

CS520 Web Programming

Collaborative Filtering

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
- ◆ The user feedback based approach

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 | | | 4 | |
| 4 | 2 | 5 | | 5 |
| 5 | 4 | 1 | | 1 |
| 6 | ?? | 2 | 5 | |

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

Solving the Problem

- ◆ Intuition: Ken's rating for Item 6 is likely to be high
 - Ken's ratings are similar to Meg's
 - Ken's ratings are opposite of Lee's
- ◆ Develop the algorithm
 1. Quantify rating similarity
 2. Calculate the predicted rating

Similarity Measure

- ◆ Pearson Correlation Coefficient
 - A measure of linear correlation of two random variables

Pearson Correlation Coefficient

- ◆ Let x and y be two users, and $r_{x,j}$ be the rating of item i by user x

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

So what is $w_{ken,lee}$?? what's the range of $w_{i,j}$?

Predict the Rating

- ◆ The predicted rating $p_{x,i}$ should be a function of
 - The past ratings of user x
 - The ratings of other users for item i , weighted by their similarity to user x

Predicted Rating

- ◆ $\hat{p}_{x,i}$ is the predicted rating of item i by user x

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

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

Variations and Optimizations

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

Other Similarity Measures ...

- ◆ Spearman Correlation
 - Uses ranks instead of raw rating scores
- ◆ Cosine similarity
- ◆ Mean squared difference
- ◆ Entropy-based
- ◆ ...

... Other Similarity Measures

Cosine similarity: $\cos(\mathbf{X}, \mathbf{Y}) = \frac{\mathbf{X} \cdot \mathbf{Y}}{|\mathbf{X}| |\mathbf{Y}|} = \frac{\sum x_i y_i}{\sqrt{\sum x_i^2} \sqrt{\sum y_i^2}}$

Mean squared difference: $msd(\mathbf{X}, \mathbf{Y}) = \frac{\sum (x_i - y_i)^2}{N}$

Entropy-based association: $h(\mathbf{X}, \mathbf{Y}) = -\sum p_{i,j} \ln p_{i,j}$

Significance Weighting

- ◆ Weight users in addition 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

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

Mean absolute deviation: $s = \frac{1}{n} \sum_{i=1}^n |r_i - \bar{r}|$

Standardized measurement (*z-score*): $z_i = \frac{r_i - \bar{r}}{s}$

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

And The Winners Are

- ◆ Similarity measure
 - Pearson Correlation
 - Spearman Correlation
- ◆ Significance weighting
- ◆ Neighborhood selection
 - Best n neighbors with $n \approx 20$
- ◆ Combine neighborhood ratings
 - Deviation from mean

Other Recommendation Algorithms

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

Collaborative Filtering Libraries

- ◆ http://en.wikipedia.org/wiki/Collaborative_filtering#Software_libraries

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.