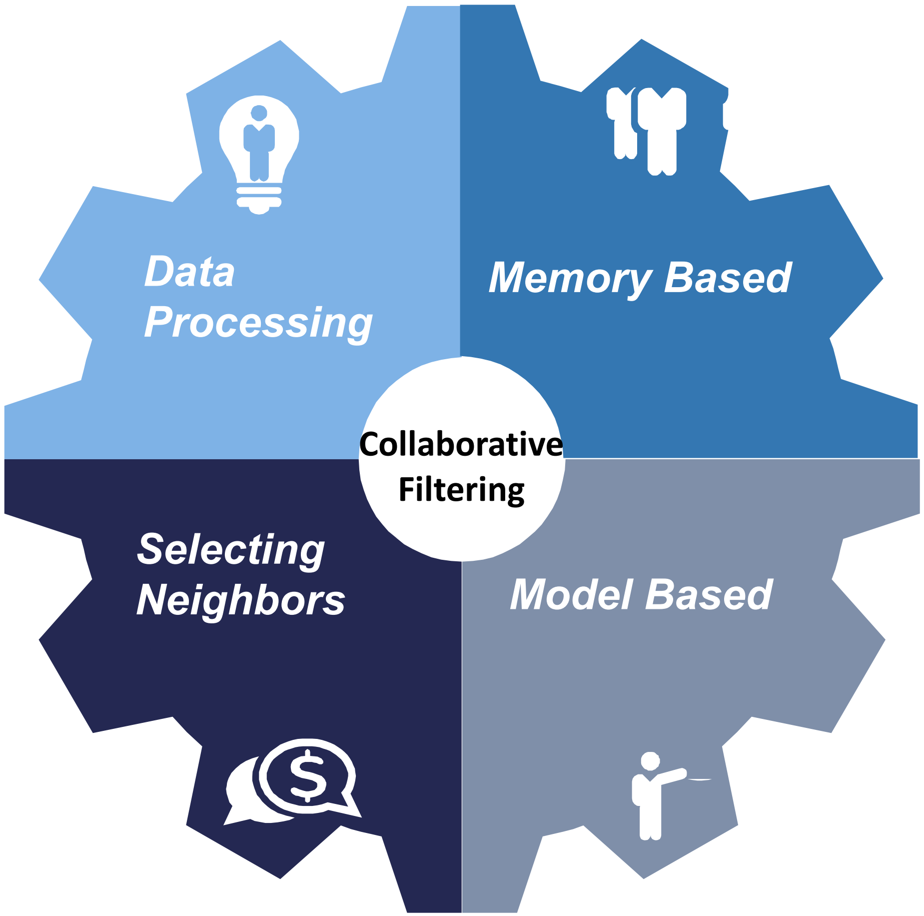
Collaborative Filtering

Group 4:

Mingyue Kong
Nicole Alyse Smith
Noah Chasek-Macfoy
Judy Jinhui Cheng
Yun Li

Table of Contents

1. Project Introduction



Web Data

Train: 4151* 269

Test: 665 * 269

Movie Data

Train: 5055 * 1619

Test: 5055 * 1597

2. Memory Based Performance

1 Pearson
2 Spearman
3 Vector(cosine)
4 Mean Square
5 SimRank

Similarity Weighting Comparison:

		Pearson	Spearman	Vector	Mean Square	SimRank
Rank Score	**MS**	37.5723	37.5723	37.8173	**45625.77?***	/
MAE	**Movie**	1.085	1.085	1.095303	**326.54?***	1.0497
Run time	**MS**	0.76h	0.70h	0.62h	2.7h	/
	Movie	1.75h	2.28h	1.5h	(1000 rows) 1.5h	(1000 rows) over 8h

****: need further analysis***

3.1 Selecting Neighbors

Idea: which other users' data to use in the prediction for a user.

Neighboring improves

(1) accuracy, as high correlates can be exceptionally more valuable as predictors, and

(2) computational time, as commercial collaborative filtering has millions of users and cannot consider all possible combinations of users.

3.2 Selecting Neighbors

There are two methods for selecting neighbors:

Correlation-thresholding: Set an absolute threshold, where all neighbors with correlations above a given threshold are selected.

Advantages: A high threshold produces high correlates

Disadvantages: A high threshold gives you a small neighborhood that limits prediction coverage for many items.

Top-n-neighbors: Pick the best *n* correlates for a given *n*.

Advantages: Choosing *n* does not limit prediction coverage, whereas the threshold approach does.

Disadvantages: High *n* will result in noise(many low correlates) for users who have high correlates; picking a low *n* will result in poor prediction for users who do not have many high correlates.

3.3 Selecting Neighbors

Results

movie data top n neighbors	
n	computational time
10%	1963.017
30%	2303.331
50%	2757.009
70%	3178.991

movie data thresholding	
threshold	computational time
0.9	525.02
0.7	1087.584
0.5	1244.405
0.2	2296.728

MS data top n neighbors	
n	computational time
10%	224.677
30%	275.706
50%	302.482
70%	350.525

MS data thresholding	
threshold	computational time
0.8	27.608
0.6	77.137
0.4	113.58
0.2	212.585

Future considerations:

We could combine the two methods, i.e. combining a low threshold with a high n or a high threshold with a low n.

4.1 Model Based Process

1	2	3	4	5
Step 1	Step 2	Step 3	Step 4	Step 5
Review mathematics behind EM Algorithm	Initialize parameters	Write expectation step of EM algorithm	Write maximization step of EM algorithm	Create hard assignments

4.2 Model Based Performance

• We had to initialize at random points to get any movement between iterations in EM algorithm

• Variation between clusters for predicted ratings for movie j start out very small (0.009655562 in one case after 3 iterations) and diverge over time

• Distribution of the users over the clusters also diverges over time, after 3 iterations we have ~64% of all users in a single cluster

• In 4 iterations the L-2 norm of change between iterations went from ~3.5 to ~.2, it must have some kind of exponential decay convergence if it takes so long to get from .2 to .01

• Fun fact: apparently you can subset cells of a matrix with a 2 col matrix of coordinates in the extract brackets

• Graph ideas: convergence over time, histogram of predicted ratings

4.3 Details of Model Based Performance

Model Results

•Tau: L-2 norm of difference between soft assignment matrices from two consecutive iterations.

•We used a threshold of .01 for convergence…

•Movie data never reached convergence threshold, possibly because it has so many more dimensions than that MS data.

•Thus, tau is likely not directly comparable between data sets.

4.3 Details of Model Based Performance

Movie Data - max iterations: 750; clusters: 12

Final Tau: 0.0115

•Total iterations: 750

•Elapsed Time: 8.16 hours !!

•Cluster Distribution: 88% in group 11, 10% in group 4… the rest are below 1%

•MAE Score: 1.107312

4.3 Details of Model Based Performance

Movie Data - max iterations: 750; clusters: 12

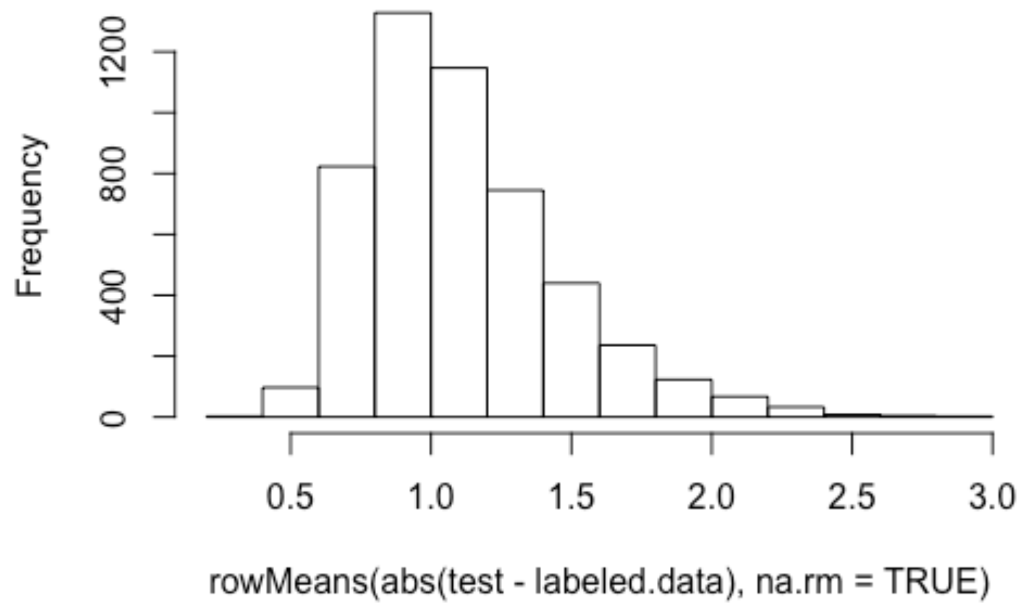
<u>Notes:</u>

- There was an bug in the EM clustering function when we trained this model; we were calculating gamma (probability of user i rating movie j with rating k) by dividing by all movies rather than all movies rated by user i.

- 88% of users in group 11 corresponded to mu (probability of being in cluster c) of 14% for group 11. How do we interpret this?

4.3 Details of Model Based Performance

Histogram of individual user MAE scores



4.3 Details of Model Based Performance

Movie Data - max iterations: 100; clusters: 10

Final Tau: 0.02726874

Total iterations: 99

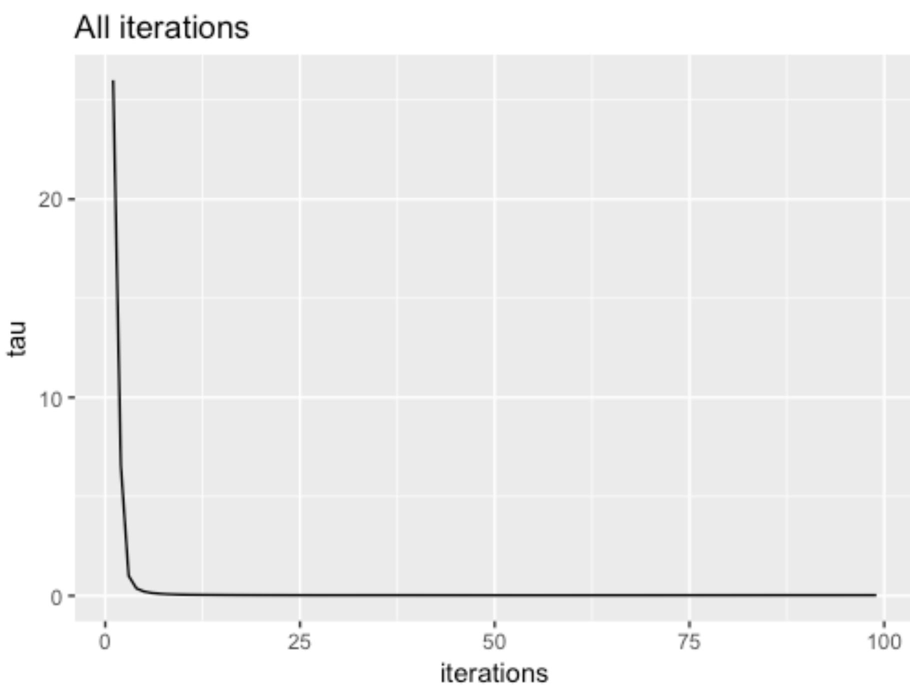
Elapsed Time: lost to history, but approx. 1 hr

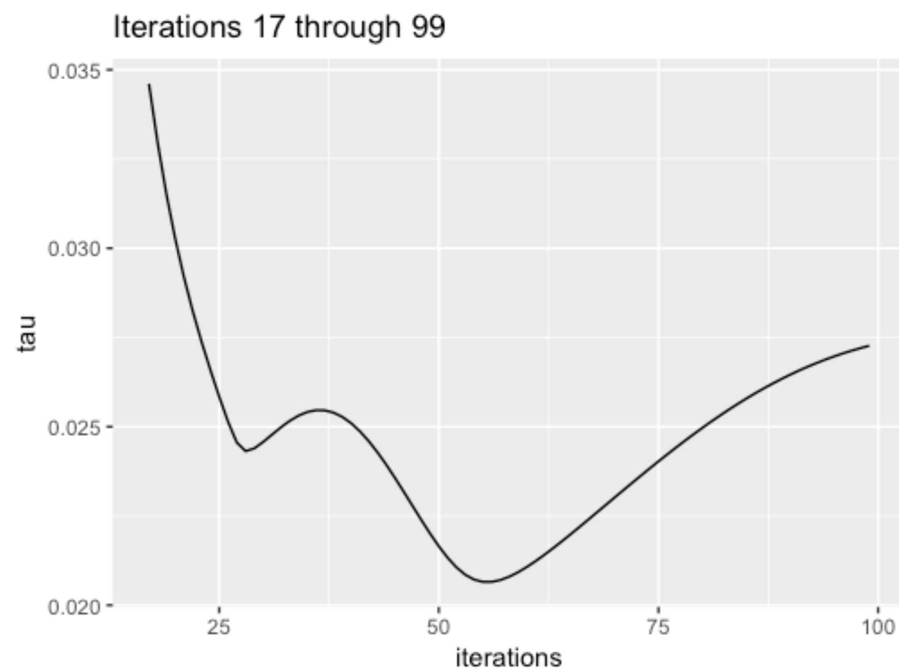
Distribution over clusters: 96% in cluster 2

MAE Score: 1.107282

4.3 Details of Model Based Performance

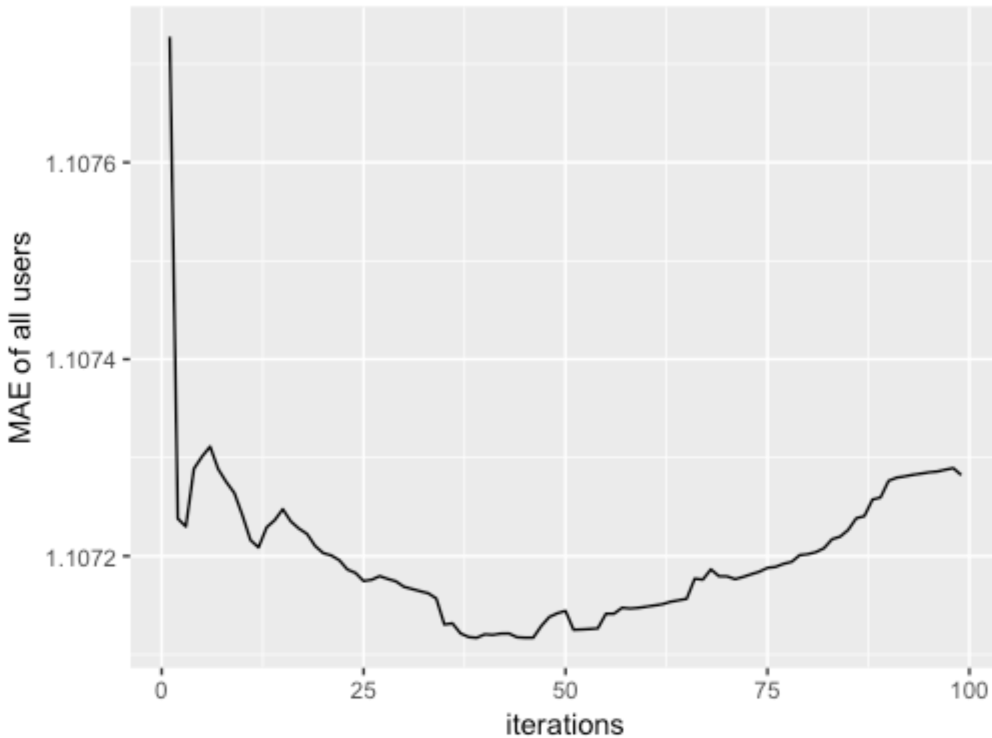
Tau convergence over time





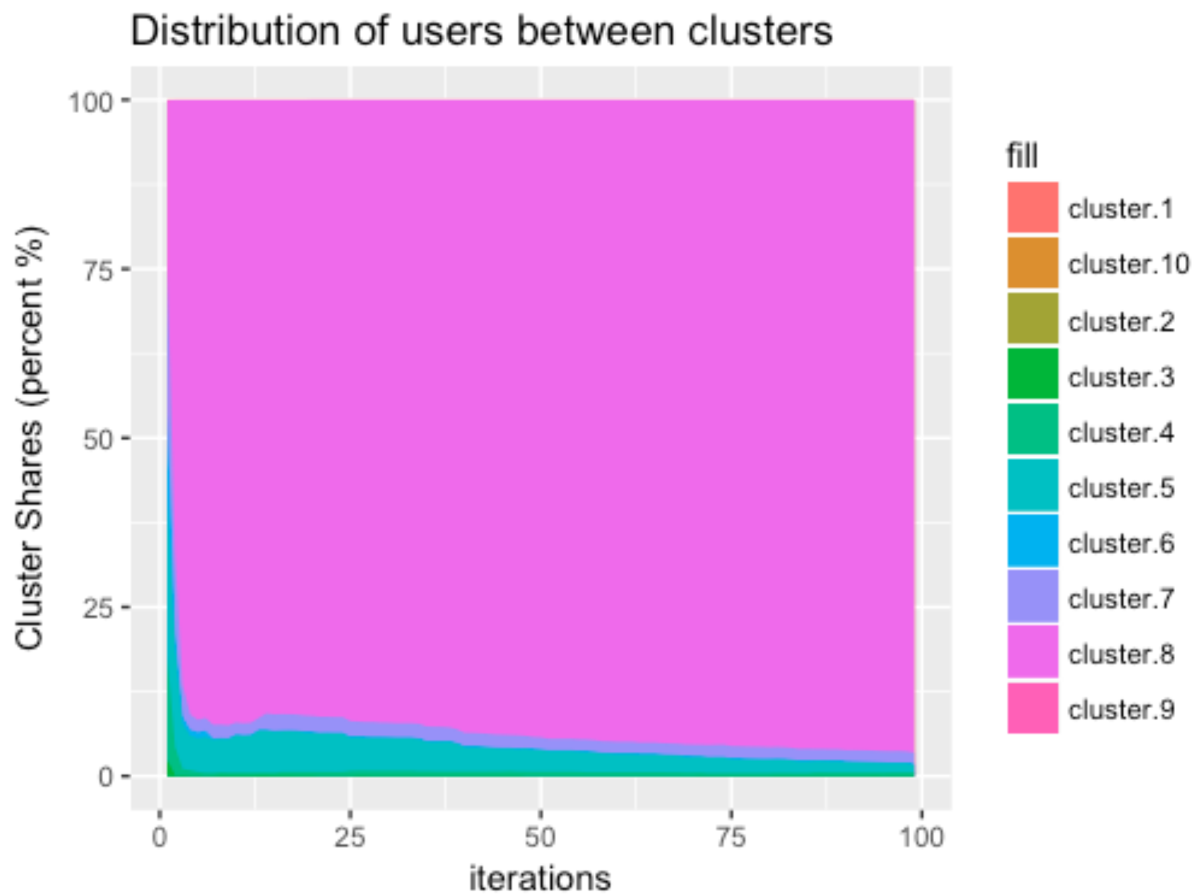
4.3 Details of Model Based Performance

MAE over time

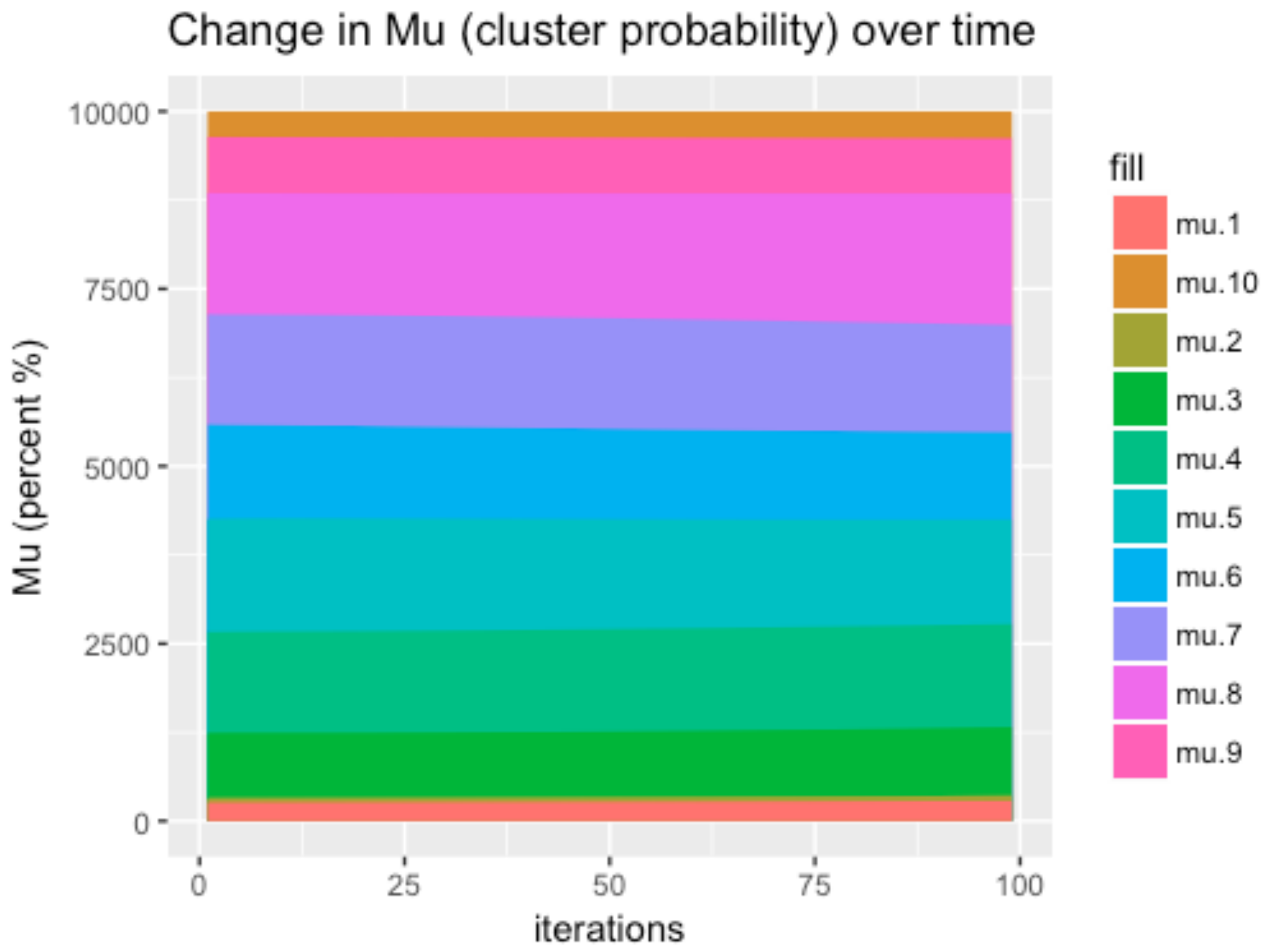


4.3 Details of Model Based Performance

Cluster Distribution over time



4.3 Details of Model Based Performance



4.3 Details of Model Based Performance

Virtually no improvement or even variation in MAE over time. Never below 1.

Tau improves early on, but is virtually constant soon after

Cluster distribution suggests our algorithm is optimizing to cluster every user in a single group

Compare tau convergence and cluster distribution over time

Virtually no movement in Mu

Something is not right here…

4.3 Details of Model Based Performance

MS Data - max iterations: 700; clusters: 10

Final Tau: 1.151118e-13

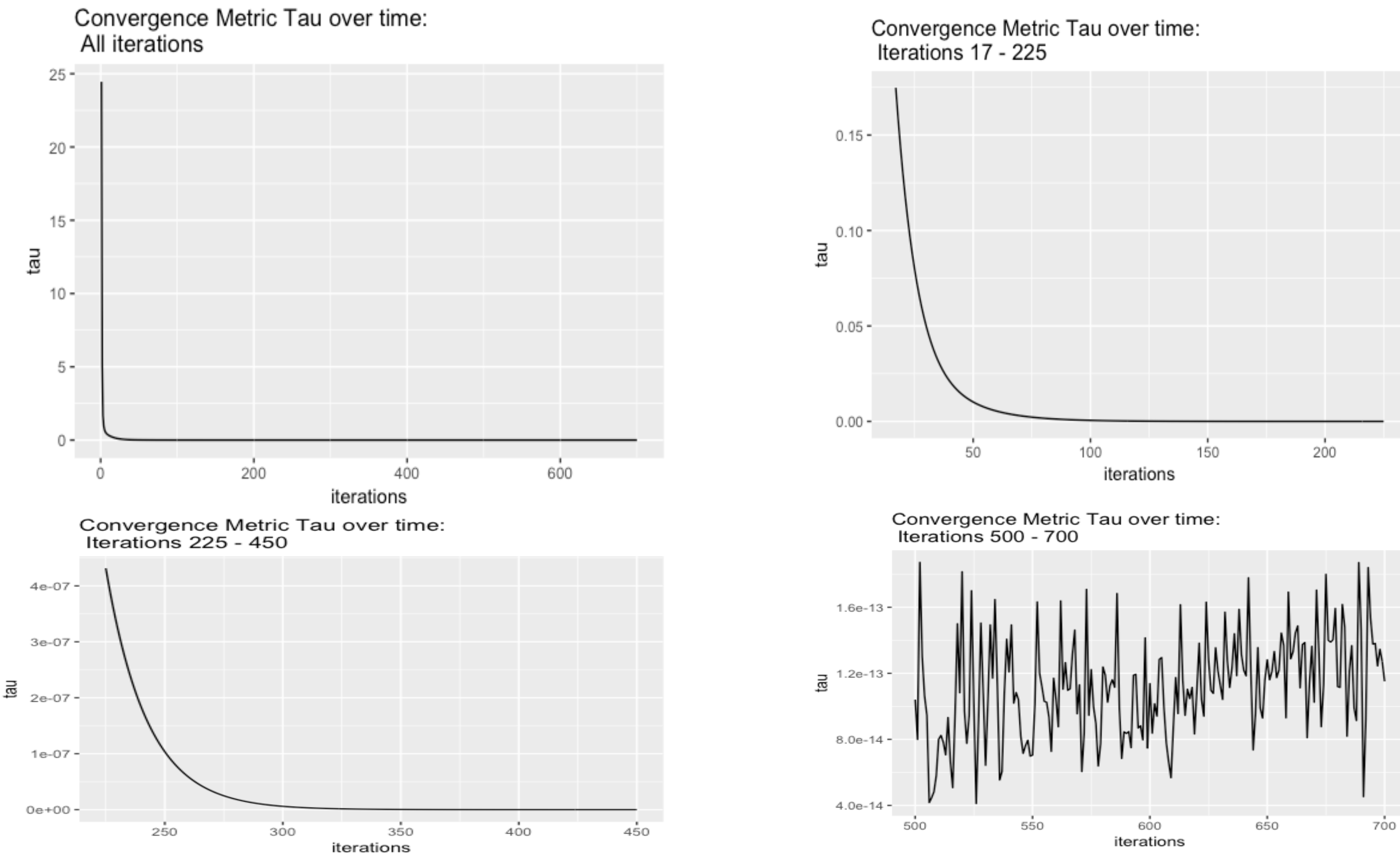
Total iterations: 700

Elapsed Time: 0.34 hours

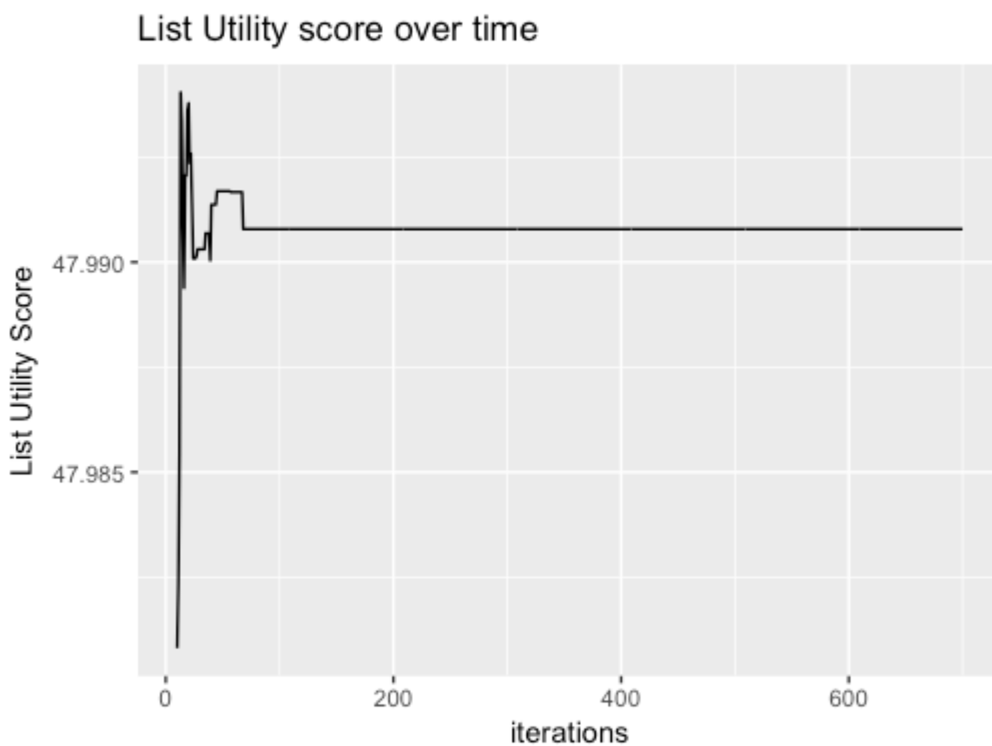
Distribution over clusters: 92% in group 6

Expected Utility Score: 47.99079

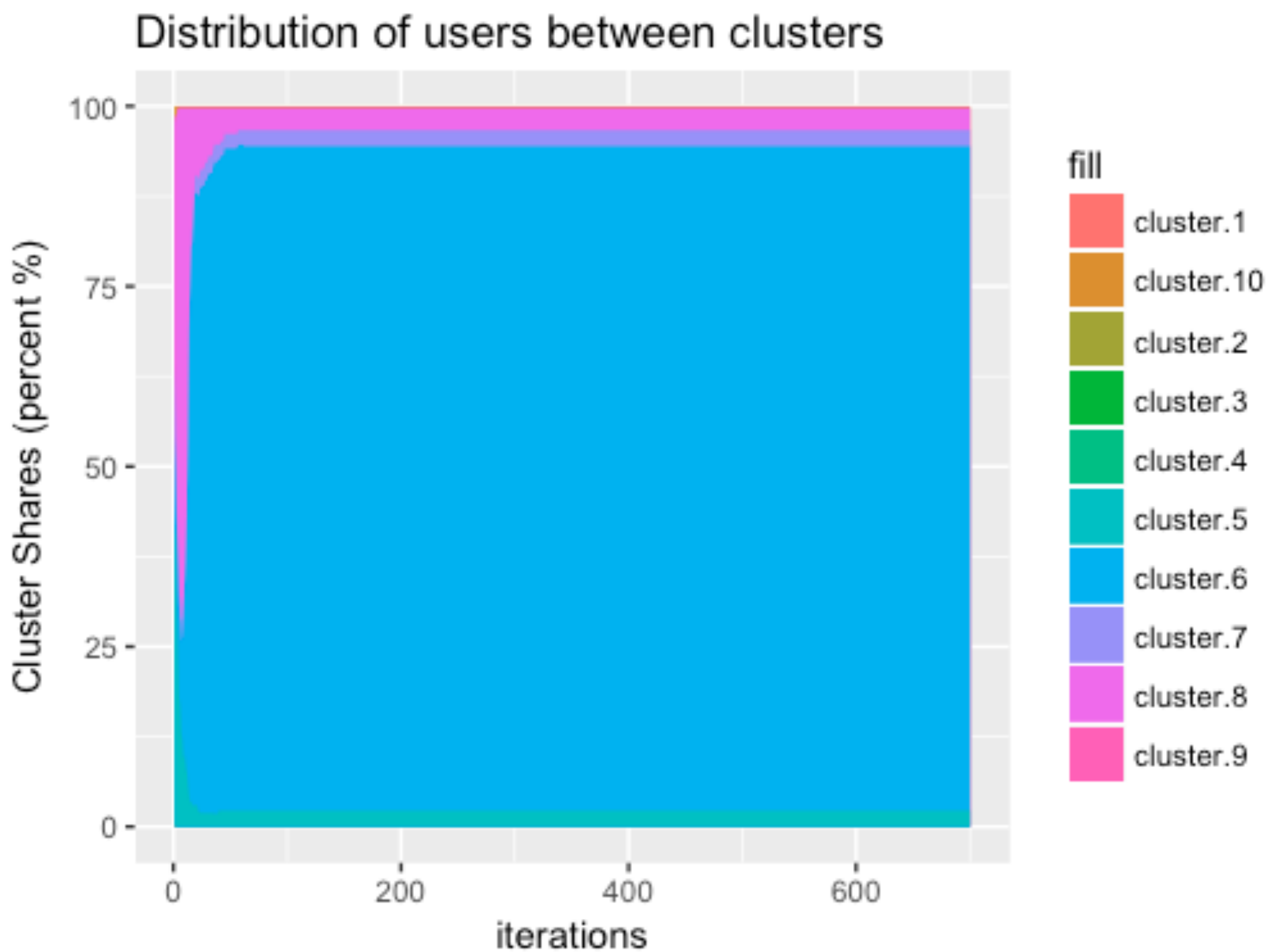
4.3 Details of Model Based Performance



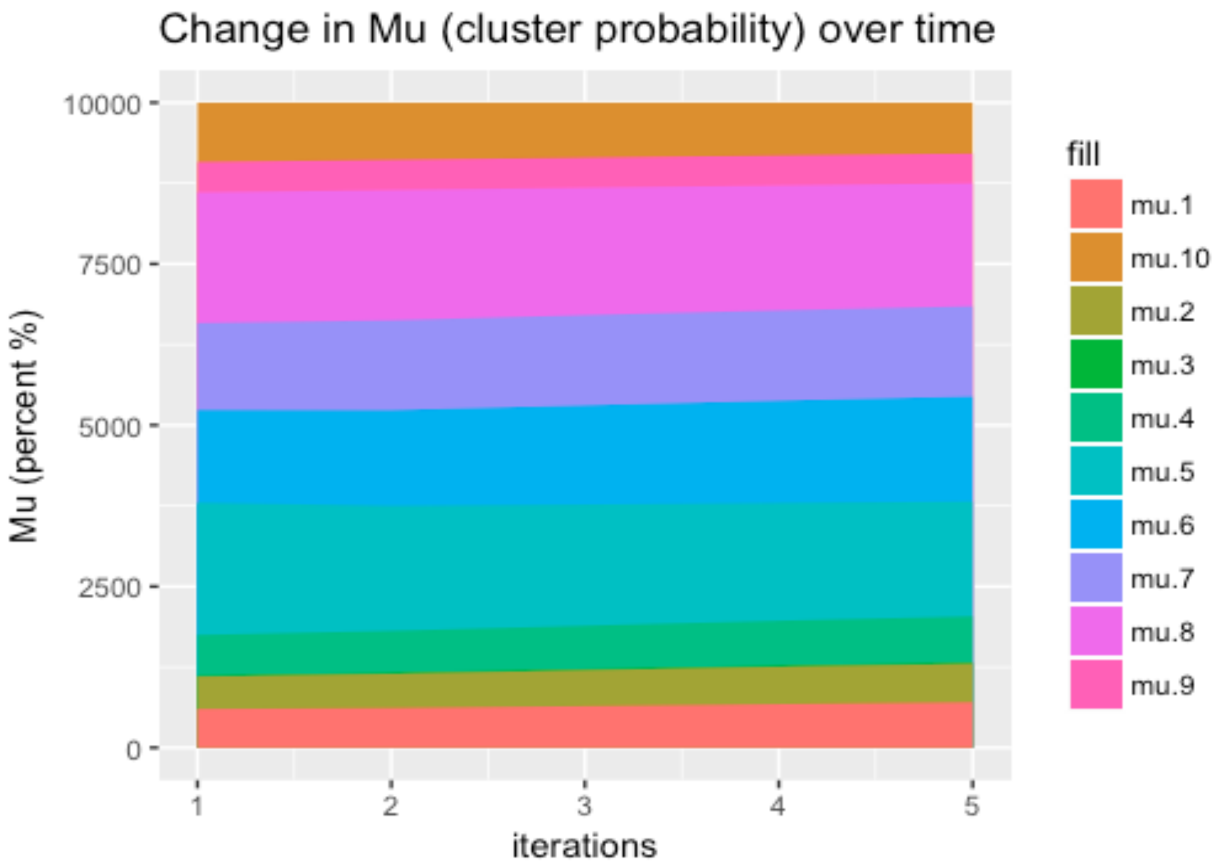
4.3 Details of Model Based Performance



4.3 Details of Model Based Performance



4.3 Details of Model Based Performance



4.3 Details of Model Based Performance

MS Data - max iterations: 700; clusters: 10

Much, much faster computationally than movie model training. Probably because of the fewer users, items, and rating levels.

The fact that both data sets grouped most users into the same cluster suggests the problem is with our algorithm

Pretty much no convergence after 200 iterations

The entire range of List Utility scores is between 47.77052 and 47.99538. It seemingly randomly starts at the low end and stabilizes within 30 iterations.

What does it mean that tau continued to converge after the Utility score stabilized?

Why does Mu stay pretty much unchanged through the iterations, but the cluster distributions converge?

5. Leftovers

Process

- We spent a lot of understanding the math behind collaborative filtering.

Improvements

- Model-based: Minimize number of loops in the EM algorithm to improve computational time.
- Model-based: Use cross-validation.

Thank you!

Group 4:

Mingyue Kong
Nicole Alyse Smith
Noah Chasek-Macfoy
Judy Jinhui Cheng
Yun Li