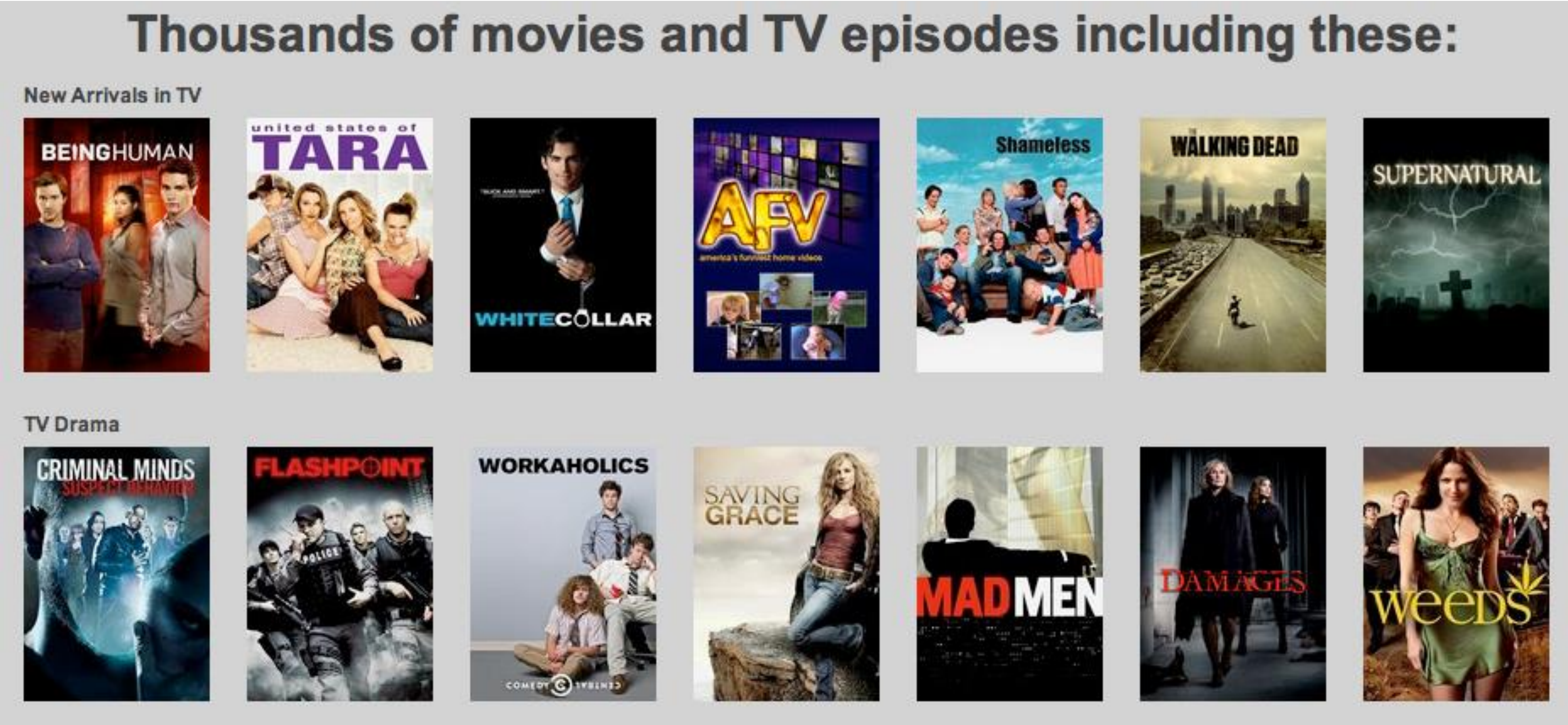
RECOMMENDER SYSTEM

EMMANUEL JAKOBOWICZ


Thousands of movies and TV episodes including these:
New Arrivals in TV
BEINGHUMAN
united states of TARA
WHITECOLLAR
AFV
Shameless
WALKING DEAD
SUPERNATURAL
TV Drama
CRIMINAL MINDS
FLASHPOINT
WORKAHOLICS
SAVING GRACE
MAD MEN
DAMAGES
weeds

Just For Today

Browse Recommended

Recommendations

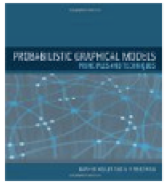
- Amazon Instant Video
- Appstore for Android
- Baby
- Beauty
- Books
- Books on Kindle
- Camera & Photo
- Clothing & Accessories
- Computers & Accessories
- Electronics
- Grocery & Gourmet Food
- Health & Personal Care
- Home Improvement
- Industrial & Scientific
- Jewelry
- Kitchen & Dining
- MP3 Downloads
- Magazine Subscriptions
- Movies & TV
- Music
- Musical Instruments
- Office & School Supplies
- Patio, Lawn & Garden
- Shoes
- Software
- Sports & Outdoors
- Toys & Games
- Video Games
- Watches

These recommendations are based on items you own and more.

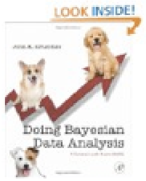
view: **All** | New Releases | Coming Soon — More results

1. 

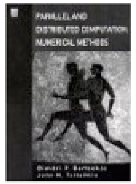
Convex Optimization
by Stephen Boyd (March 8, 2004)
Average Customer Review: (13)
In Stock
List Price: ~~$84.00~~
Price: $68.13
44 used & new from $61.32
Add to Cart | Add to Wish List
I own it | Not interested | Rate this item
Recommended because you purchased **Nonlinear Programming** and more (Fix this)

2. 

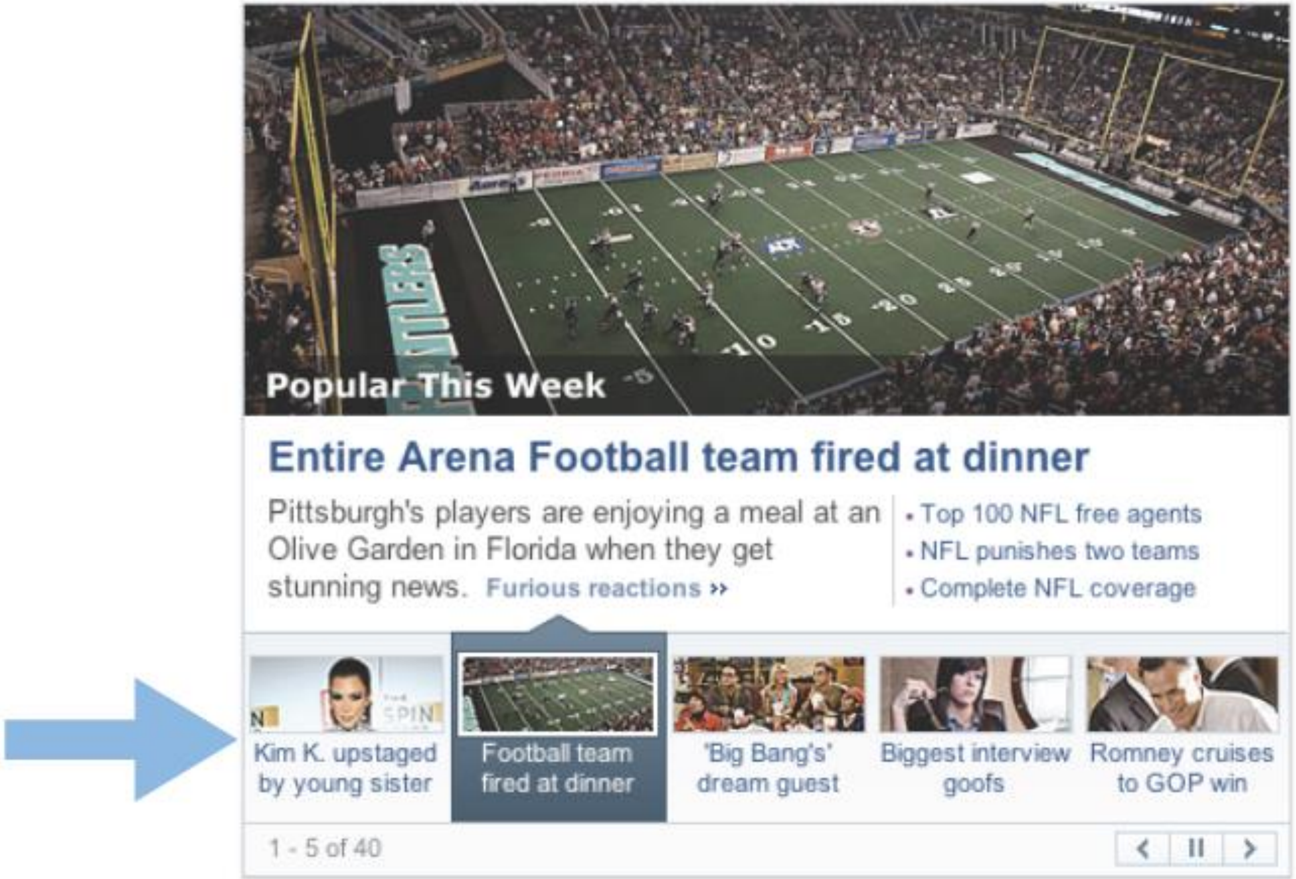
Probabilistic Graphical Models: Principles and Techniques (Adaptive Computation and Machine Learning series)
by Nir Friedman (July 31, 2009)
Average Customer Review: (11)
In Stock
List Price: ~~$95.00~~
Price: $93.55
47 used & new from $91.33
Add to Cart | Add to Wish List
I own it | Not interested | Rate this item
Recommended because you purchased **Nonlinear Programming** and more (Fix this)

3. 

Doing Bayesian Data Analysis: A Tutorial with R and BUGS
by John K. Kruschke (November 10, 2010)
Average Customer Review: (15)
In Stock
List Price: ~~$89.95~~
Price: $77.98
44 used & new from $68.40
Add to Cart | Add to Wish List
I own it | Not interested | Rate this item
Recommended because you purchased **Bayesian Nonparametrics** and more (Fix this)

4. 

Parallel and Distributed Computation: Numerical Methods (Optimization and Neural Computation)
by Dimitri P. Bertsekas (January 1, 1997)
Average Customer Review: (1)
In Stock
Price: $49.50
15 used & new from $45.49
Add to Cart | Add to Wish List
I own it | Not interested | Rate this item
Recommended because you purchased **Nonlinear Programming** (Fix this)



Adapt to your preferences

“The Web is leaving the era of search and entering one of discovery. What's the difference?

Search *is what you do when you're looking for something.* ***Discovery*** *is when something wonderful that you didn't know existed, or didn't know how to ask for, finds you.” –* *CNN Money, “The race to create a 'smart' Google*

SOME FACTS

- Netflix: 2/3 of the movies watched are recommended
- Google News: recommendations generate 38% more click-throughs
- Amazon: 35% sales from recommendations

THE PROBLEM

Estimate a utility function

to predict how

a user will like an item.

WHAT IS A GOOD RECOMMENDATION?

- is relevant to the user: *personalized*
- is diverse: it represents all the possible interests of one user
- Does not recommend items the user already knows or would have found anyway.
- Expands the user's taste into neighboring areas

- Users take into account only few suggestions.
- There is a need to do better on the top scoring recommended items

EXAMPLES

- Movie recommendation (Netflix)
- Related product recommendation (Amazon)
- Web page ranking (Google)
- Social recommendation (Facebook)
- News content recommendation (Yahoo)
- Priority inbox & spam filtering (Google)
- Online dating (Meetic)

WHAT SHOULD YOU DO?

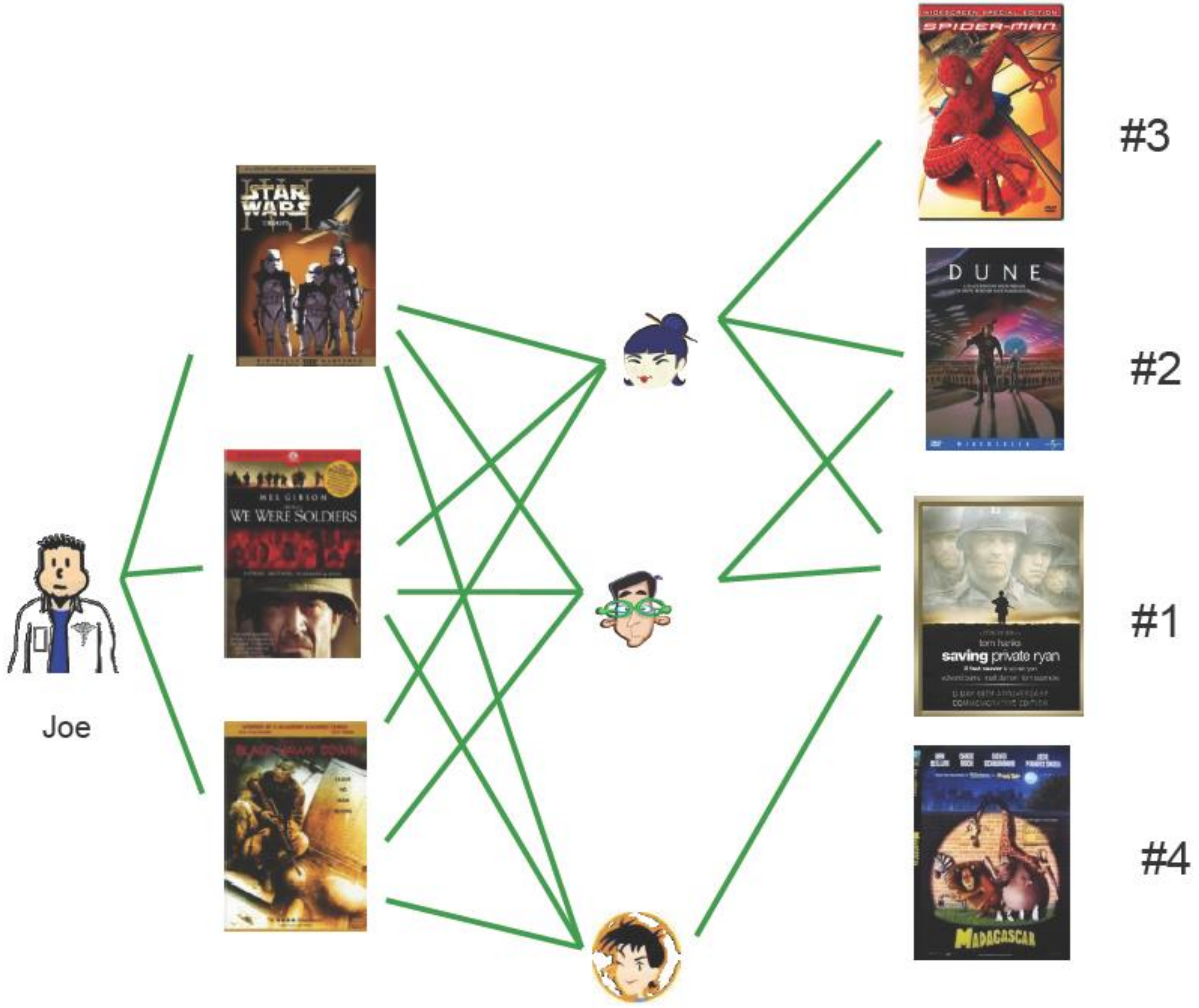
- Depends on the domain and particular problem
- best approach is Collaborative Filtering.
- Other approaches can be combined to improve results

- What matters?
  - Data preprocessing: outlier removal, denoising, removal of global effects
  - "Smart" dimensionality reduction
  - Combining methods

COLLABORATIVE FILTERING

- The task of predicting (filtering) user preferences on new items by collecting taste information from many users (collaborative).
- Challenges:
  - many items to choose from
  - very few recommendations to propose
  - few data per user
  - no data for new user
  - very large datasets

NEIGHBORHOOD METHODS



NEIGHBORHOOD METHODS

- (user,user) similarity to recommend items
  - good if item base is smaller than user base
  - good if item base changes rapidly
  - traverse bipartite similarity graph
- (item,item) similarity to recommend new items that were also liked by the same users
  - good if the user base is small

PROPERTIES

- Intuitive
- No (substantial) training
- Handles new users / items
- Easy to explain to user
- Accuracy & scalability questionable

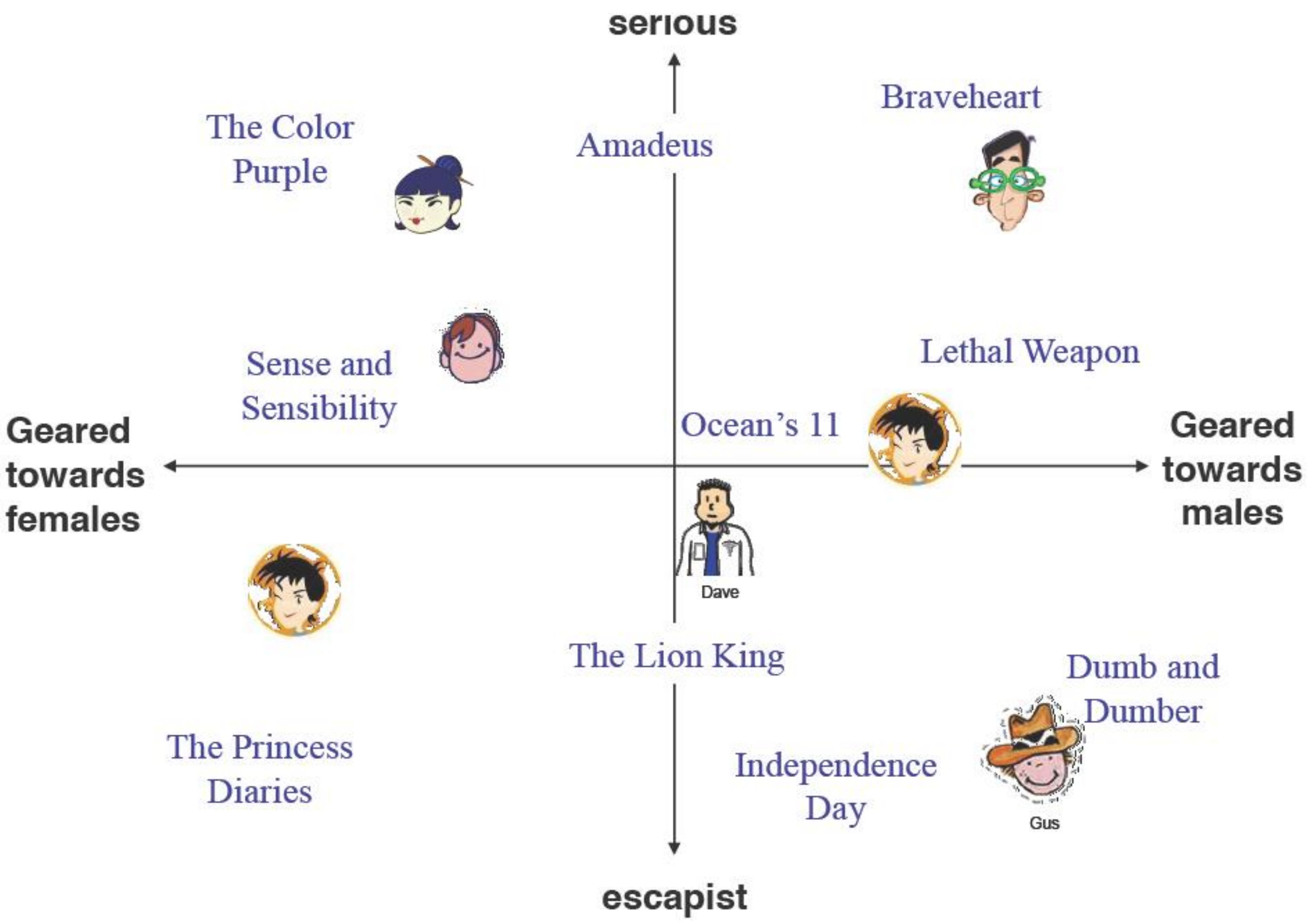
- Bias problems:
  - Some items are significantly higher rated
  - Some users rate substantially lower
  - Ratings change over time
- Bias correction is crucial for nearest neighborhood recommender algorithm
  - Offset per user
  - Offset per movie
  - Time effects
  - Global bias

SIMILARITY MEASURES

- Pearson correlation coefficient
- Empirical Pearson correlation coefficient
- Jaccard similarity
- Observed/expected ratio

MATRIX FACTORIZATION

- Using a factorial method like SVD (singular value decomposition) to obtain similarities



TYPES OF COLLABORATIVE FILTERING

- Memory based collaborative filtering
  - User-based
  - Item-based
- Model based collaborative filtering

USER-BASED COLLABORATIVE FILTERING

- Each user has expressed an opinion for some items:
  - Explicit opinion: rating score
  - Implicit: purchase records or listen to tracks
- Target (or Active) user for whom the recommendation task is performed
  1. Identify set of items rated by the target user
  2. Identify which other users rated 1+ items in this set (neighborhood formation)
  3. Compute how similar each neighbor is to the target user (similarity function)
  4. In case, select k most similar neighbors
  5. Predict ratings for the target user's unrated items (prediction function)
  6. Recommend to the target user the top N products based on the predicted ratings

Use Pearson correlation as similarity measure

ITEM-BASED COLLABORATIVE FILTERING

- The basic steps:
  - Identify set of users who rated the target item i
  - Identify which other items (neighbours) were rated by the users set
  - Compute similarity between each neighbour & target item (similarity function)
  - In case, select k most similar neighbours
  - Predict ratings for the target item (prediction function)

Use Pearson correlation as similarity measure

PERFORMANCE ISSUES

- User-based similarity is more **dynamic**.
  - Precomputing user neighbourhood can lead to poor predictions.
- Item-based similarity is **static**.
  - We can precompute item neighbourhood. Online computation of the predicted ratings.

CONCLUSIONS ON MEMORY BASED COLLABORATIVE FILTERING

- Pros:
  - Requires minimal knowledge engineering efforts
  - Users and products are symbols without any internal structure or characteristics
  - Produces good-enough results in most cases
- Cons:
  - Requires a large number of explicit and reliable "ratings"
  - Requires standardized products: users should have bought exactly the same product
  - Assumes that prior behaviour determines current behaviour without taking into account "contextual" knowledge

MAJOR PROBLEM: SPARSE DATA

- Typically large product sets & few user ratings e.g. Amazon:
  - in a catalogue of 1 million books, the probability that two users who bought 100 books each, have a book in common is 0.01
  - in a catalogue of 10 million books, the probability that two users who bought 50 books each, have a book in common is 0.0002
- Collaborative filtering must have a number of users ~ 10% of the product catalogue size
- Methods for dimensionality reduction
  - Matrix Factorization
  - SVD
  - Clustering

Need for models and machine learning

MODEL BASED COLLABORATIVE FILTERING

- Models are learned from the underlying data
- Models of user ratings (or purchases):
  - Clustering (classification)
  - Association rules
  - Matrix Factorization
  - Restricted Boltzmann Machines
  - Recurrent Neural Networks
  - Other models:
    - Bayesian network (probabilistic)
    - Probabilistic Latent Semantic Analysis …

CLUSTERING

- Cluster customers into categories based on preferences & past purchases
- Compute recommandations at the cluster level:

    all customers within a cluster receive the same recommendations

Which method could be used?

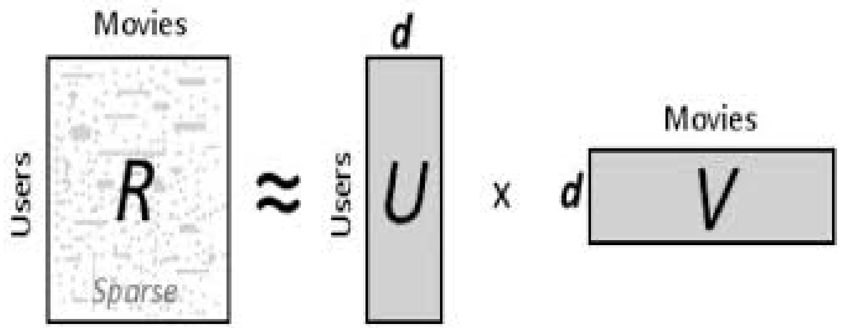
CLUSTERING

- Pros:
  - It can also be applied for selecting the k most relevant neighbours in a CF algorithm
  - Faster: recommendations are per cluster
- Cons:
  - less personalized: recommendations are per cluster vs. in CF they are per user

ASSOCIATION RULES

- Past purchases used to find relationships of common purchases
- Pros:
  - Fast to implement
  - Fast to execute
  - Not much storage space required
  - Not « individual » specific
  - Very successful in broad applications for large populations, such as shelf layout in retail stores
- Cons:
  - Not suitable if preferences change rapidly
  - Rules can be used only when enough data validates them. False associations can arise

MATRIX FACTORIZATION



- Matrix factorization allows to obtain factors for both users and products
- Then distances can be computed

DEEP LEARNING METHODS

- Restricted Boltzmann Machines
  - A (generative stochastic) Neural Network Learns a probability distribution over its inputs
  - Used in dimensionality reduction, CF, topic modeling, feature learning
  - Essential components of Deep Learning methods (DBN's, DBM's)

- Recurrent Neural Networks for CF
  - RNN's are Neural Networks designed to model sequences
  - RNN for CF tries to predict the next item given all previous ones
    - *e.g. predict the next song, video, given the history of streams.*
  - *RNN's model the current output as a function of the previous output and a hidden (latent) state*
  - RNN models the items

OTHER APPROACHES

- Content-based Recommendations
  - Recommendations are based on the information on the *content* of items rather than on other users' opinions.
  - Use a machine learning algorithm to model the users' preferences from examples based on a description of the content.
- Context-aware Recommendations
  - User preferences may differ with context (time of day, season, mood, device characteristics, location, options offered by the system etc.). A context-aware recommender system takes the context into account when computing suggestions.

CONTENT-BASED RECOMMENDATIONS

- Suitable for text-based products (web pages, books)
- Items are "described" by their features (e.g. keywords)
- Users are described by the keywords in the items they bought
- Recommendations based on the match between the content (item keywords) and user keywords
- The user model can also be a classifier (Neural Networks, SVM, Naïve Bayes...)

- Pros:
  - No need for data on other users.
  - No cold-start or sparsity problems.
  - Can recommend to users with unique tastes.
  - Can recommend new and unpopular items
  - Can provide explanations of recommended items by listing content-features that caused an item to be recommended.
- Cons:
  - Only for content that can be encoded as meaningful features.
  - Some types of items (e.g. movies, music)are not amenable to easy feature extraction methods
  - Even for texts, IR techniques cannot consider multimedia information, aesthetic qualities, download time: a positive rating could be not related to the presence of certain keywords
  - Users' tastes must be represented as a learnable function of these content features.
  - Hard to exploit quality judgements of other users.

CONTEXT-AWARE RECOMMENDATIONS

- Context is a dynamic set of factors describing the state of the user at the moment of the user's experience
- Context factors can rapidly change and affect how the user perceives an item
  - Temporal: Time of the day, weekday/end
  - Spatial: Location, Home, Work etc.
  - Social: with Friends, Family
- Recommendations should be tailored to the user & to the current *Context* of the user

CONLUSIONS

- Recommender Systems are an important application of Machine Learning
- Recommender Systems have the potential to become as important as Search is now
- However, Recommender Systems are more than Machine Learning
  - Economical models
  - ...
- Recommender Systems are fairly new but already grounded on well-proven technology
  - Collaborative Filtering
  - Machine Learning
  - Content Analysis
  - Social Network Analysis
  - …

SOME RESOURCES

- Recsys Wiki: http://recsyswiki.com/
- Recsys conference Webpage: http://recsys.acm.org/
- Recommender Systems Books Webpage: http://www.recommenderbook.net/
- MyMediaLite Project: http://www.mymedialite.net/

BUILDING A SIMPLE RECOMMANDATION TOOL IN PYTHON

- Use sparse data and sklearn with binder