

Master Thesis
Recommender Systems Comparison

Vasileios Symeonidis

27-05-2017

Contents

| | | |
|-----------|---|------------|
| I | Master Thesis | i |
| 1 | Intro | i |
| 2 | Collaborative filtering | i |
| 2.1 | Content based | i |
| 2.2 | Latent Factors | i |
| 3 | Our Experiment | ii |
| 3.1 | Infrastructure | ii |
| 3.1.1 | Apache Spark | ii |
| 3.2 | Dataset | ii |
| 3.3 | Metrics | ii |
| 3.3.1 | Mean Absolute Error | ii |
| 3.3.2 | Execution Time | ii |
| 4 | Results | ii |
| 5 | Conclusion | iii |
| 6 | References | iv |
| II | Appendices | v |
| A | Code used | vi |
| A.1 | User Based Collaborative Filtering | vi |
| A.2 | Product Based Collaborative Filtering | vi |
| A.3 | Latent Factors | vi |
| A.4 | infra code | vi |
| B | Metrics | vi |
| B.1 | What is the mean absolute error | vi |
| B.2 | Time | vi |

Part I

Master Thesis

1 Intro

This is the introduction for this master thesis. Why we need recommendation systems? Retailers can propose the right product to the right target group. User get advertisements they may be interested in.[2]

History, what has been tried so far?

2 Collaborative filtering

What is collaborative filtering. [2]

2.1 Content based

In content based recommender systems we try to recommend based on features we know. There are two types of content based recommender systems. On the one hand we have the user based recommendation. This recommendation is done by trying to match users profiles, in order to find which item the user i might like. But in real world we don't have the needed information to make the recommendation to the user. On the other hand we have the item-product based recommendation, in this case we are trying to find user that might like the given product. This is much easier due to the fact that you know more about a product than a user, and you can classify them easily.

In this case we have a matrix R that contains the rates given by users to items. This matrix most of the times will be low in density, this is because each user does not rate each product. The second matrix we come across is the M . This matrix contains all the movies with their genres. Each characteristic is binary. For example, the movie with id i is both action and comedy and none of the other genres.

$$w = R^{-1}M^T \quad (1)$$

Normalized

$$w = (\lambda I + R^T R)^{-1} R^T M \quad (2)$$

2.2 Latent Factors

In latent factors recommender systems we follow a similar approach but, in case of ALS(Alternating least squares), we are trying to find metrics that may lead us to the correct recommendation. Those metrics are not distinct, and

may change in a number of iterations. Those metrics are inducted from the R matrix as we define it above. This makes this approach more tolerant to missing values, or wrong quality measures. Thus this metric as will be presented below is more efficient on prediction and time.

ALS explanation.

$$w = (\lambda I + X^T X)^{-1} X^T Y$$

Figure 1: **Equation**

3 Our Experiment

3.1 Infrastructure

3.1.1 Apache Spark

What is map reduce

How spark differentiates from its predecessors, hadoop yarn

Spark lightweight in memory data transformation Resilient Distributed Datasets (RDDs)

mllibs

add spark jira note

broadcasting rdd

//cite the mastering apache spark book [1]

a Spark cluster to be created on AWS EC2 storage.

New trends on spark <https://github.com/apache-spark-on-k8s/spark> cite this repository too.

3.2 Dataset

What is the dataset about. This dataset contains users, movies and the rating user made about the movies. This dataset is split to multiple subsets of 80000 training sets and respective 20000 reviews. [3]

3.3 Metrics

3.3.1 Mean Absolute Error

$$MAE = \frac{\sum_{i=1}^n |y_i - x_i|}{n} = \frac{\sum_{i=1}^n \sqrt{(y_i - x_i)^2}}{n} \quad (3)$$

3.3.2 Execution Time

Time is measured in milliseconds.

4 Results

Table 1: **Content Based Algorithm Results**

| Training Dataset | Testing Dataset | Mean Absolute Error | Execution time (ms) |
|------------------|-----------------|---------------------|---------------------|
| u1.base | u1.test | 1.6467431428213226 | 30514 |
| u2.base | u2.test | 1.6055222166704628 | 27714 |
| u3.base | u3.test | 1.608925907479106 | 27164 |
| u4.base | u4.test | 1.6259192043203685 | 26687 |
| u5.base | u5.test | 1.6284658627202895 | 27124 |
| ua.base | ua.test | 1.6425364580036836 | 26640 |
| ub.base | ub.test | 1.6357196576385744 | 26861 |

Table 2: **Latent Factors Algorithm Results**

| Training Dataset | Testing Dataset | Mean Absolute Error | Execution time (ms) |
|------------------|-----------------|---------------------|---------------------|
| u1.base | u1.test | 1.1818684937209607 | 10195 |
| u2.base | u2.test | 1.1800652808093945 | 6517 |
| u3.base | u3.test | 1.1783366748334452 | 5377 |
| u4.base | u4.test | 1.1730543877181654 | 5433 |
| u5.base | u5.test | 1.1686585291940668 | 5217 |
| ua.base | ua.test | 1.2008035300836668 | 5214 |
| ub.base | ub.test | 1.2134460078406009 | 5083 |

husjdfslfdkjsafkij

jhsdkfhksjhfkshf

5 Conclusion

As a conclusion we can see that als is better on both metrics from the content based.

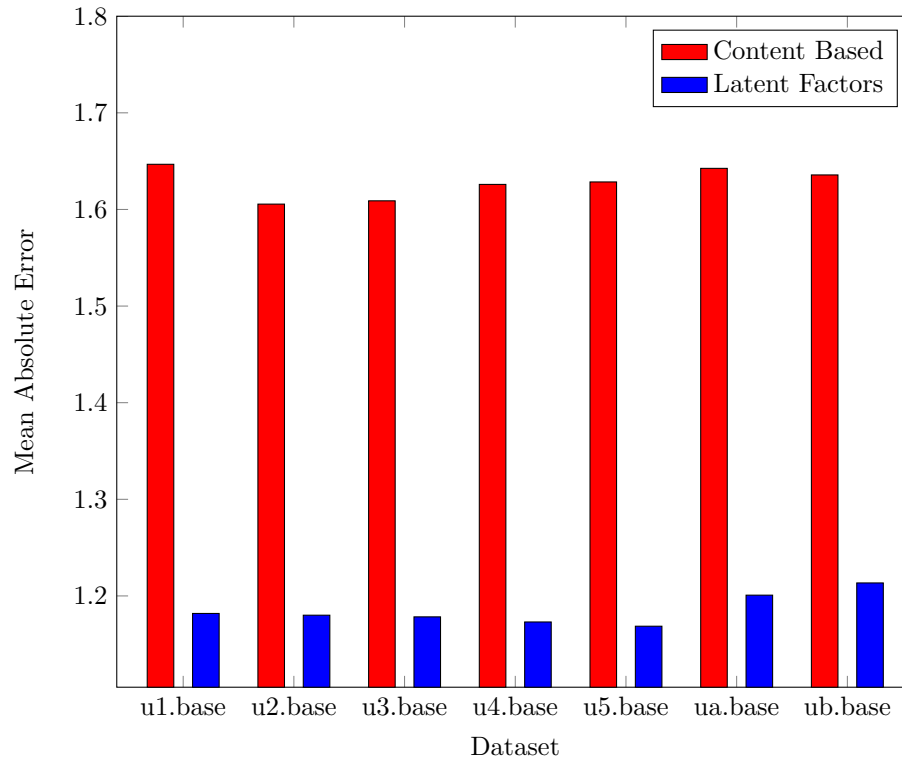


Figure 2: **Latent Factors vs Content Based on Mean Absolute Value**

6 References

References

- [1] “Apache Spark lightning-fast cluster computing.” <https://spark.apache.org/>. Accessed: 2017-05-21.
- [2] P. Melville and V. Sindhvani, “Recommender systems,” *Encyclopedia of Machine Learning and Data Mining*, pp. 1056–1066, 2017.
- [3] “MovieLens grouplens.” <https://grouplens.org/datasets/movielens/>. Accessed: 2017-05-22.

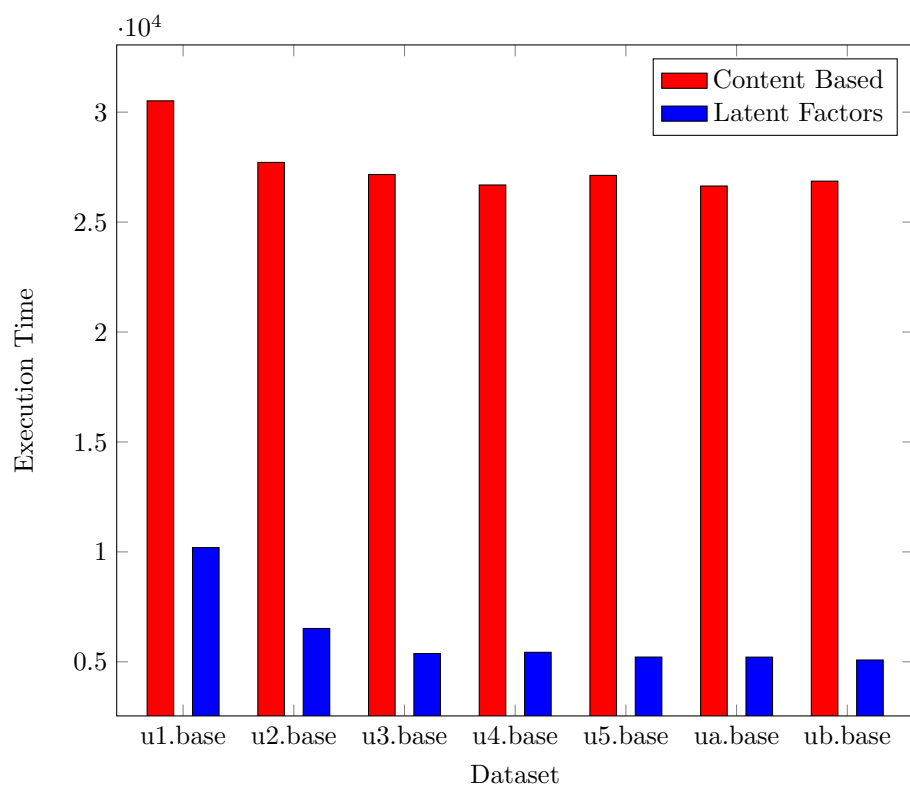


Figure 3: **Latent Factors vs Content Based on Execution Time**

Part II

Appendices

A Code used

A.1 User Based Collaborative Filtering

A.2 Product Based Collaborative Filtering

A.3 Latent Factors

A.4 infra code

B Metrics

B.1 What is the mean absolute error

B.2 Time

List of Tables

| | | |
|---|--|-----|
| 1 | Content Based Algorithm Results | iii |
| 2 | Latent Factors Algorithm Results | iii |

List of Figures

| | | |
|---|--|----|
| 1 | Equation | ii |
| 2 | Latent Factors vs Content Based on Mean Absolute Value | iv |
| 3 | Latent Factors vs Content Based on Execution Time . . | v |