Evaluation of Recommender Systems

Joonseok Lee

Georgia Institute of Technology

2011/04/12

Agenda

- Recommendation Task
  - Recommending good items
  - Optimizing utility function
  - Predicting ratings
- Evaluation Protocols and Tasks
  - Online evaluation
  - Offline experiment
- Evaluation Metrics
- Case Studies
- Other Issues

Task 1: Recommending Good Items

- Recommending some good items: more important not to present any disliked item.
  - Media items: movie, music, book, etc.

- Recommending all good items: more important not to skip any liked item.
  - Scientific papers which should be cited
  - Legal databases

Task 2: Optimizing Utility

- Maximizing profit!
  - Buy more than originally intended.
  - Keeping users in the website longer. (Banner advertisement)

- Generalization of Task Type 1.
  - Weighted sum of purchased items' profit.
  - When advertisement profit is considered, target function to be optimized can be very complex.

Task 3: Predicting Ratings

- Predict unseen ratings based on observed ratings.
  - Common in research community
  - Netflix competition
- Common practice
  - Recommending items according to the predicted ratings.
  - Is this a correct strategy?

Agenda

- Recommendation Task
  - Recommending good items
  - Optimizing utility function
  - Predicting ratings
- Evaluation Protocols and Tasks
  - Online evaluation
  - Offline experiment
- Evaluation Metrics
- Case Studies
- Other Issues

Online Evaluation

- Test with real users, on a real situation!
  - Set up several recommender engines on a target system.
  - Redirect each group of subjects to different recommenders.
  - Observe how much the user behaviors are influenced by the recommender system.

- Limitation
  - Very costly.
  - Need to open imperfect version to real users.
    - May give negative experience, making them to avoid using the system in the future.

Offline Experiments

- Filtering promising ones before online evaluation!
  - Train/Test data split
  - Learn a model from train data, then evaluate it with test data.

- How to split: Simulating online behaviors
  - Using timestamps, allow ratings only before it rated.
  - Hide ratings after some specific timestamps.
  - For each test user, hide some portion of recent ratings.
  - Regardless of timestamp, randomly hide some portion of ratings.

Agenda

- Recommendation Task
  - Recommending good items
  - Optimizing utility function
  - Predicting ratings
- Evaluation Protocols and Tasks
  - Online evaluation
  - Offline experiment
- Evaluation Metrics
- Case Studies
- Other Issues

Task 3: Predicting Ratings

- Goal: Evaluate the accuracy of predictions.
- Popular metrics:
  - Root of the Mean Square Error (RMSE) $RMSE = \sqrt{\frac{1}{n}\sum_{\{i,j\}}\left(p_{i,j} - r_{i,j}\right)^2};$
  - Mean Average Error (MAE) $MAE = \frac{\sum_{\{i,j\}}\left|p_{i,j} - r_{i,j}\right|}{n}$
  - Normalized Mean Average Error (NMAE) $NMAE = \frac{MAE}{r_{\max} - r_{\min}}$
- Do not differentiate between errors
  - Ex: (5 stars – 4 stars) == (3 stars – 2 stars)

Task 1: Recommending Items

- Goal: Suggesting good items (not discouraging bad items)
- Popular metrics:

	Recommended	Not recommended
Preferred	True-Positive (tp)	False-Negative (fn)
Not preferred	False-Positive (fp)	True-Negative (tn)

$$\text{Precision} = \frac{\#\text{tp}}{\#\text{tp} + \#\text{fp}}$$

$$\text{Recall (True Positive Rate)} = \frac{\#\text{tp}}{\#\text{tp} + \#\text{fn}}$$

$$\text{False Positive Rate (1 - Specificity)} = \frac{\#\text{fp}}{\#\text{fp} + \#\text{tn}}$$

Task 1: Recommending Items

- Popular graphical models
  - Precision-Recall Curve: Precision, Recall
  - ROC Curve: Recall, False Positive Rate

- So, what to use?
  - Depend on problem domain and task.
  - Example
    - Video rental service: False positive rate is not important.
      → Precision-Recall Curve would be desirable.
    - Online dating site: False positive rate is very important.
      → ROC Curve would be desirable.

Task 2: Optimizing Utility

- Goal: modeling the way of users interacting with the recommendations.
- Popular metrics:
  - Half-life Utility Score

$$R_a = \sum_j \frac{1}{2^{(idx(j)-1)/(\alpha-1)}}$$

$$R = \frac{\sum_a R_a}{\sum_a R_a^{max}},$$

  - Generalized version with a utility function

$$R_a = \sum_j \frac{u(a,j)}{2^{(idx(j)-1)/(\alpha-1)}}$$

Agenda

- Recommendation Task
  - Recommending good items
  - Optimizing utility function
  - Predicting ratings
- Evaluation Protocols and Tasks
  - Online evaluation
  - Offline experiment
- Evaluation Metrics
- Case Studies
- Other Issues

Case Study Setting

- Goal: Demonstrate that incorrect choice of evaluation metric can lead different decision.
- Algorithms used: User-based CF
  - Neighbor size = 25

Case Study 1

- Task: prediction vs. recommendation

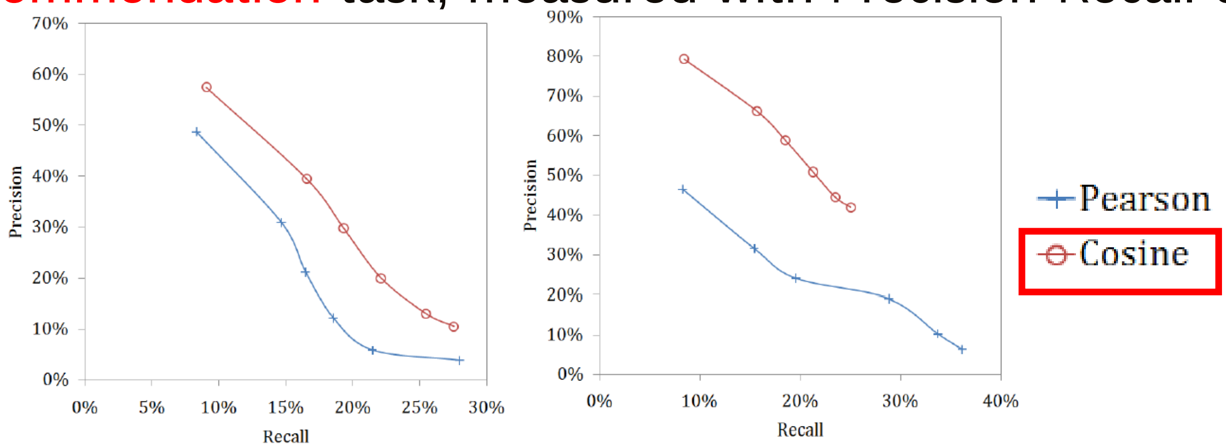
- Algorithms to compare
  - Pearson Correlation $$w(a,i) = \frac{\sum_j (v_{a,j} - \bar{v}_a)(v_{i,j} - \bar{v}_i)}{\sqrt{\sum_j (v_{a,j} - \bar{v}_a)^2 \sum_j (v_{i,j} - \bar{v}_i)^2}}$$
  - Cosine Similarity $$w(a,i) = \sum_{j \in I_{a,i}} \frac{v_{a,j}}{\sqrt{\sum k \in I_a v_{a,k}^2}} \frac{v_{i,j}}{\sqrt{\sum k \in I_i v_{i,k}^2}}$$
- Dataset
  - Netflix (users with more than 100 ratings only)
  - BookCrossing (extremely sparse)

Case Study 1

- Experimental Results
  - Prediction task, measured with RMSE

	Netflix	BookCrossing
Pearson	1.07	3.58
Cosine	1.90	4.5

  - Recommendation task, measured with Precision-Recall curve



Case Study 2

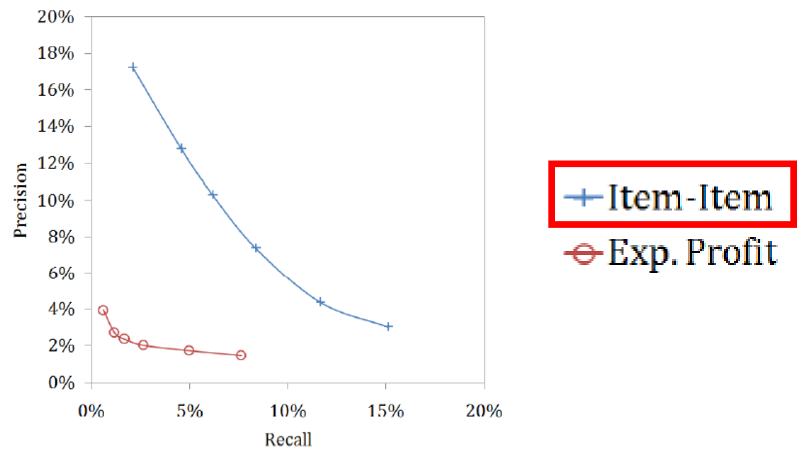
- Task: recommendation vs. utility maximization

- Algorithms to compare
  - Item to Item: maximum likelihood estimate for the conditional probabilities of each target item given each observed item.
  - Expected Utility: reflecting expected utility on the item-to-item method.

- Dataset
  - Belgian Retailer
  - News Click Stream

Case Study 2

- Experimental Results
  - Recommendation task, measured with Precision-Recall curve



  - Utility maximization task, measured with Half-life Utility score

	Score
Item-Item	0.01
Exp. Profit	0.05

Agenda

- Recommendation Task
  - Recommending good items
  - Optimizing utility function
  - Predicting ratings
- Evaluation Protocols and Tasks
  - Online evaluation
  - Offline experiment
- Evaluation Metrics
- Case Studies
- Other Issues

Other Issues

- User interface (UI) plays an important role.
  - Lots of design choices
    - Image vs. Text?
    - Horizontal vs. Vertical?
  - User study in an HCI manner

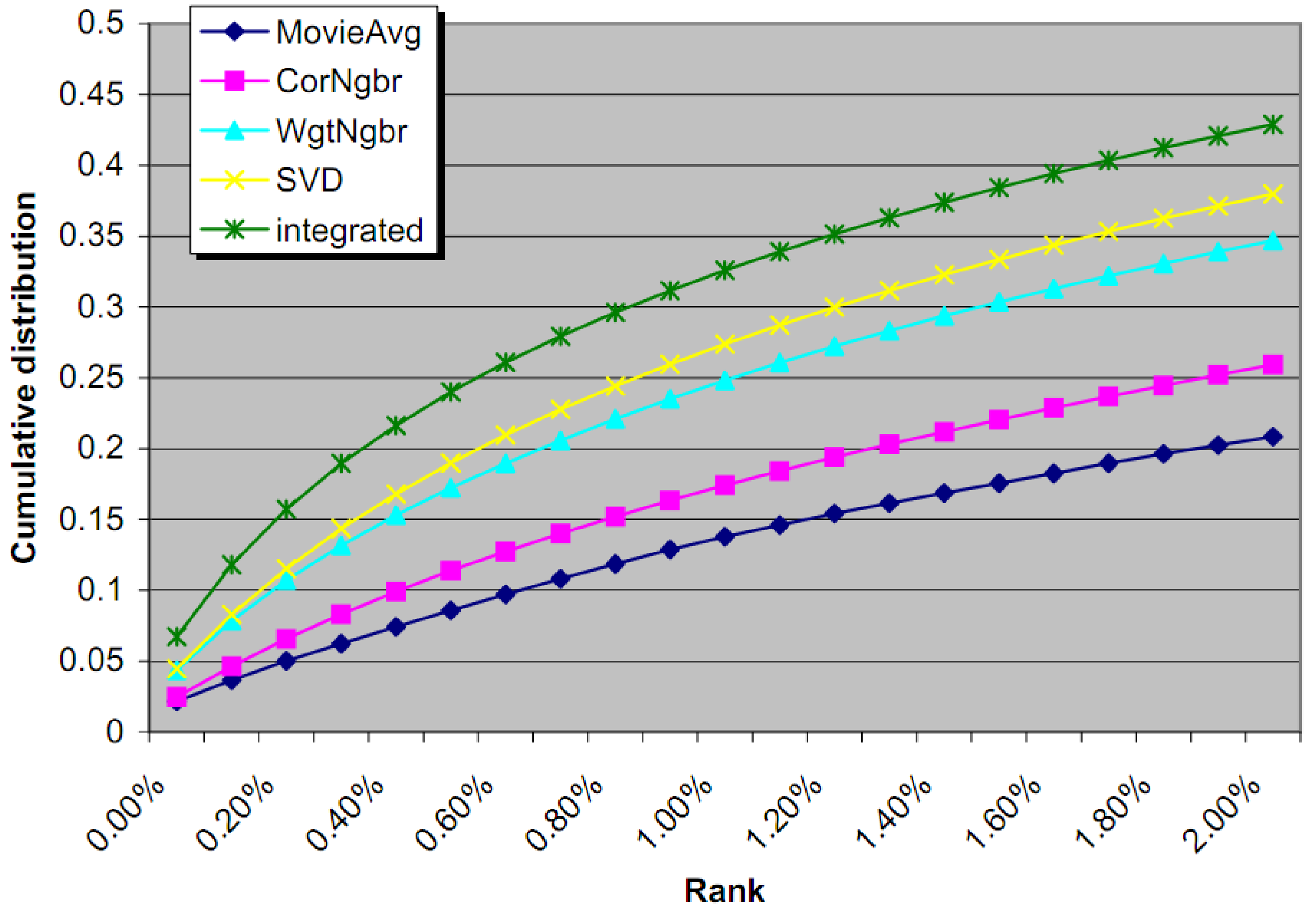
- Eliciting Utility function is not straightforward.
  - How much this recommended movie contributed the user to maintain subscription?

Other Issues (Koren, KDD'08)

- Is lowering RMSE meaningful for users, indeed?
  - Mix a favorite movie (rated as 5) with 1,000 random movies.
  - Estimate rating and rank those 1,001 movies.
  - Observe where the favorite movie is located.

- If the prediction is precise, the favorite movie should locate at the top!

Other Issues (Koren, KDD'08)



RMSE

0.8870

0.9025

0.9107

0.9406

1.0534

Any question?

THE END

Thank you very much!