RS at eBay (2)

Architecture of RS:

- uv edge weight = dot product of the coordinates of a user u and of an item v
- Location of users and items → similarity/dissimilarity



Assume edge weights:

- 2 = "like"
- -2 = "dislike"

Need to solve the matrix:

$-1 * x - 2 * y = 2$ (edge from A)
$-1 * x + 2 * y = -2$ (edge from B)

Solution = $(0, -1)$

Graph-based Tools (1)

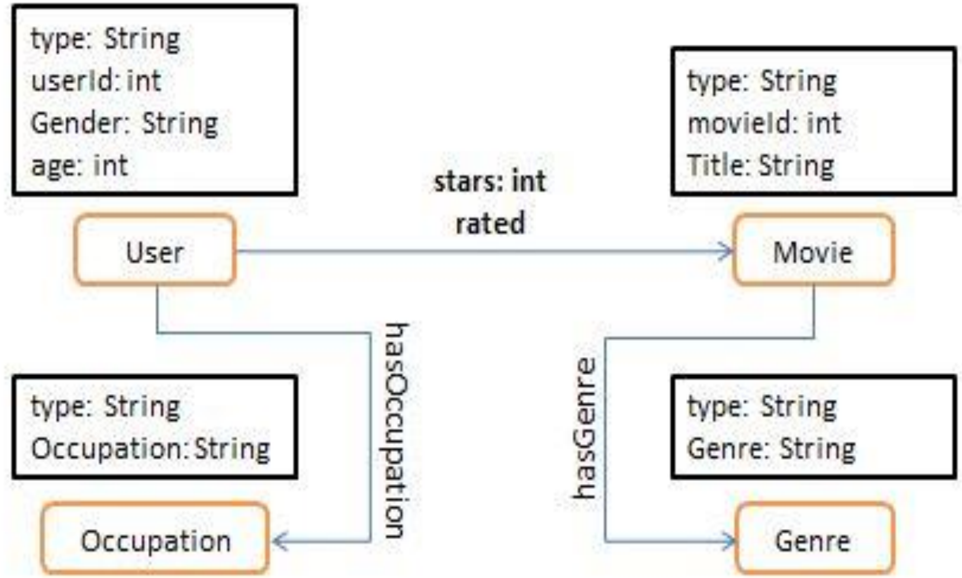
Package for analyzing data in graph analytics, computer vision, etc.

GraphLab

```
# Creating a recommendation system in GraphLab Create
import graphlab
my_recommender = graphlab.recommender.create(data)
recommendations = m.recommend()

#Getting recommendations for a set of users
personalized_recs = my_recommender.recommend(users=my_user)

#Restricting recommendations to a particular set of items
constrained_recs = my_recommender.recommend(items=candidates)

#Excluding previously seen observations
unseen_recs = my_recommender.recommend(exclude=ignore_these)
```

Graph-based Tools (2)

Property Graph



- Edges are directed, $E \subseteq (V \times V)$
- Edges are labeled, $\lambda : E \rightarrow \Sigma$
- Properties are a map from the elements of the graphs $(V \cup E)$ and keys (R) to their values (S), $\mu : (V \cup E) \times R \rightarrow S$

Conclusion

We need recommender systems and graphs are a valuable tool!

Thank You!

Interesting …

Pancake is a new Mozilla Labs project focused on exploring how we search, navigate and discover data on the Web. One of the future "blue sky" ideas is building an extremely powerful content discovery and recommendation engine.

References

Images:

http://www.scottgduncan.com/help-recommenders-write-great-letters-recommendation/

https://blog.mozilla.org/labs/2012/01/pancake-a-new-project-from-mozilla-labs/

http://twobitbard.weebly.com/general-blog/ants-by-jo-vonbargen

http://michaelhsu.tw/2013/06/21/rhadoop-%E5%AF%A6%E4%BD%9C%E6%8E%A8%E8%96%A6%E7%B3%BB%E7%B5%B1/

Other:

http://www.slideshare.net/antiraum/recommender-engines?related=3

Jones, Tim. *Recommender Systems, Part 1: Introduction to Approaches and Algorithms Page 1 of 8 Recommender Systems, Part 1: Introduction to Approaches and Algorithms*. Tech. IBM, 12 Dec. 2013. Web. 10 Dec. 2014.

Konstan, Joseph A. "Introduction to Recommender Systems." University of Minnesota. Coursera. Retrieved from https://class.coursera.org/recsys-001/lecture. Web. 14 Dec. 2014

Lakshmi, Soanpet Sree, and T. Adi Lakshmi. "Recommendation Systems: Issues and challenges.". (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (4), 2014, 5771-5772. Web. 15 December 2014

Melville, Prem, and Vikas Sindhwani. "Recommender systems." *Encyclopedia of machine learning*. Springer US, 2010. 829-838.

O'Brien, Jeffrey M. "The Race to Create a 'smart' Google." *Fortune*. Time Inc., 20 Nov. 2006. Web.15Dec.2014. Retrieved from http://archive.fortune.com/magazines/fortune/fortune_archive/2006/11/27/8394347/index.htm

Anand Rajaraman and Jeffrey David Ullman. 2011. *Mining of Massive Datasets*. Cambridge University Press, New York, NY, USA.

Baluja, Shumeet, et al. "Video suggestion and discovery for youtube: taking random walks through the view graph." *Proceedings of the 17th international conference on World Wide Web*. ACM, 2008.

Bhunje, Sourabh. Collaborative filtering (CF) methodologies. Digital image. *The E Geek*. N.p., 29 May 2014. Web. 15 Dec. 2014.

Dubois, Chris. „Recommender Systems and Text analysis with GraphLab Create". GraphLab Conference. 21 July 2014. San Francisco, CA. Web. 15 Dec. 2014

Dietmar Jannach, Markus Zanker, Alexander Felfernig, and Gerhard Friedrich. 2010. *Recommender Systems: An Introduction* (1st ed.). Cambridge University Press, New York, NY, USA.

Davidson, James, et al. "The YouTube video recommendation system." *Proceedings of the fourth ACM conference on Recommender systems*. ACM, 2010.

Huang, Zan, Hsinchun Chen, and Daniel Zeng. "Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering." *ACM Transactions on Information Systems (TOIS)* 22.1 (2004): 116-142.