Item-Item Extensions

Introduction

- We've now introduced item-item CF
- Item-item CF is very flexible
- This time: extending it

Extending Item-Item

- Item-item is good for extending directly
- Simple parts with well-defined interfaces provide a lot of flexibility
- It's easy to understand what extensions do

Example: User Trust

- Goal: incorporate user trustworthiness into item relatedness computation
  - User's global reputation, not per-user trust
- Solution: weight users by trust before computing item similarities
- High-trust users have more impact
- Massa and Avesani. 2004. 'Trust-Aware Collaborative Filtering for Recommender Systems'

$$w_{ij} = \frac{\sum_u \rho_u \hat{r}_{ui} \hat{r}_{uj}}{\sqrt{\sum_u \rho_u \hat{r}_{ui}^2} \sqrt{\sum_u \rho_u \hat{r}_{uj}^2}}$$

Extension: Papers and PageRank

- Recommending research papers: useful to consider items as users who purchase the paper's citations
  - Same idea can apply to web pages
- Goal: incorporate paper 'importance' into recommender
- Solution: weight paper user vectors by the paper's PageRank (or HITS hub score)
- Ekstrand et al., 2010. Automatically Building Research Reading Lists.

Restructuring: Item-Item CBF

- Basic item-item algorithm structure doesn't care how similarity is computed
- So why not use content-based similarity?
- Resulting algorithm really isn't a collaborative filter
- But it can work pretty well!
- Example: using Lucene to compare documents as neighborhood & similarity function

Restructuring: Deriving Weights

- Item-item compares individual item pairs
- Alternative approach: infer coefficients from data
  - Find coefficients w_{ij} that minimize squared error
  - Learn coefficients with standard machine learning / optimization algorithm (gradient descent)

Conclusion

- Item-item CF is flexible and versatile
- Many interesting recommenders can be built by reconfiguring it

Item-Item Hybrids and Extensions