

Learning Multiple-Question Decision Trees for Cold-Start Recommendation

Mingxuan Sun Fuxin Li Joonseok Lee Ke Zhou
Guy Lebanon Hongyuan Zha

College of Computing
Georgia Institute of Technology

02/07/2013

Recommendation Systems



Cold-start Recommendation

Challenge: rapid profiling of new users

Interview: learn preferences for new users

- **Static**: ask questions about selected items
[Rashid2002, Golbandi2010]

MoviePick^{alpha}

Name (optional):

Answer the questions below. Your individual responses will be kept confidential.

For each item, select an option. When finished, click the "Submit" button.

What do you feel about the following movies?

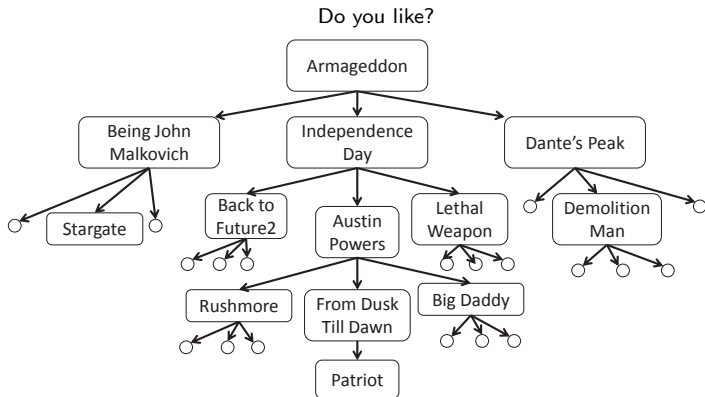
The Royal Tenenbaums	<input type="radio"/> Like	<input type="radio"/> Dislike	<input checked="" type="radio"/> Unknown
Lost in Translation	<input type="radio"/> Like	<input type="radio"/> Dislike	<input checked="" type="radio"/> Unknown
Independence Day	<input type="radio"/> Like	<input type="radio"/> Dislike	<input checked="" type="radio"/> Unknown
Being John Malkovich	<input type="radio"/> Like	<input type="radio"/> Dislike	<input checked="" type="radio"/> Unknown

- Previous answers not used to improve question selection!

Cold-start Recommendation

Interview: learn preferences for new users

- **Adaptive**: query according to previous answers
[Rashid2008,Golbandi2011,Zhou2011]



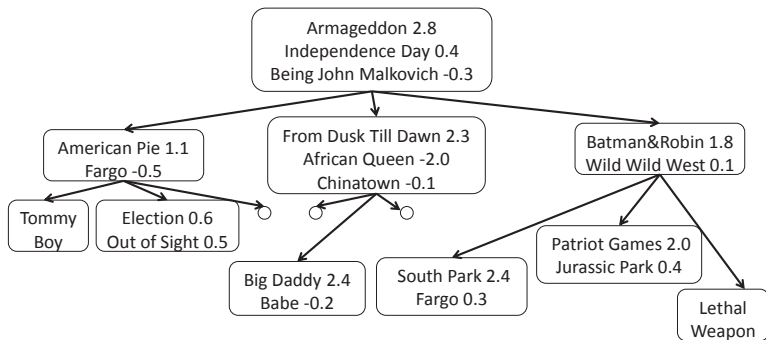
- Users repeatedly select “Unknown” before locating familiar items. Considerable amounts of user interaction!

Ours: Decision Tree with Multi-questions

- Higher recommendation accuracy
- Less user interaction

Challenges

- Search all splits: infeasible
- Learn an uniform user profile: inefficient



Learning Framework - Learn to Split at Each Node

w : weight for movies, x_i : answer of user i , l : max questions

Probabilities assigned to the left and right branches:

$$p_i = \frac{1}{1 + c \exp(-x_i^\top w)} \quad q_i = \frac{1}{1 + c \exp(x_i^\top w)}$$

Three branches:

$$L(w) = \{i | p_i \text{ max}\}, D(w) = \{i | q_i \text{ max}\}, U(w) = \{i | \text{otherwise}\}$$

Objective: to minimize the prediction loss:

\bar{x}_i : answers user already submitted

r_{ij} : observed rating of user i for movie j

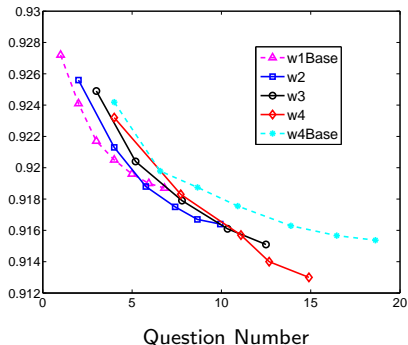
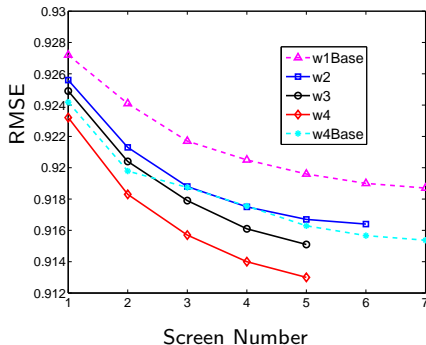
v_j : movie profiles

$$\begin{aligned} \min_{w, Z_L, Z_D, Z_U} & \sum_{i \in L(w)} \sum_{j \in O_i} (r_{ij} - (Z_L \bar{x}_i)^\top v_j)^2 + \sum_{i \in D(w)} \sum_{j \in O_i} (r_{ij} - (Z_D \bar{x}_i)^\top v_j)^2 \\ & + \sum_{i \in U(w)} \sum_{j \in O_i} (r_{ij} - (Z_U \bar{x}_i)^\top v_j)^2 \quad \text{s.t. } \|w\|_0 \leq l \end{aligned}$$

where Z_L, Z_D, Z_U are linear regressors

Prediction Accuracy: Netflix

- w1Base: single question
- w4Base: linear combination of 4 single-question trees



- Multi-questions: better prediction
- Ours: better prediction with fewer questions
- More results in the poster!