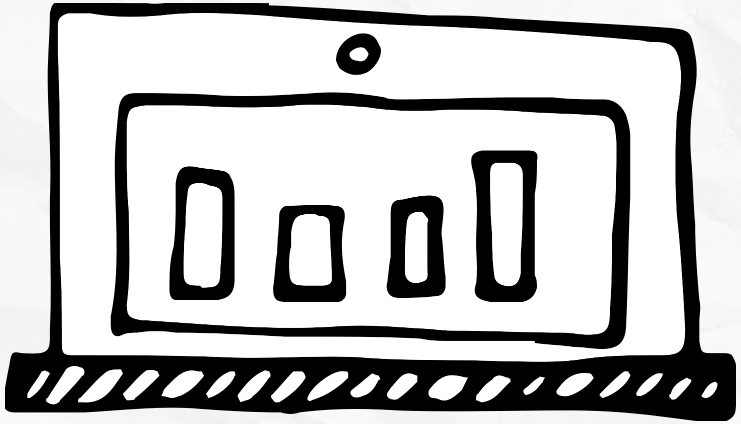
ADS Group1

Collaborate Filtering

Team Members:
Cao, Xinlei
Guo, Xiaoxiao
Li, Xinrou
Utomo, Michael
Wang, Jingyi

2018.04.18

CONTENTS



PART 01

About Dataset

- The first data set is Anonymous Microsoft Web Data. It is an implicit voting data, with each vroot characterized as being visited (vote of one) or not (no vote).
- The second data set is EachMovie. It is an explicit voting data, with votes ranging in value from 1 to 6.

Data Processing

Microsoft Web Data
(Implicit Voting)

	Web 1	Web 2	…	Web m
User 1	0	1		1
User 2	1	0		0
…				
User n	1	1		0

1 for visited
0 for not visited

EachMovie Data
(Explicit Voting)

	Movie 1	Movie 2	…	Movie m
User 1	5	NA		3
User 2	NA	4		1
…				
User n	6	2		2

1-6 for ratings
NA for not rated

PART 02

Memory-Based Algorithm

- Goal
- Framework

Goal

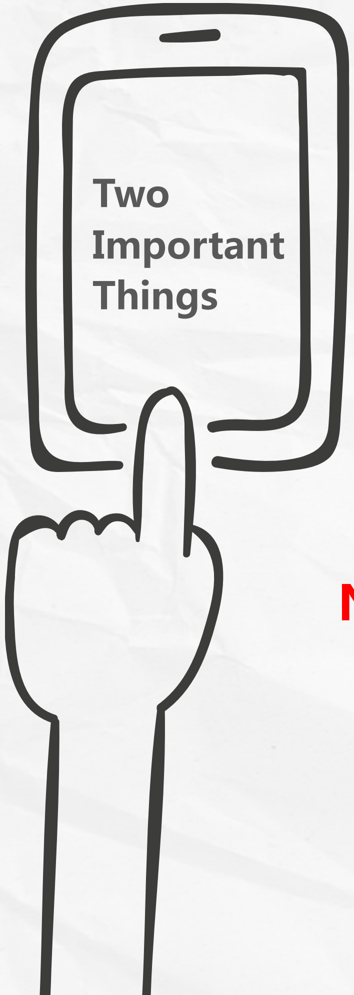
$$p_{a,i} = \bar{r}_a + \frac{\sum_{u=1}^{n}(r_{u,i} - \bar{r}_u) * w_{a,u}}{\sum_{u=1}^{n} w_{a,u}}$$

Make prediction by performing a weighted average of deviations from the neighbors' mean

Two important things here:

- Weight ($w_{a,u}$) : the similarity weight between the active user a and neighbor u
- n : the number of neighbors

Memory-Based Algorithm



Weight ($w_{a,u}$)

Number of neighbors (n)

Component	Variants Tested
Similarity Weight	• Pearson Correlation • Vector Similarity • SimRank
Variance Weighting	• No • Yes
Selecting Neighbors	Weight Threshold

Similarity Weight

For movie data, we set ratings larger than 4 as "like" and smaller than or equal to 4 as "dislike"

Pearson

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

Vector

The cosine of the angle formed by two vectors (normalized inner product of two vectors)

SimRank

Two objects are similar if they are related to similar objects

$$w(a,i) = \frac{\sum_j (v_{a,j} - \overline{v}_a)(v_{i,j} - \overline{v}_i)}{\sqrt{\sum_j (v_{a,j} - \overline{v}_a)^2 \sum_j (v_{i,j} - \overline{v}_i)^2}}$$

$$w(a,i) = \sum_j \frac{v_{a,j}}{\sqrt{\sum_{k \in I_a} v_{a,k}^2}} \frac{v_{i,j}}{\sqrt{\sum_{k \in I_i} v_{i,k}^2}}$$

$$\begin{cases} \mathbf{S}^{(0)} = \mathbf{I}_n \\ \mathbf{S}^{(k+1)} = \left(c \cdot \mathbf{Q} \cdot \mathbf{S}^{(k)} \cdot \mathbf{Q}^T\right) \bigvee \mathbf{I}_n \quad (\forall k = 0, 1, \cdots) \end{cases}$$

Variance Weighting

Idea:

Knowing a user’ s rating on distinguishing (high variance) items is more valuable than others in discerning a user’ s interest.

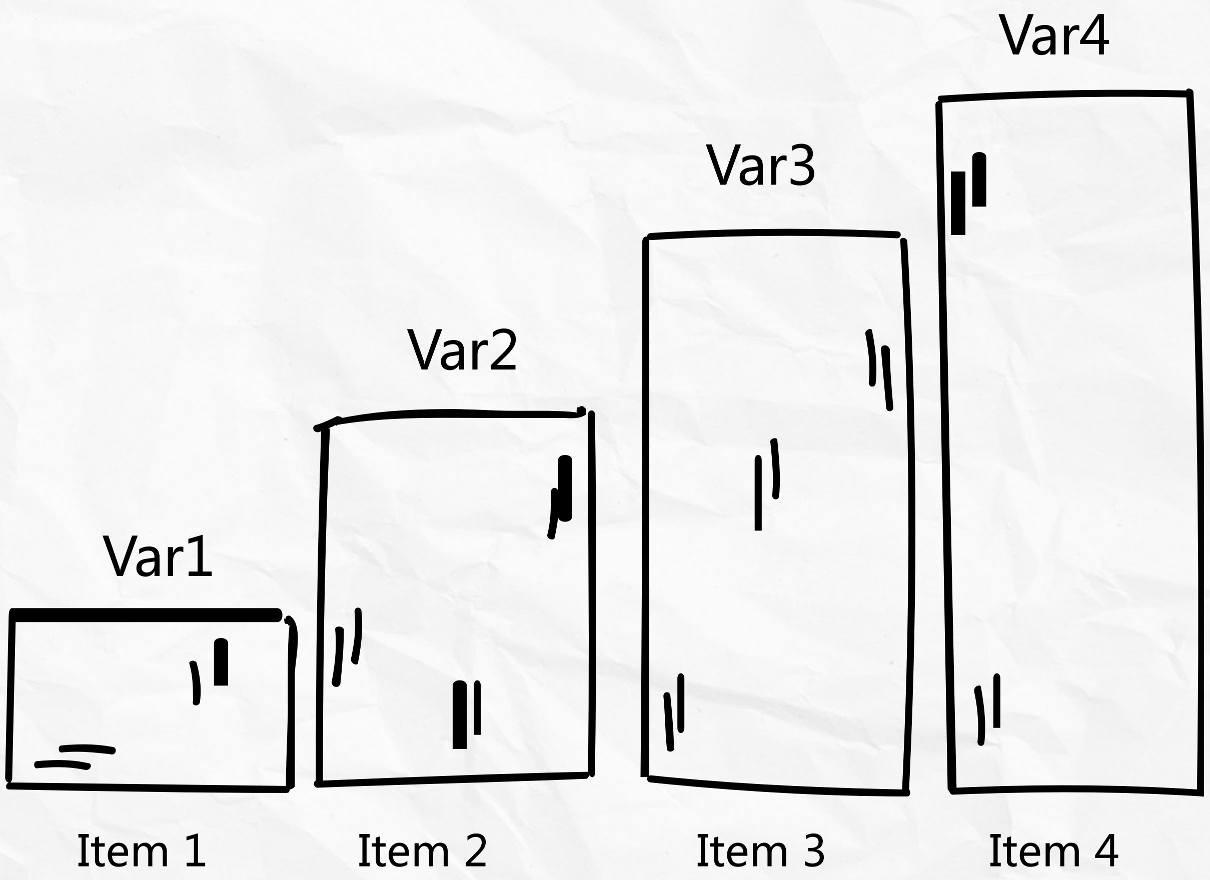
Implement:

$$w_{a,u} = \frac{\sum_{i=1}^{m} v_i * z_{a,i} * z_{u,i}}{\sum_{i=1}^{m} v_i}$$

$$v_i = \frac{var_i - var_{min}}{var_{max}}$$

Where Z stands for the scaled rating

By incorporating a variance weight term, we will increase the influence of items with high variance in ratings and decrease the influence of items with low variance.



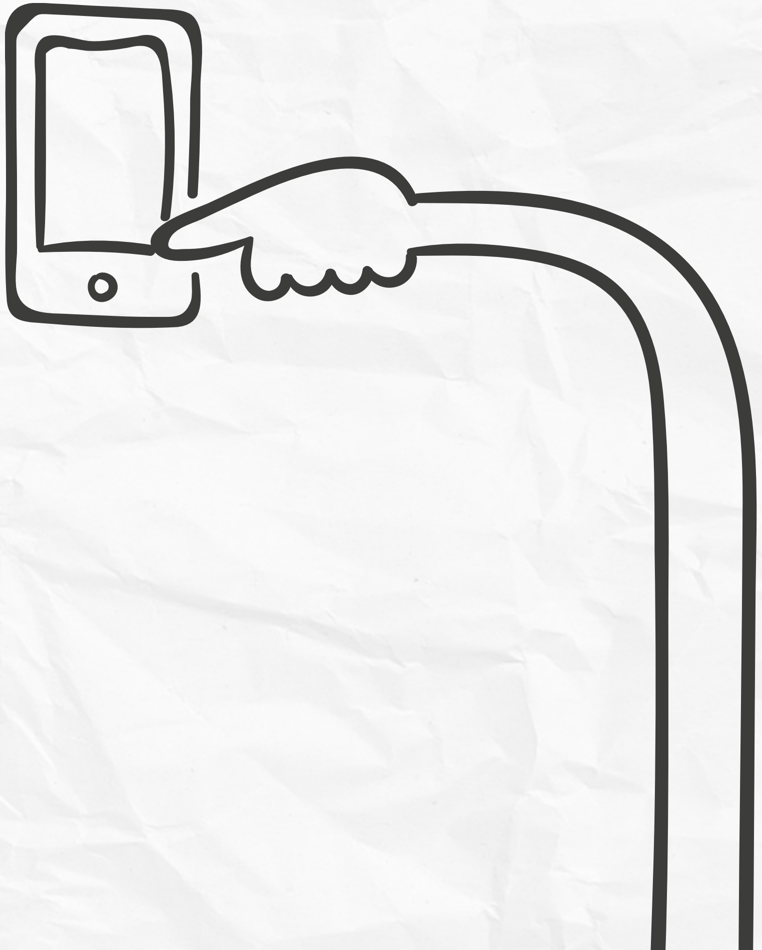
Neighbors Selection

Weight Threshold

Idea:

Users with High correlates can be exceptionally more valuable as predictors than those with low correlations. So we choose users with relatively high correlates (larger than a given threshold) as our predictors.

However, the larger the threshold, the smaller the prediction coverage.



PART 03
Model-Based Algorithm

- Cluster Models
- EM Algorithm

Cluster Models

Assumptions:

Users in the same cluster have the same voting behavior. That is, users in the same cluster have the same probability to give one specific item one specific score.

Goal:

Estimate $$\mathbb{E}[V_b^{(i)} | v_j^{(i)}, j \in I(i)] = \sum_{k=1} k \cdot P\big(V_b^{(i)} = k | v_j^{(i)}, j \in I(i)\big)$$

Estimate parameters by maximize the log-likelihood function

$$\sum_{i=1}^{N} \log \Big[\sum_{c=1}^{C} \mu_c \cdot \prod_{j \in I(i)} P(V_j^{(i)} = v_j^{(i)} | \Delta_i = c)\Big]$$

EM Algorithm

Use EM Algorithm to find the maximum of the log-likelihood function:

$$\sum_{i=1}^{N} \log \left[\sum_{c=1}^{C} \mu_c \cdot \prod_{j \in I(i)} P(V_j^{(i)} = v_j^{(i)} | \Delta_i = c) \right]$$

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

  To avoid local optima, don't use uniform initial values.

- *Step 2:* Expectation.

  Compute the responsibilities for each user i

$$\hat{\pi}_i^c = \frac{\hat{\mu}_c \cdot \hat{\phi}_c(D(i))}{\sum_{c=1}^{C} \hat{\mu}_c \cdot \hat{\phi}_c(D(i))} \tag{9}$$

  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

$$\begin{aligned} \hat{\mu}_c &= \frac{\sum_{i=1}^{N} \hat{\pi}_i^c}{N}, \quad \text{for} \quad c = 1, ..., C \\ \hat{\gamma}_{c,j}^{(k)} &= \frac{\sum_{i: j \in I(i)} \hat{\pi}_i^c \cdot \mathbb{I}(v_j^{(i)} = k)}{\sum_{i: j \in I(i)} \hat{\pi}_i^c}, \quad \text{for} \quad \forall c, j, k \end{aligned} \tag{10}$$

  where $\mathbb{I}(\cdot)$ is the indicator function taking values in $\{0, 1\}$.

  The idea is this step is that in order to calculate MLE in a weighted multinomial distribution, we only need to take the weighted frequency for each class.

- *Step 4:* Iteration.

  Iterate steps 2 and 3 until convergence.

PART 04
Evaluation and Results

- For dataset 1: Rank Score
- For dataset 2: MAE

Metrics

Ranked Score

Estimate the expected utility of a particular ranked list to a user

$$R_a = \sum_j \frac{\max(v_{a,j} - d, 0)}{2^{(j-1)/(\alpha-1)}}$$

$$R = 100 \frac{\sum_a R_a}{\sum_a R_a^{max}}$$

MAE

Mean Absolute Error

mean(abs(prediction - true))

Performance

Memory-Based Algorithm

Microsoft Web Data
(Implicit Voting)

Ranked Score	Pearson	Vector
None	35.75975	36.30890
Variance	45.73053	

EachMovie Data
(Explicit Voting)

MAE	Pearson	Vector	SimRank
None	1.091662	1.069133	1.051652
Variance	1.268181		

Results:

Similarity Weight:

- MS Data: Vector similarity performs better than Pearson correlation. Maybe because the entries here are 0 or 1, and thus hard to show linear relationships.
- EachMovie Data: SimRank outperforms the others, but the difference among them is only very slight.

Variance Weighting:

- MS Data: Pearson correlation with variance weighting performs better than without as we expected.
- EachMovie Data: Applying variance weighting to Pearson correlation decreases the prediction accuracy. One reason may be the variance weighting method ignores the fact that a user who disagrees with the popular feeling provides a lot of information.

Performance

Memory-Based Algorithm VS Model-Based Algorithm

Microsoft Web Data

Ranked Score	Pearson	Vector
None	35.75975	36.30890
Variance	45.73053	

Number of Clusters	Ranked Score
3	46.36221
5	41.30776
7	45.98717
9	40.55384

Results:

The highest ranked score of our cluster model is 46.36221 with 3 clusters, which is higher than the best performance of our Memory-based models.

Thus, the cluster models may outperform the memory-based models for MS Data.

THANK YOU