Project 3: Algorithm Implementation and Evaluation

Group 7

Apr, 2018

Model Algorithm

Paper 1, section 2.3

Chose class number from c(2,6,8,10)

1-fold Cross-validation - selected 6 with highest rank score

Used C=6 to evaluate the model, the final rank score is 32.3

- Clean datasets
- EM Algorithm
- Estimate parameters of model using training data
- Estimate parameters of model using training data
- Cross-validation to choose number of classes
- Evaluation

Memory-based

Let $r_{\mathrm{u,m}}$ be the rating by user 'u' for movie 'm' and $\bar{r}_{\mathrm{u}}$ be user u's average rating.

$$\hat{r}_{\mathrm{a,m}} = \bar{r}_{\mathrm{a}} + \frac{\sum_{\mathrm{u} \in \{\mathrm{users}\}} (r_{\mathrm{u,m}} - \bar{r}_{\mathrm{u}}) \times w_{\mathrm{u,a}}}{\sum_{\mathrm{u} \in \{\mathrm{users}\}} w_{\mathrm{u,a}}},$$

- Use the **entire user-item database** to generate predictions based on **active user's neighbors** (those with similar interests)
- **Deviation-from-mean** idea: users may rate **centered around different points**.
    - One user may rate 1 through 5, another may rate 3 through 5
- Similarity weights considered
    - Pearson, Spearman, Vector Similarity, Entropy, Mean Square Difference, SimRank

Results - run time

		MS_sim	MS_pred	MS_pred_norm	movie_sim	movie_pred	movie_pred_norm
Pearson	Time (s)	963.859	323.29	503.818	2627.089	3050.675	4772.046
Spearman	Time (s)	2221.998	392.926	483.97	3430.973	3076.679	4791.602
Cosine (VS)	Time (s)	637.659	372.509	474.867	2129.846	3270.824	4742.013
Entropy	Time (s)	14220.546	347.793	490.684	23212.994	3125.486	5645.941
MSD	Time (s)	592.686	400.468	510.579	1990.68	3035.799	4894.69

- Movie data takes much longer to train and predict than Microsoft due to data and UI matrix size
- Before normalization, Pearson is time-efficient to run for both Microsoft and movie data
- After normalization, Cosine is most time-efficient run for both
- Entropy took very long time to train similarity matrix, followed by Spearman

Simrank: structural-context similarity

Paper 4

- We compute a measure that says “two objects are similar if they are related to similar objects”
- Based on a simple and intuitive graph-theoretic model
- s(a, b) : similarity between objects a, b

$$s(a,b) = \begin{cases} 1, \ a = b, \\ 0, \text{ if } I(a) = \emptyset \text{ or } I(b) = \emptyset, \\ \dfrac{c}{|I(a)||I(b)|} \displaystyle\sum_{i=1}^{|I(a)|} \sum_{j=1}^{|I(b)|} s(I_i(a), I_j(b)), \text{ otherwise,} \end{cases}$$

- ❑ C: decay factor between 0 and 1
- ❑ I(a), I(b): sets of **in-neighbors** of vertices a, b

In other words, s(a, b) = **Average** similarity between in-neighbors of a, **I(a)** and in-neighbors of b, **I(b)**

Computing Simrank: Iterative process

$$R_0(a,b) = \begin{cases} 1, & a = b, \\ 0, & \text{otherwise}, \end{cases} \qquad (1)$$

$$R_{k+1}(a,b) = \frac{C}{|I(a)||I(b)|} \sum_{i=1}^{|I(a)|} \sum_{j=1}^{|I(b)|} R_k(I_i(a), I_j(b)) \qquad (2)$$

$R_k(a, b)$ converges to s(a, b)

(2) written in matrix form:

$$S_{k+1} = c\underline{W^T S_k W} - c\,\mathrm{diag}(\underline{W^T S_k W}) + I,\ S_0 = I$$

W is the adjacency matrix A normalized by columns $W = AD^{-1}$

$$D_{i,i} = \sum_{j=1}^{n} A_{i,j}$$

Enhancement: rating normalization

Paper 2, sec 7

An extension to the GroupLens algorithm is to account for the differences in spread between users' rating distributions by converting ratings to z-scores, and computing a weighted average of the z-scores (Equation 6).

$$p_{a,i} = \overline{r}_a + \sigma_a * \frac{\sum_{u=1}^{n} \boxed{\frac{r_{u,i} - \overline{r}_u}{\sigma_u}} * w_{a,u}}{\sum_{u=1}^{n} w_{a,u}} \qquad (6)$$

Evaluation

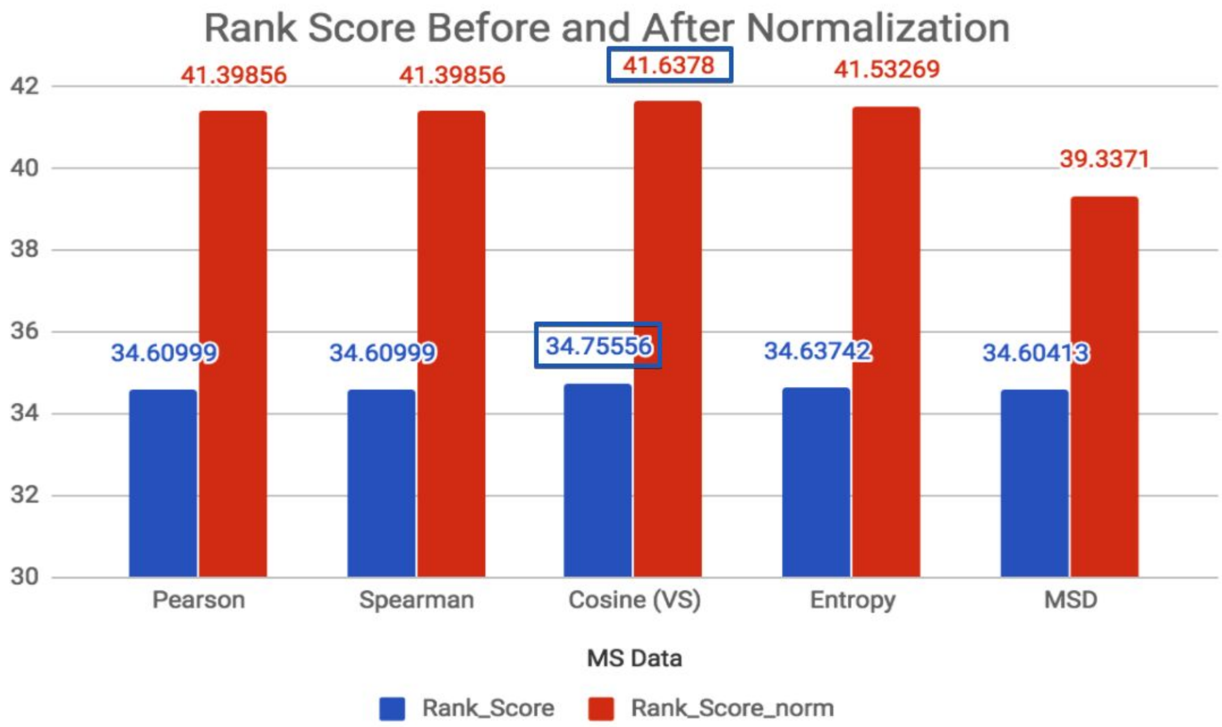
We used Ranked Score to evaluate Microsoft's web data for implicit voting, MAE and ROC on movie data for explicit voting.

- Mean Absolute Error: MAE is used when algorithm gives the user a rating indicating potential interest in a topic (movie data)

$$MAE = \frac{\sum_{(\mathrm{u,i}) \in \mathrm{test}} |\hat{r}_{\mathrm{u,i}} - r_{\mathrm{u,i}}|}{|\mathrm{test}|}$$

- Ranked Score: used when algorithm gives the user an ordered list of recommendation. Estimate expected utility of a particular ranked list to a user. (web data)

MS data evaluation



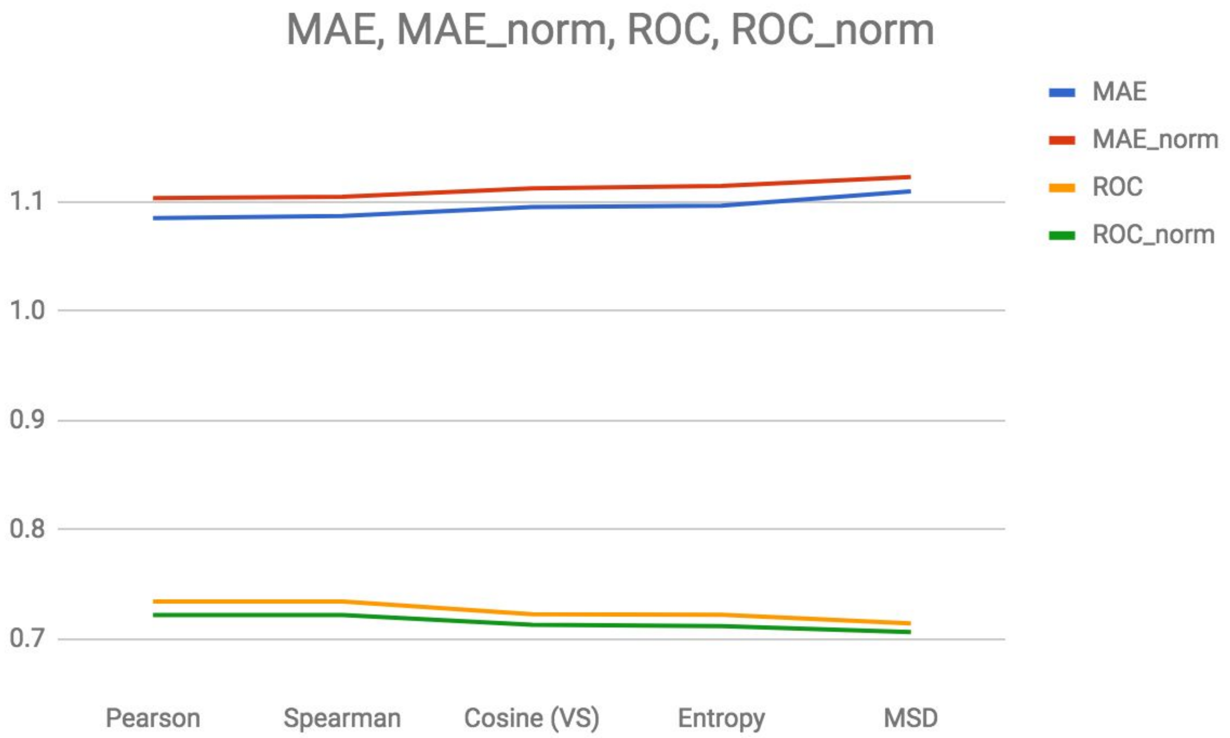
Results - MS data

MS Data	MS_sim (s)	MS_pred (s)	MS_pred_norm (s)	Rank_Score	Rank_Score_norm
Pearson	963.859	323.29	503.818	34.60999	41.39856
Spearman	2221.998	392.926	483.97	34.60999	41.39856
Cosine (VS)	→ 637.659	372.509	474.867	34.75556	41.6378
Entropy	→ 14220.546	347.793	490.684	34.63742	41.53269
MSD	592.686	400.468	510.579	34.60413	39.3371

For Microsoft data:

- Cosine and Entropy performs better than the other weights, but Entropy took too long to train similarity matrix.
- MSD took least amount of time to train but also performs worse than others
- Normalization improved performance

Movie data evaluation



Results - movie data

Movie Data	movie_sim (s)	movie_pred (s)	movie_pred_norm (s)	MAE	ROC	MAE_norm	ROC_norm
Pearson	2627.089	3050.675	4772.046	1.085059	0.7342	1.103532	0.7217
Spearman	3430.973	3076.679	4791.602	1.087122	0.7341	1.104699	0.7217
Cosine (VS)	2129.846	3270.824	4742.013	1.095303	0.7226	1.112472	0.7129
Entropy	23212.994	3125.486	5645.941	1.096575	0.7219	1.114704	0.7116
MSD	1990.68	3035.799	4894.69	1.10973	0.714	1.122731	0.7062

For movie data:

- Pearson and Spearman perform better than the other weights, processing time is relatively fast too
- Entropy took longest to train, and the performance is average
- MSD took least amount of time to train but also performs worse than others
- Normalization didn't improve performance

Collaborative Filtering challenges

Sparse - few rated values in User-Item matrix

Cold start - difficult for new users, items

Scale - need to scale well with millions of users and items

Speed - predictions need to be fast

Synonyms - different name same thing

Increase diversity - can the algorithm discover new items?

Reference

Paper in class

https://arxiv.org/pdf/1410.0717.pdf

https://openproceedings.org/2010/conf/edbt/LiHHJSYW10.pdf