



Department of Engineering Management, Information and Systems

EMIS 8331 – Advanced Data Mining

---

# Recommender Systems

## - Content, Collaborative, Hybrid

Scott F Eisenhart

---

# Tutorial Abstract

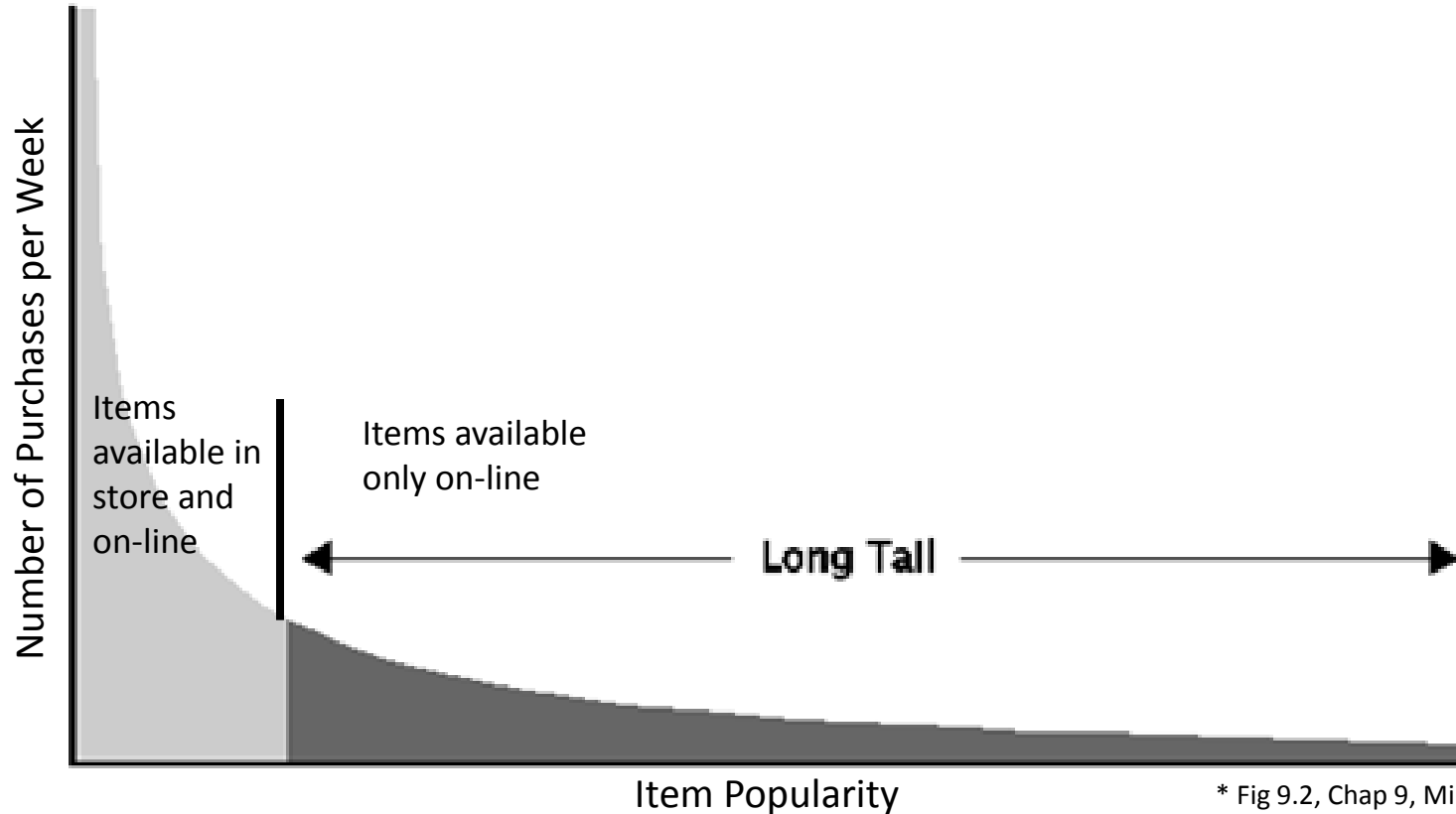
- As the internet store front eclipses and crushes the traditional brick and mortar stores the challenge for on-line retailers and businesses is how to engage with customers and get them to buy products. An e-commerce web site is not constraint by the physical limits on shelf space and can offer a virtually un-limited number of products. Other businesses such as travel sites and on-line news offer a world of choices to their users. The number of choices can be overwhelming. This challenge has given rise to a class of applications that predict user choices and make personalized recommendations of products and services. These systems are called Recommender Systems. This tutorial will introduce the types of recommender systems, describe the architecture, current practices, limitations and future work.

# Introduction

---

- Simple Goal: Recommend items to a user to maximize a utility function
- A recommender system delivers an item or list of personalized items to a user.

# The “Long Tail”



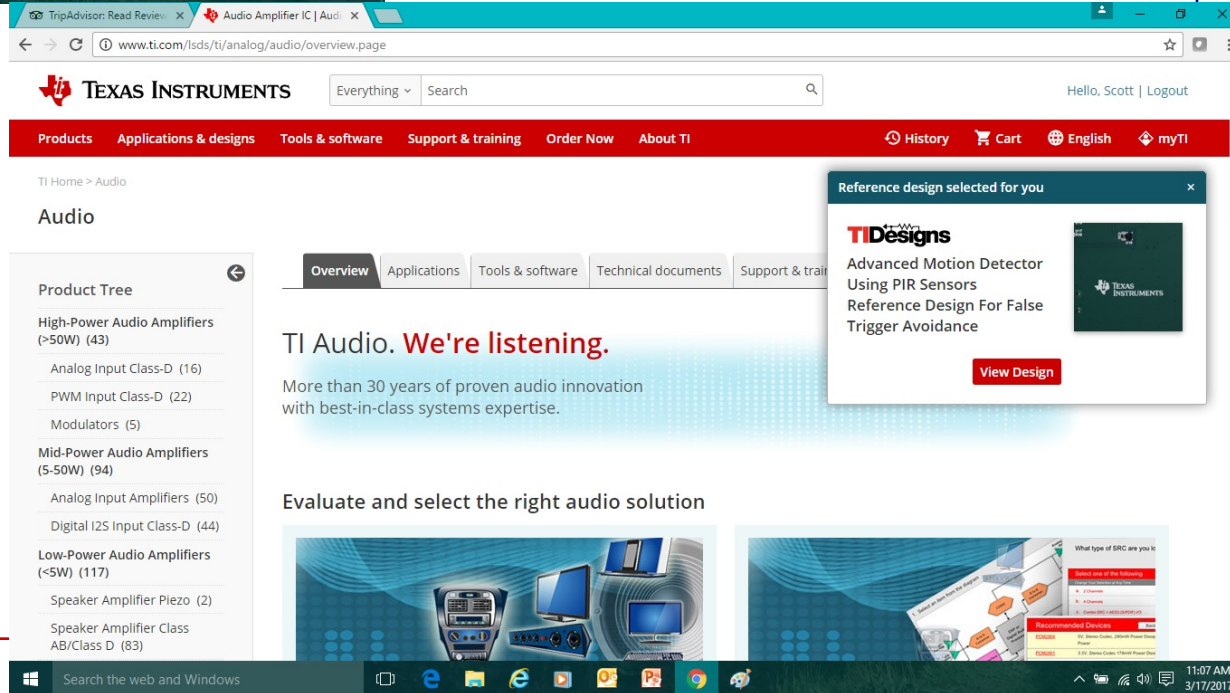
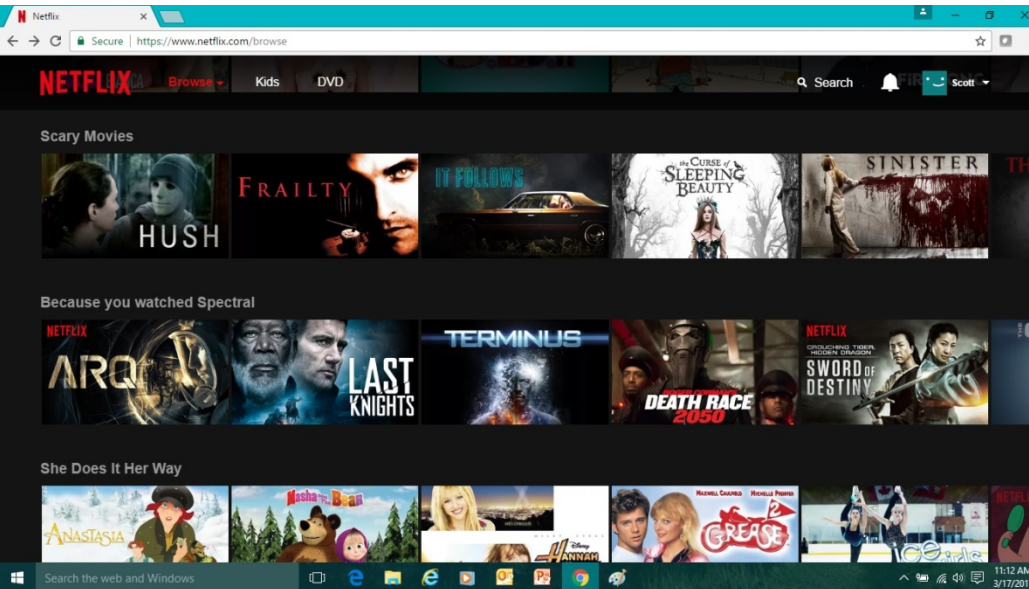
\* Fig 9.2, Chap 9, Mining Massive Data Sets

- Brick and Mortar Stores = Constrained Choice
- On-line Stores = Unlimited Choice

# Applications of Recommenders - 1

Entertainment  
- Movie, music, news, etc

E-Commerce  
- Products  
- Books, PCs, etc



# Applications of Recommenders - 2

---

- Major sites are usually equipped with several recommendation techniques: simple popularity lists to sophisticated.
  - Pages visited on the web
  - Liked items, purchased items
  - Community-based, Social filtering

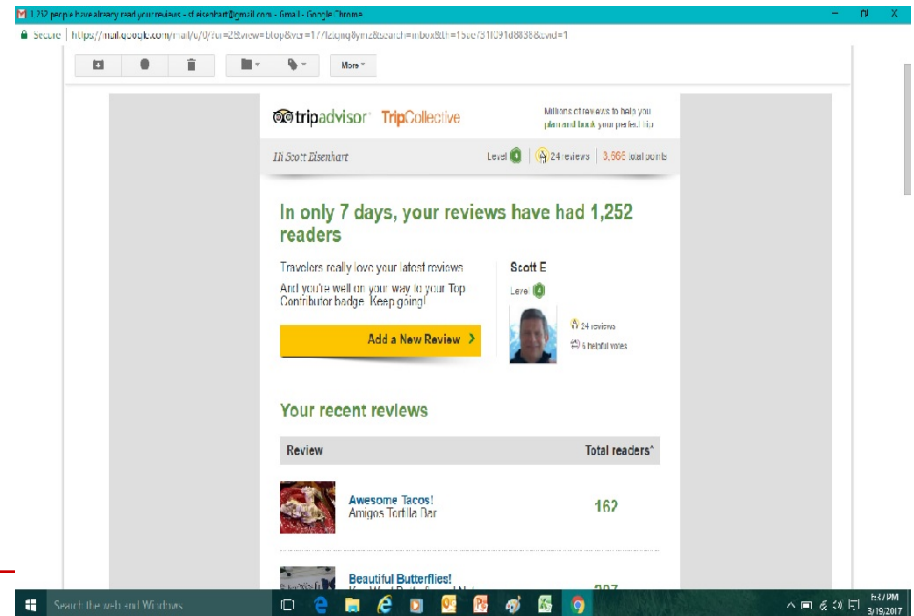
# Benefits of Recommenders

## For Providers

- Increase # of Items Sold
- Sell More Diverse items
- Increase User Satisfaction
- Gather User Preferences

## For Users

- Find Good Items
- Find a Sequence of Items
- More Effective Browsing
- Influence Others
- Help Others

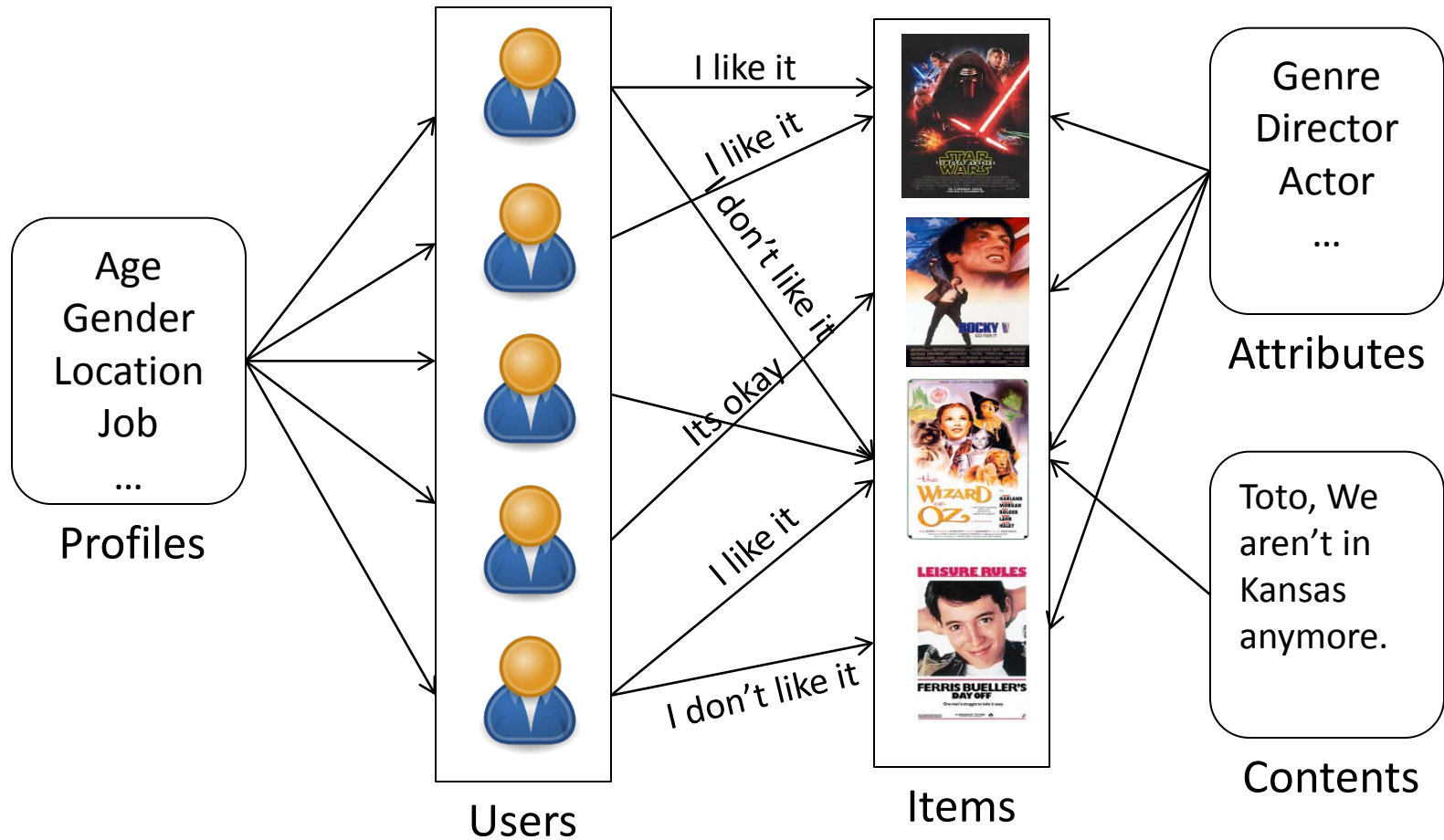


# Recommender Approaches

- Non-personalized & stereo typed
  - Popularity, Group Preference
- Product Association
  - People who bought A also bought B
- Content Based
  - The user will be recommended items similar to the ones the user preferred in the past
- Collaborative
  - The user will be recommended items that people with similar tastes and preferences liked in the past
- Hybrid



# Recommender Systems Architecture



# Recommendation System – Rating Matrix

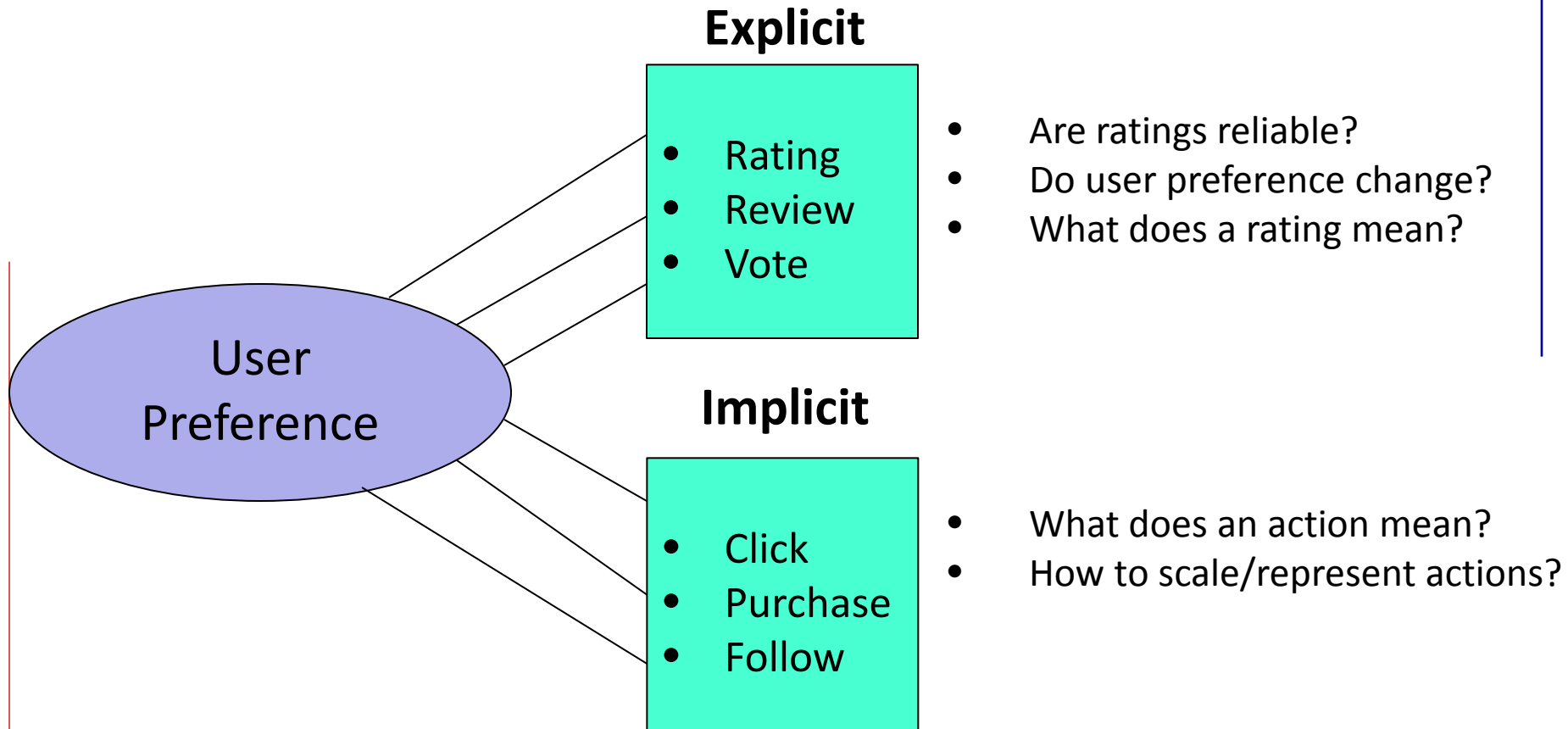
- C: user (profile) = {ID, age, gender, income, etc}
- S: item (profile) = {ID, title, genre, director, lead actor, etc}
- Utility function that measures the usefulness of item S to user C;  
ie  $u: C \times S \rightarrow U$
- U can be specified or computed.
- The goal is to predict the blanks

	Harry Potter	Matrix	Twilight	Star Wars	Toy Story	Manchester by the Sea
Mary	4			5	1	
Robert	5	5	4			
Sweta			2	4	5	
Yin		3				3

Utility matrix with ratings on 1-5 scale, many entries are unknown -> sparse matrix

# Preference & Rating Model

Recommenders mine what users say and do



# Example of Data collection

- Method of improving recommendations

The screenshot shows the Amazon.com interface for 'Improve Your Recommendations'. The top navigation bar includes the Amazon logo, a search bar, and links for Prime Video, Departments, Prime, Fresh, Video, Music, Sell, Gift Cards & Registry, Deals, Your Amazon.com, Orders, Account & Lists, and Cart. Below the navigation bar, there are links for 'Your Amazon.com', 'Your Browsing History', 'Recommended For You', 'Improve Your Recommendations', 'Your Profile', and 'Learn More'. The main content area is titled 'Your Amazon.com > Improve Your Recommendations' and includes a sub-header 'Items you've purchased'. A sidebar on the left lists various collection categories like 'Items you've purchased', 'Videos you've watched', etc. The main content area displays two items: a Staedtler Mechanical Pencil and Brooklyn Beans Assorted Coffee. The 'Rate this Item' section for the pencil is circled in red, showing a 5-star rating and options to mark it as a gift or not use for recommendations.

**EDIT YOUR COLLECTION**

- Items you've purchased
- Videos you've watched
- Items you've marked "I own it"
- Items you've rated
- Items you've marked "Not interested"
- Items you've marked as gifts

**Need Help?**  
Visit our help area to

**Items you've purchased**

**Your Rating:**

**Rate this Item**  
 ★★★★★  
 This was a gift  
 Don't use for recommendations

**Rate this Item**  
 ★★★★★  
 This was a gift  
 Don't use for recommendations

# Major Challenges

- Data sparsity - large # of items, little overlap with two similar users
- Scalability – data is sparse but large sites have millions of users and items
- Cold Start – insufficient info on new users
- User Interface – how info is presented to user
- Evaluation of recommendations – best solution for a situation
- Diversity vs accuracy – popular items versus less obvious items
- Time stamps – old opinions and search vs current

# Recommenders

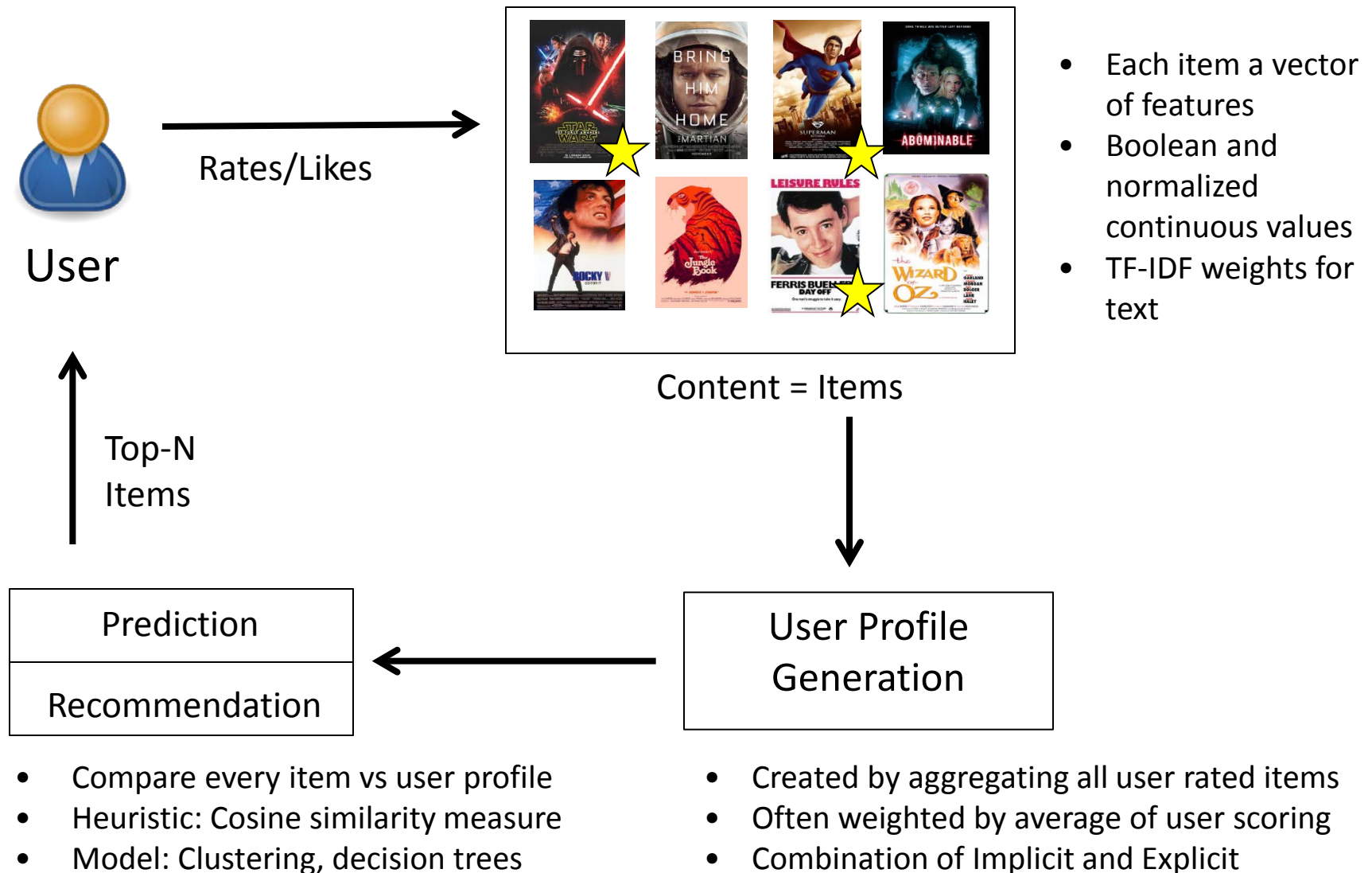
---

- Basic Concepts and Terminology
- High Level Architecture
- Explanation
- Pros and Cons

# Content based Recommenders

- Main idea is to recommend items to user  $x$  similar to previous items rated highly by  $x$ 
  - Movies: Genre, Director, Actors
  - News: Similar content read before
  - People: Common Friends
- Roots in information retrieval and information filtering
- Commonly using textual information
  - Documents, news, web sites

# Content based RS – High Level Architecture





# Content Based RS – Item Profiles

- When the items are text based the attributes are keywords
- Recommendations made based on keywords
- We don't care about a term that appears everywhere..."the"
- Most common method to specify keyword "weights" is:
- TF-IDF = Term Frequency \* Inverse Document Frequency

## Weighting

- TF – how often a term appears in a document
- IDF – how few documents contain the term

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

$tf_{i,j}$  = number of occurrences of  $i$  in  $j$

$df_i$  = number of documents containing  $i$

$N$  = total number of documents

# Content Based RS - Similarity

- The **cosine measure similarity** is another similarity metric that depends on envisioning user preferences as points in space. User preferences as points in an **n-dimensional space**.
- When two items are similar, they'll have similar ratings, and so will be relatively close in space and the angle formed between these two lines will be relatively small. When the two items are dissimilar, their points will be distant forming a wide angle.

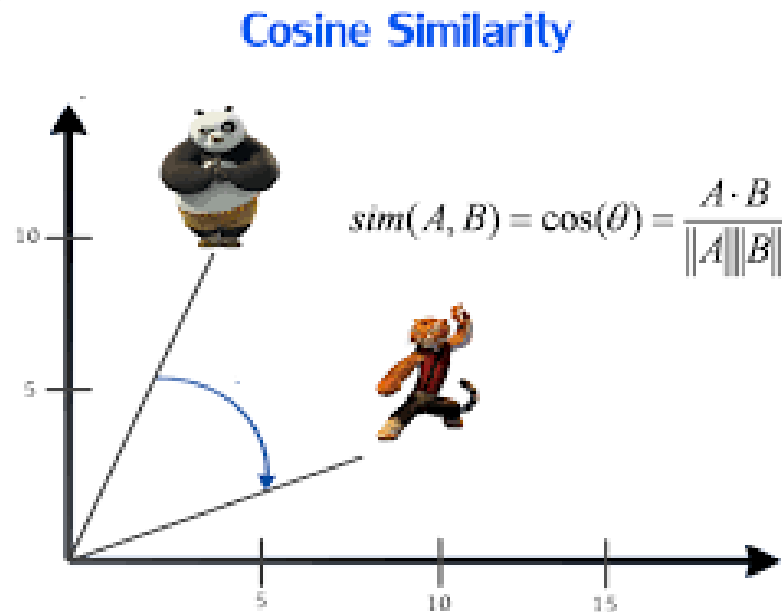


Image source: [sungsoo.github.io](https://github.com/sungsoo)

Def: Mahout in Action

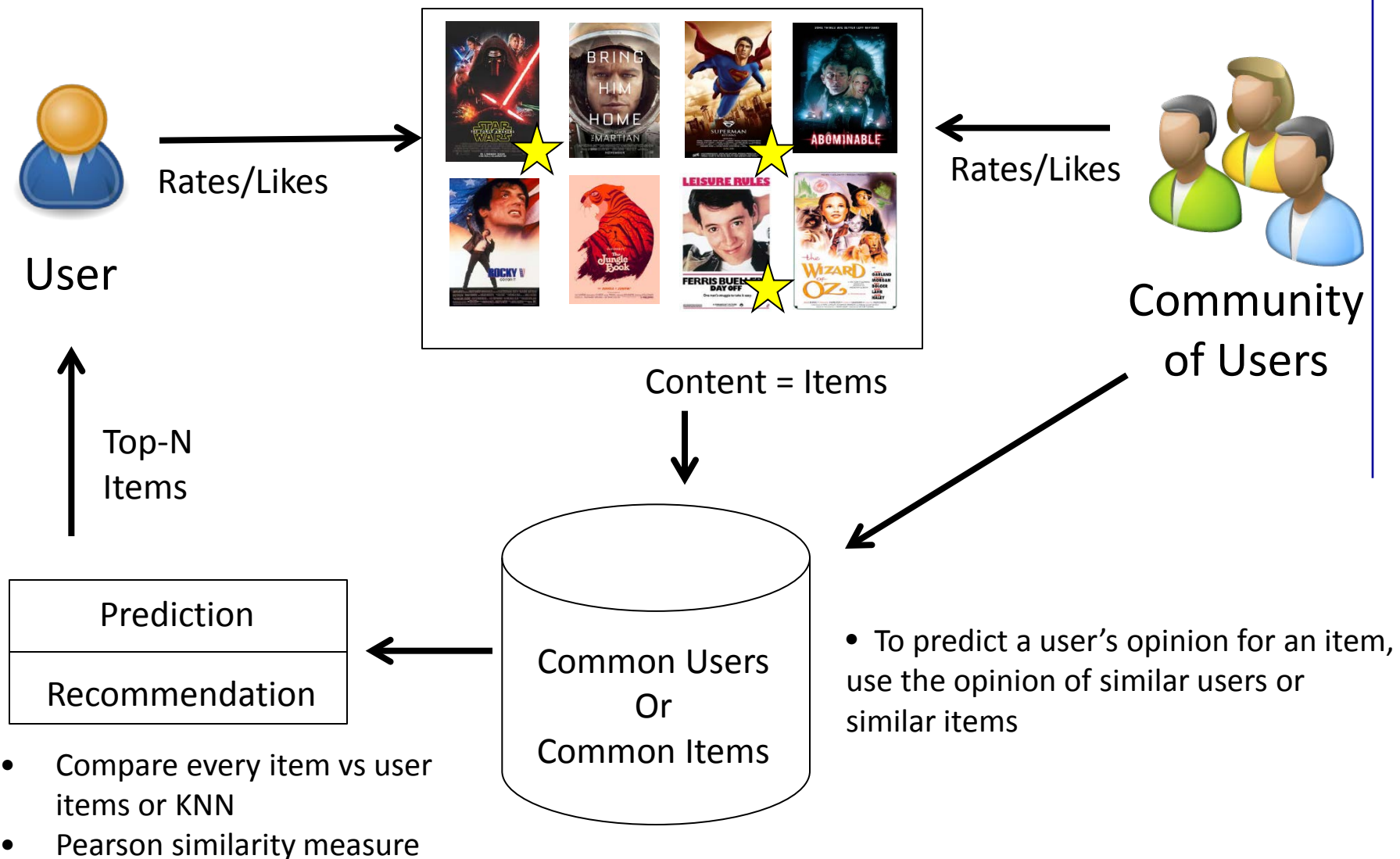
# Content based RS – Pros and Cons

- Pros
  - No data from other users required
  - Recommendations to users with unique tastes
  - Able to recommend new and unpopular items
  - Easy to explain results to user
- Cons
  - **Cold Start Problem**
    - The user has to rate items before system can make prediction
  - Limited Content Analysis
    - Features for images, music hard to categorize
  - Overspecialization
    - No recommendations beyond user profile

# Collaborative Filtering Based Recommenders


- Recommended items are chosen based on the past evaluations of a large group of similar users, called a neighborhood, either item to item or user to user.
- Does NOT rely on features of items (content approach)
- The collaborative filter approaches can be divided into two types:
  - Memory-based CF (User Based) – chose a similar set of users using some type of distance metric, try to infer ratings as an average of ratings made by neighborhood on similar items.
    - *"You read these 10 books, so you might also like to read ..."*
  - Model-based CF (Item Based) – Directly recommend items similar to items that have been rated highly by the user.
    - *"Users that read this book also read ..."*

# Collaborative Filtering based RS – High Level Architecture



# User-based Collaborative Filtering

	Item 1	Item 2	Item 3	Item 4	Item 5
User 1		1	?	5	2
User 2	2	?	5	1	5
User 3	5	3		4	1
User 4		4	3		
User 5	1	5	2		
User 6	5			4	1



- Objective is to make a recommendation to a new user -> User 1 based on other users rating
- Step 1: Pick a set of similar users using a distance metric
- Step 2: Determine the User1 neighborhood (items both users have rated)
- Step 3: Infer the ratings User 1 would assign to items not yet rated as an average of ratings made by similar users on those items.
- Step 4: Provide a Score and Top-N set of recommendations to User 1
- Challenge: How to correct for user to user variation in rating?
- Computationally expensive.

# Item-Based Collaborative Filtering

	Item 1	Item 2	Item 3	Item 4	Item 5
User 1		1	3	2	4
User 2	2		5	1	5
User 3	5	5	?	4	1
User 4		1	3		
User 5	1	5	2		
User 6	5			3	1



Compute only co-rated items

Objective is to make a recommendation to a new user based ratings of similar items

Step 1: Compute the similarity between highly rated user items and all items

Step 2: Select and retain only the top k most similar items (analogous to neighborhood)

Step 3: Predict items based on a weighted sum of ratings made by user on similar items; also other techniques include regression, probabilistic, cluster models, Bayesian networks, more recently latent semantic analysis.

Step 4: Provide a score and Top-N set of recommendations to User 1

Challenge: Sub-optimal recommendations

Requires frequent retraining (can be done offline)

# Similarity - Collaborative Filtering

- Both content based and CF approaches use the same cosine measure for calculating similarity.
- In content systems it is used to measure similarity between vectors containing TF-IDF weights.
- In CF systems it measures similarity between vectors of actual user-specified ratings.
- Often values are normalized by subtracting the average value of the row or column
- Other ways to compute similarity

$$r = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2} \sqrt{\sum_i (y_i - \bar{y})^2}}$$

Pearson correlation coefficient

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

Jaccard Index (Binary Data)



# Classification of Recommender Systems

Approach	Heuristic Based	Model Based
Content Based	<ul style="list-style-type: none"><li>• TF-IDF</li><li>• Clustering</li></ul>	<ul style="list-style-type: none"><li>• Bayesian Classifiers</li><li>• Clustering</li><li>• Decision Trees</li><li>• Neural Networks</li></ul>
Collaborative	<ul style="list-style-type: none"><li>• Nearest Neighbor (cosine, correlation)</li><li>• Clustering</li><li>• Graph Theory</li></ul>	<ul style="list-style-type: none"><li>• Bayesian Networks</li><li>• Clustering</li><li>• Neural Networks</li><li>• Linear Regression</li><li>• Probabilistic Models</li></ul>
Hybrid	<ul style="list-style-type: none"><li>• Linear Combination</li><li>• Voting Schemes</li><li>• Combinations</li></ul>	<ul style="list-style-type: none"><li>• Combining Content and Collaborative Approaches</li></ul>

# Collaborative Filter based RS – Pros and Cons

- Pros
  - Works for any kind of item; no feature selection is needed
- Cons
  - **Cold Start Problem**
    - The users has to rate items before system can make prediction
  - Sparsity
    - Hard to find pairs of users + items
  - First Rater
    - New items have no ratings
  - Popularity Bias
    - Crowd out unique recommendations
  - Gray sheep
    - Groups of users are needed with overlapping characteristics. Even if such groups exist, individuals who do not consistently agree or disagree with any group of people will receive inaccurate recommendations

# Hybrid Recommenders - Summary

Hybrid Method	Description
Weighted	The scores (or votes) of several recommendation techniques are combined together to produce a single recommendation
Switching	The system switches between recommendation techniques depending on the current situation
Mixed	Recommendations from several different recommenders are presented at the same time
Feature Combining	Features from different recommendation data sources are thrown together into a single recommendation algorithm
Cascade	One recommender refines the recommendations given by another.
Feature Augmentation	Output from one technique is used as an input feature to another.
Meta-Level	The model learned by one recommender is used as input to another.

# Extending Recommender Systems

- Multi-Criteria Ratings
  - Restaurant
    - Food, Service, Decor
- User x Item Space + Contextual Info
  - Time: time of the year, season, day
  - Shared circumstances: People
  - Place
- Non-intrusiveness
  - What is the optimal number of new user ratings needed?
- Effectiveness Measures
  - Usefulness vs Quality/Accuracy

# Very Large Data Sets - SVD

- Challenge: The rating matrix becomes very large and is sparse
- Technique: Reduce the dimensionality of the matrix while retaining as much information as possible
- The matrix product  $UV$  of utility matrix  $M$  gives values for all user-item pairs, that value can be used to predict the blanks values in the utility matrix.
- Root Mean Square (RMSE) is used to measure how close the product  $UV$  is to a given utility matrix.

# Open Source Resources

---

- Dr. Hahsler SMU Short Course
- [http://michael.hahsler.net/other\\_courses/ICMA Recommendation Tools/](http://michael.hahsler.net/other_courses/ICMA_Recommendation_Tools/)
- There is a comprehensive list by Graham Jenson on Github:
- [https://github.com/grahamjenson/list\\_of\\_recommender systems](https://github.com/grahamjenson/list_of_recommender_systems)

# References

## Web Sites

- <https://medium.com/hacking-and-gonzo/how-reddit-ranking-algorithms-work-ef111e33d0d9#.1je924rrk>
- <https://medium.com/hacking-and-gonzo/how-hacker-news-ranking-algorithm-works-1d9b0cf2c08d#.y4f93w2k8>
- <http://recommender-systems.org/>

## Papers

- **Adomavicius, G and Tuzhilin, A (2005). “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, Vol 17, No.6**
- Suwar, Karypis, Konstan and Riedl, (2001). “Item-based Collaborative filtering recommendation algorithms”, WWW10, Hong Kong
- Bobadilla, Ortega, Hernando, Gutierrez (2013). “Recommender systems survey, Knowledge based systems” 46 (2013) 109-132.
- Lu, Medo, Yeung, Zhang, Zhang, Zho (2012). “Recommender systems”, Physics Reports 519 (2012), 1-49
- M Hahsler, “recommenderlab: A Framework for Developing and Testing Recommendation Algorithms”
- R. Burke, “Hybrid Recommender Systems: Survey and Experiments “
- Koren, Bell, Volinsky, (2009).”Matrix Factorization Techniques for Recommender Systems” IEEE Computer Society
- B.C. Chen, Yahoo Labs, “Latent Factor Models for Web Recommender Systems”

## Books

- A. Rajaraman and D. Ullman (2012), Mining of Massive Datasets, Chapter 9, Recommendation Systems, Pg 277
- R. Forte (2015), Mastering Predictive Analytics with R, Chapter 11, Recommendation Systems, Pg 339
- Ricci, Rokach, Shapira (2011), Recommender Systems Handbook, Chapter 1

## Courses

- Coursera, University of Minnesota, (2016) “Introduction to Recommender Systems: Non-Personalized and Content Based”
- **Leskovec, Rajaraman, Ullman, Stanford, “Mining of Massive Datasets”, Youtube**
- M. Hahsler, SMU (2016), Short Course, “Recommender Systems, Harnessing the Power of Personalization”