Group 4

Collaborative Filtering Algorithms

Group Member: Du Guo
Yu Tong
Xiuruo Yan
Lan Wen
Yuhan Zha

CONTENTS

1. Data Sets
2. Memory Based Algorithm
3. Model Based Algorithm
4. Evaluation
5. Result Analysis

Data Set Info

Data Set 1: Anonymous Microsoft Web Data

Binary (0 for not visited & 1 for visited)

User: User indices range from 10010 to 42708

Feature (web): Site indices ranging from 1000 to 1295

Train: 4151* 85

Test: 665 * 85

Data Set 2: Movies Grading Data

6 Categories (movie rating from 1 to 6)

User: User indices range from 1 to 74418

Feature(Movie):Movie indices range from 1 to 1648

Train: 5055 * 1619

Test: 5055 * 1597

Implementation: Variants being tested

Algorithm	Component	Variants	Data
Memory-based Algorithm	Similarity Weight	Pearson Correlation	1,2
		Mean Square Difference	1,2
		SimRank	1
	Variance Weighting	No	1,2
	Selecting Neighbors	Weight Threshold	1,2
		Best-n-estimator	1,2
		Combined	1,2
	Rating Normalization	Deviation for Mean	1,2
		Z-score	1,2
Model-based Algorithm	Cluster Models		1

Similarity Weight:

Weight all users with respect to similarity with the active user:

- **Pearson Correlation**

Measures the degree to which a linear relationship exists between two variables

$$w_{a,u} = \frac{\sum_{i=1}^{m}(r_{a,i} - \overline{r}_a) * (r_{u,i} - \overline{r}_u)}{\sigma_a * \sigma_u}$$

- **Mean Square Difference**

$$MSD_{a,u} = \frac{\sum_h (r_a - r_u)^2}{n}$$

$$w_{a,u} = \frac{max(MSD_{a,u}) - MSD_{a,u}}{max(MSD_{a.u})}$$

SimRank

For the algorithm of SimRank, we mainly implement these two equations below

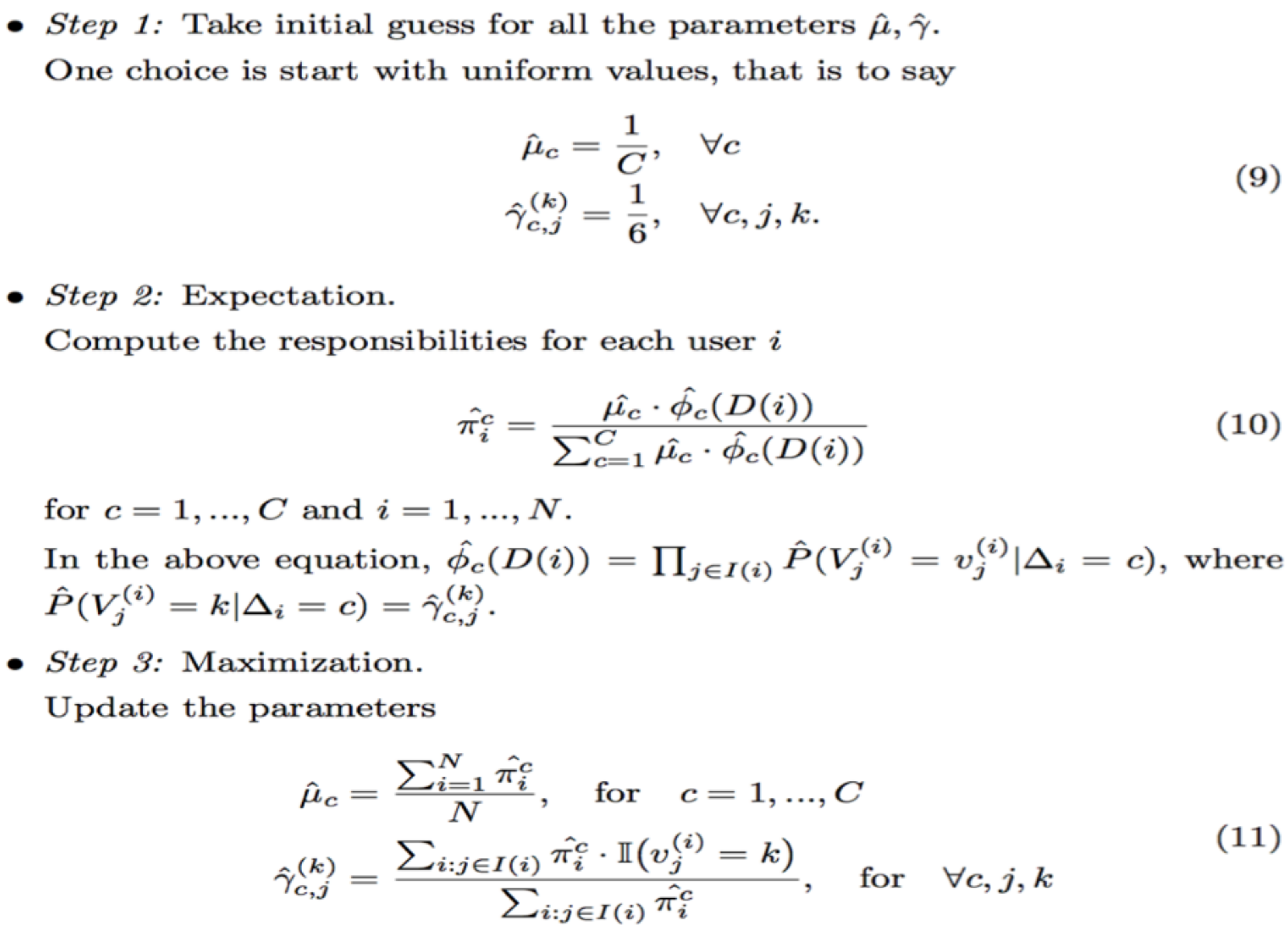
$$s(c,d) = \frac{C_2}{|I(c)||I(d)|} \sum_{i=1}^{|I(c)|} \sum_{j=1}^{|I(d)|} s(I_i(c), I_j(d))$$

$$s(A,B) = \frac{C_1}{|O(A)||O(B)|} \sum_{i=1}^{|O(A)|} \sum_{j=1}^{|O(B)|} s(O_i(A), O_j(B))$$

The way to compute SimRank is the naïve method:

$$R_0(a,b) = \begin{cases} 0 & (\text{if} \quad a \neq b) \\ 1 & (\text{if} \quad a = b) \end{cases} \qquad R_{k+1}(a,b) = \frac{C}{|I(a)||I(b)|} \sum_{i=1}^{|I(a)|} \sum_{j=1}^{|I(b)|} R_k(I_i(a), I_j(b))$$

Model Based Algorithm – Cluster Model



- *Step 1:* Take initial guess for all the parameters $\hat{\mu}, \hat{\gamma}$.

  One choice is start with uniform values, that is to say

$$\begin{aligned} \hat{\mu}_c &= \frac{1}{C}, \quad \forall c \\ \hat{\gamma}_{c,j}^{(k)} &= \frac{1}{6}, \quad \forall c, j, k. \end{aligned} \tag{9}$$

- *Step 2:* Expectation.

  Compute the responsibilities for each user i

$$\hat{\pi_i^c} = \frac{\hat{\mu_c} \cdot \hat{\phi}_c(D(i))}{\sum_{c=1}^{C} \hat{\mu_c} \cdot \hat{\phi}_c(D(i))} \tag{10}$$

  for $c = 1, ..., C$ and $i = 1, ..., N$.

  In the above equation, $\hat{\phi}_c(D(i)) = \prod_{j \in I(i)} \hat{P}(V_j^{(i)} = v_j^{(i)} | \Delta_i = c)$, where $\hat{P}(V_j^{(i)} = k | \Delta_i = c) = \hat{\gamma}_{c,j}^{(k)}$.

- *Step 3:* Maximization.

  Update the parameters

$$\begin{aligned} \hat{\mu}_c &= \frac{\sum_{i=1}^{N} \hat{\pi_i^c}}{N}, \quad \text{for} \quad c = 1, ..., C \\ \hat{\gamma}_{c,j}^{(k)} &= \frac{\sum_{i:j \in I(i)} \hat{\pi_i^c} \cdot \mathbb{I}(v_j^{(i)} = k)}{\sum_{i:j \in I(i)} \hat{\pi_i^c}}, \quad \text{for} \quad \forall c, j, k \end{aligned} \tag{11}$$

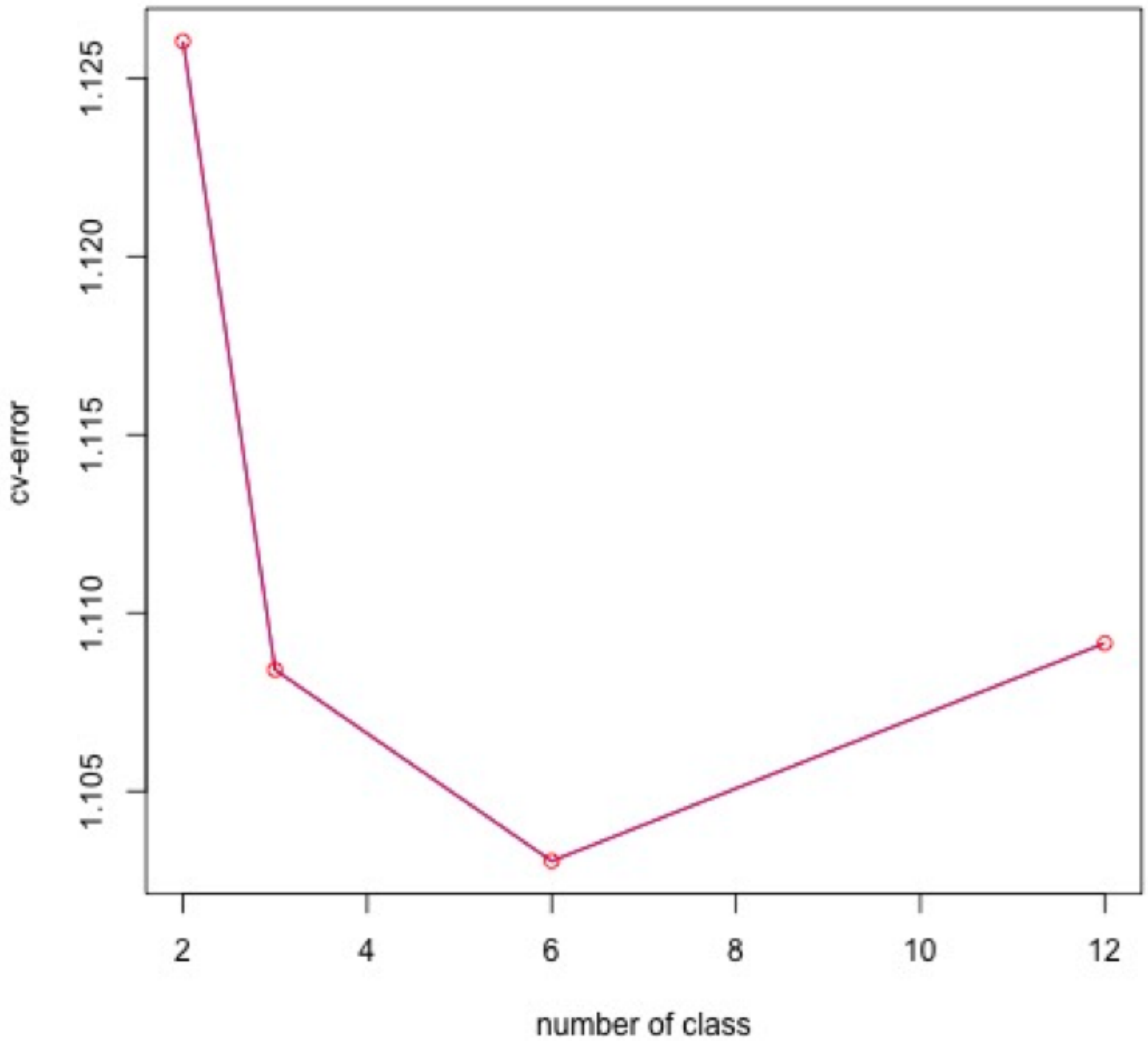
For initialization,

Set parameter γ random variable which sum by class is 1.

For iteration,

We set the iteration as 100; There is also another stop sign which is

$|\pi_i - \pi_{i-1}| \leq 0.05$


cv-error
1.125
1.120
1.115
1.110
1.105
2
4
6
8
10
12
number of class

Rating Normalization

- **Deviation for mean**

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

- **Z-score:**

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

Evaluation

For this part, we applied three evaluation methods to compare the results:

For dataset 1 – Rank Score

- Rank Score

❖ It measures the expected utility of a ranked recommendation list Ra and normalized by the maximum achievable utility Ra_max.

❖ The expected utility of a list is simply the probability of viewing a recommanded item times its utility.

For dataset 2 – MAE

- MAE

❖ MAE is simply to calculate the average absolute deviation for a user :

$$S_a = \frac{1}{m_a} \sum_{j \in P_a} |p_{a,j} - v_{a,j}|$$

Results Analysis

Dataset 1: Ranked Scoring					
Selecting Nbors Similarity Weight	Weight Threshold		Best-N-Estimator		Combined
	0.05	0.2	20	40	0.05 + 40
Pearson Correlation	45.7274	44.05147	32.68916	34.62473	33.7009
Mean-Squared-Difference	47.77998	47.77998	33.38771	35.12559	35.04042
SimRank	45.985	32.1453	32.3775	33.39524	35.0523

Dataset 2: MAE

Selecting Nbors Similarity Weight	Weight Threshold		Best-N-Estimator		Combined
	0.05	0.2	20	40	0.05 + 40
Pearson Correlation	1.187658	1.185793	1.167949	1.167326	1.183761
Mean-Squared-Difference	1.168654	1.168654	1.170718	1.169313	1.166298

THANK YOU