CS520 Web Programming
Recommendation Systems

Chengyu Sun
California State University, Los Angeles

Recommendation Systems

- *Predict* items a user may be interested in based on information about the user and the items
- An effective way to help people cope with information overload
- Examples: Amazon, Netflix, Tivo, …

So How Can We Do It?

- The content based approach
  - E.g. full text search results
- The user feedback based approach
  - E.g. rating and modding
- *Which one is better?? Any room for improvement??*

Collaborative Filtering

- Rate items based on the ratings of other users *who have similar taste as you*

Problem Definitions

- Prediction
  - Given: a user and k items
  - Return: predicted rating for each item
- Recommendation
  - Given: a user
  - Return: k items from the database with the highest predicted rating

Basic Assumptions

- Items are evaluated by users explicitly or implicitly
  - Ratings, reviews
  - Purchases, browsing behaviors
  - …
- We may map explicit and implicit evaluations to a rating scale, e.g. 1-5.

Heuristic

- ◈People who agreed in the past are likely to agree in the future

Problem Formulation

- ◈User-Item Matrix

Item	Ken	Lee	Meg	Nan
1	1	4	2	2
2	5	2	4	4
3			3	
4	2	5		5
5	4	1		1
6	??	2	5	

So what would be Ken's rating for Item 6??

Pearson Correlation Coefficient

- ◈Let `x` and `y` be two users, and `r`$_{x,j}$ be the rating of item `i` by user `x`

$$w_{x,y} = \frac{\mathrm{cov}(x,y)}{\sigma_x \sigma_u} = \frac{\sum_i (r_{x,i} - \bar{r}_x)(r_{y,i} - \bar{r}_y)}{\sqrt{\sum_i (r_{x,i} - \bar{r}_x)^2}\sqrt{\sum_i (r_{y,i} - \bar{r}_y)^2}}$$

So what is $w_{ken,lee}$ *??*

Predicted Rating

- ◈`p`$_{x,i}$ is the predicted rating of item `i` by user `x`

$$p_{x,i} = \bar{r}_x + \frac{\sum_u (r_{u,i} - \bar{r}_u) \times w_{x,u}}{\sum_u w_{x,u}}$$

So what is $p_{ken,6}$ *??*

Algorithm Quality Metrics

- ◈Coverage – percentage of items for which the system can produce a prediction
- ◈Accuracy
  - Statistical metrics
    - Mean Absolute Error (MAE)
  - Decision-support metrics
- ◈Efficiency
  - Throughput – number of recommendations per second

Variations and Optimizations

- ◈Similarity measure
- ◈Significance weighting
- ◈Item rating variance
- ◈Neighborhood selection
- ◈Combine neighborhood ratings

Similarity Measures

- Pearson Correlation
- Spearman Correlation
- Cosine similarity
- Entropy
- Mean-squared-difference
- …

Significance Weighting

- Weight users in additional to the similarity measure

$$w = \begin{cases} 1 & n \geq 50 \\ n/50 & n < 50 \end{cases}$$

where n is the number of items rated by both users.

Item Rating Variance

- Some items are more telling about tastes than others
  - E.g. "Sleepless in Seattle" is more telling about taste than "Titanic"
  - Give more weight to items with high variance in ratings

Neighborhood Selection

- Select a subset of users for better performance and *accuracy*
  - Correlation threshold
  - Best n neighbors

Combine Neighborhood Ratings

- Weighted average
- Deviation from mean
- Weighted average of z-scores

And The Winners Are …

- Similarity measure
  - Pearson Correlation
  - Spearman Correlation*
- Significance weighting
- Neighborhood selection
  - Best n neighbors with n≈20
- Combine neighborhood ratings
  - Deviation from mean

Other Recommendation Algorithms

- Combine collaborative and content-based filtering
- Item-item collaborative filtering
- Bayesian networks

Some Libraries

- Taste – http://taste.sourceforge.net/
- COFE – http://eecs.oregonstate.edu/iis/CoFE/
- And more – http://en.wikipedia.org/wiki/Collaborative_filtering#Software_libraries

Non-personalized Recommendation

- What if the user is new to the site?
- What if the site itself is new, i.e. no previous user transactions?

Sales Transactions

t1:	Beef, Chicken, Milk
t2:	Beef, Cheese
t3:	Cheese, Boots
t4:	Beef, Chicken, Cheese
t5:	Beef, Chicken, Clothes, Cheese, Milk
t6:	Chicken, Clothes, Milk
t7:	Chicken, Milk, Clothes

Amazon-like recommendation:

Users who purchased milk also purchased the following items:

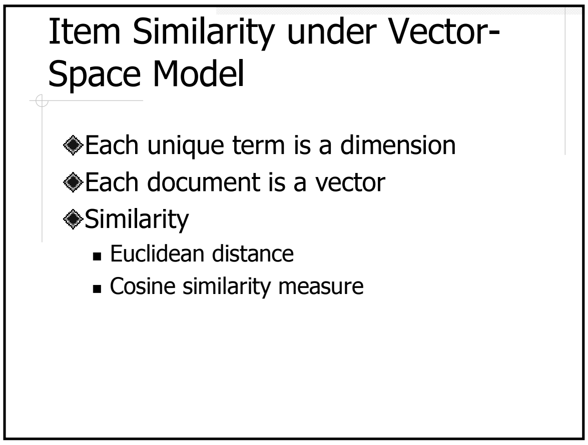
- *Clothes*
- *Chicken*

Association Rule Mining

- $\{i_1, i_2, \dots, i_n\} \rightarrow j$
- Confidence: the probability of finding item j in a transaction that has $\{i_1, i_2, \dots, i_n\}$
- Support: the number of transactions that have $\{i_1, i_2, \dots, i_n\}$ and j

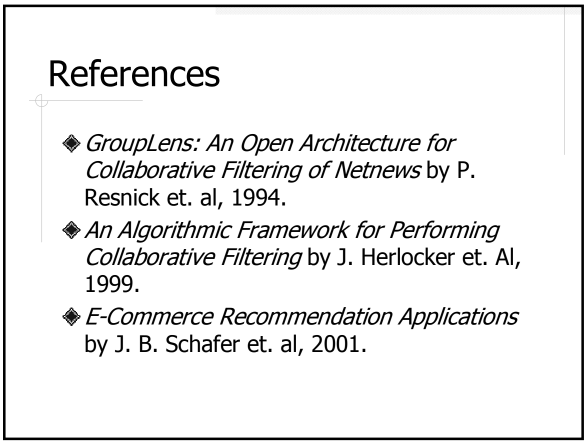
A-Priori Algorithm

- Observation: A set of items `X` has support `s`, then each subset of `X` must have support at least `s`.
- Example: find the association rules that have at least `20% support` and `50% confidence`



Item Similarity under Vector-Space Model

- Each unique term is a dimension
- Each document is a vector
- Similarity
  - Euclidean distance
  - Cosine similarity measure



References

- *GroupLens: An Open Architecture for Collaborative Filtering of Netnews* by P. Resnick et. al, 1994.
- *An Algorithmic Framework for Performing Collaborative Filtering* by J. Herlocker et. Al, 1999.
- *E-Commerce Recommendation Applications* by J. B. Schafer et. al, 2001.