Cold Start Collaborative Filtering

Ke Zhou

School of Computational Science & Engineering
College of Computing
Georgia Institute of Technology

Pairwise Preference Regression for Cold-start Recommendation

- Feature-based regression models.
- User demographic information and item content features.

Three Types of Problems

- Recommending existing items for new users
- Recommending new items for existing users
- Recommending new items for new users

	Existing Users	New Users
Existing items	I	II
New items	III	IV

Profile Construction

- Item content: genre, cast, manufacturer, production year.
- Item popularity/quality.
- User profiles: demographical information and aggregated historical behavior.
- User feature $\mathbf{z}$, item feature $\mathbf{x}$.

Bilinear Form for Prediction

- $s_{ui} = x_u^T W z_i = w^T(x_u \otimes z_i)$.

$$\min_w (r_{ui} - s_{ui})^2 + \lambda \|w\|^2$$

- Solution

$$w^\star = \left(\sum_{u,i \in O} z_i z_i^T \otimes x_u x_u^T + \lambda I \right)^{-1} \left(\sum_{ui \in O} z_i \otimes x_u \right)$$

- An alternative loss function

$$\min_w \sum_u \frac{1}{n_u} \sum_{ij} [(r_{ui} - r_{uj}) - (s_{ui} - s_{uj})]^2 + \lambda \|w\|^2$$

Experiments

- Baseline: Most popular, segmented most popular, Vibes affinity
- Measure: $nDCG_k$, $k = 1$.
- training: I; test: II, III, IV.
- For each user, partition items according to their ratings. Random sample one item from each cluster.

Experiments cont.

Cold-start setting	Algorithm	MovieLens		EachMovie	
		$nDCG_1$	STD	$nDCG_1$	STD
Existing item recommendation for new users	Random	0.4163	0.0068	0.4553	0.0055
	MP	0.6808	0.0083	0.6798	0.0166
	SMP	0.6803	0.0078	**0.6868**	0.0146
	Affinity1	0.6800	0.0077	0.6698	0.0134
	Affinity2	0.4548	0.0091	0.5442	0.0154
	Pairwise	**0.6888**	0.0078	0.6853	0.0149
New item recommendation for existing users	Random	0.4158	0.0059	0.4539	0.0052
	Affinity2	0.4489	0.0094	0.5215	0.0149
	Pairwise	**0.4972**	0.0145	**0.5821**	0.0176
New item recommendation for new users	Random	0.4154	0.0065	0.4540	0.0046
	Affinity2	0.4439	0.0102	0.5212	0.0145
	Pairwise	**0.4955**	0.0141	**0.5821**	0.0172

Learning Preferences of New Users in Recommender Systems: An Information Theoretic Approach

- How can we effectively learn preferences of new users so that they can begin receiving accurate personalized recommendations from the system?
- User effort: familiar items
- Recommendation accuracy

Strategies to Learn User Profiles

- Popularity
- Entropy: no information about rating frequency
- Entropy0: taking missing values into account. w_0 the weight for missing values

$$Entropy0(a_t) = -\frac{1}{\sum_i w_i} \sum_{i=0}^{5} p_i w_i \log p_i$$

- HELF: Harmonic mean of entropy and logarithm of frequency

$$HELF = \frac{2LF'H'}{LF' + H'},$$

where $LF' = \frac{\log(a_i)}{\log(\|U\|)}$ and $H' = \frac{\log H}{\log 5}$.

Igcn: Information Gain through Clustered Neighbors

- Adaptive to user's rating history
- Decision Tree

$$IG(a_t; W) = H(C) - \sum_r \frac{w_r |C^r_{a_t}|}{E(C; W)} H(C^r_{a_t}),$$

where $E(C; W) = \sum_r w_r \frac{|C^r_{a_t}|}{|C|}$.

Why Adaptive?

	item 1	item 2	item 3
user 1	5	1	5
user 2	5	1	1
user 3	1	5	1
user 4	1	5	1

- Globally, item 1 and item 2 are both more informative than item 3 in terms of rating entropy
- However, when we known a user rates item 1 with 5 stars, item 3 is more informative than item 2.

Igcn: Information Gain through Clustered Neighbors

Algorithm 3.1: IGCN algorithm

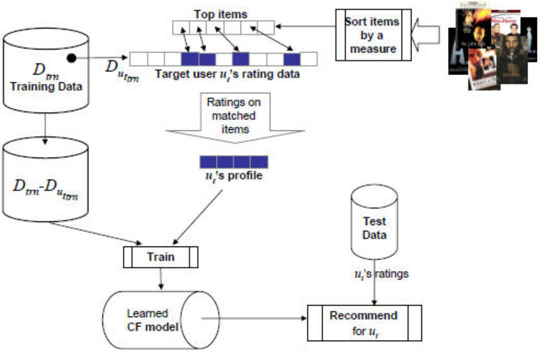
- Create c user clusters
- Compute information gain (IG) of the items
- **Non-personalized step:**
  /* *The first few ratings to build an initial profile* */
  REPEAT
  - Present next top n items ordered by their IG scores
  - Add items the user is able to rate into her profile

  UNTIL the user has rated at least i items
- **Personalized step:**
  /* *Toward creating a richer profile* */
  REPEAT
  - Find best l neighbors based on the profile so far
  - Re-compute IG based on the l users' ratings only
  - Present next top n items ordered by their IG scores
  - Add items the user is able to rate into her profile

  UNTIL best l neighbors do not change

Evaluation Process

- MAE
- Expected Utility: $U(\hat{R}, R) = R - 2|\hat{R} - R|$.



Results

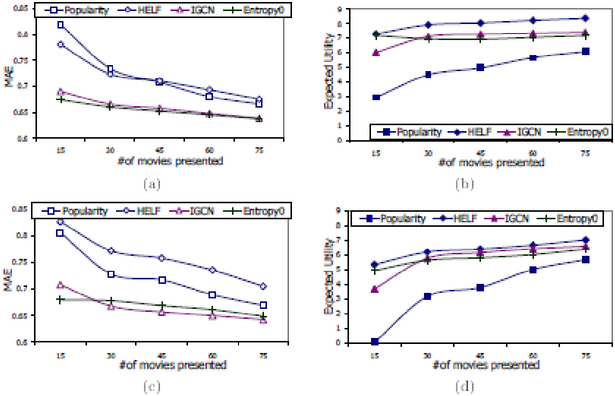


Fig. 6. How effective are the learned user profiles? Recommendation accuracy results from the offline simulations. (a)-(b) on the USER-BASED kNN, and (c)-(d) on the ITEM-BASED kNN CF algorithm. The evaluation metrics used are mean absolute error or MAE (the lower, the better) and expected utility or EU (the higher, the better).