# **Assignment 3**

### Authors:

|  |  |
| --- | --- |
| **Name** | **ID** |
| Sangi Manish Rao | 2015A7PS0235H |
| Tanuj Gupta | 2015A7PS0159H |
| Krishna Bharadwaj | 2015A7PS0076H |
| Ashrith Grandi | 2015A7PS0285H |

### Code Files:

1. cur.py
2. cur\_energy.py
3. svd.py
4. svd\_energy.py
5. collab\_filtering.py
6. collab\_filtering\_baseline.py

### Working:

Described in README file

### Language:

Python

### Packages Used:

#### Timeit

#### Numpy

#### Math

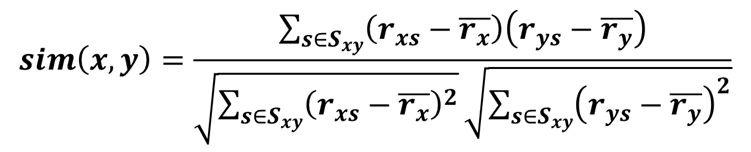
### Project Description:

In this project we have implemented three major recommender systems which are Collaborative Systems, SVD and CUR along with some modifications.

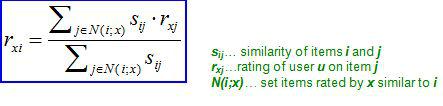
#### Collaborative Systems:

Pearson​​ correlation​​ coefficient​​ was ​​used​ ​for ​​similarity​​ measure.

Formula:

****

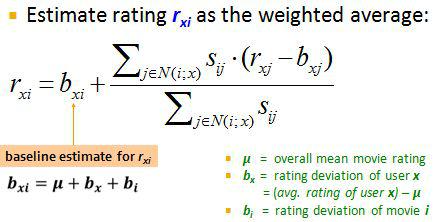
User-User ​​collaborative​​ filtering​​ is ​​used.The​​ following ​​ formula​ ​is ​​used​​ to ​​predict ​a ​​user’s ​​movie​​ rating.

****

Baseline​​ estimate​​ Collaborative-​​filtering:

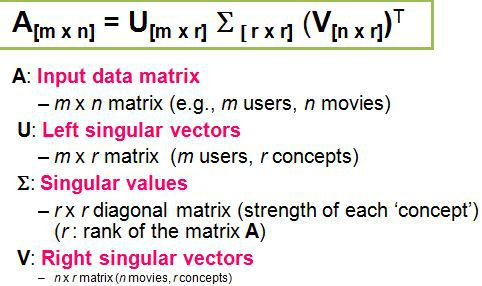
Pearson​​ correlation​​ coefficient​​ was ​​used​​ for​​ similarity ​​measure.

User-User ​collaborative​​ filtering ​​is ​​used​.The ​​following ​​formula​ is ​​used​​ to ​​predict​ ​a ​​user’s​ ​movie ​​rating

****

**Singular ​​Value​​ decomposition:**

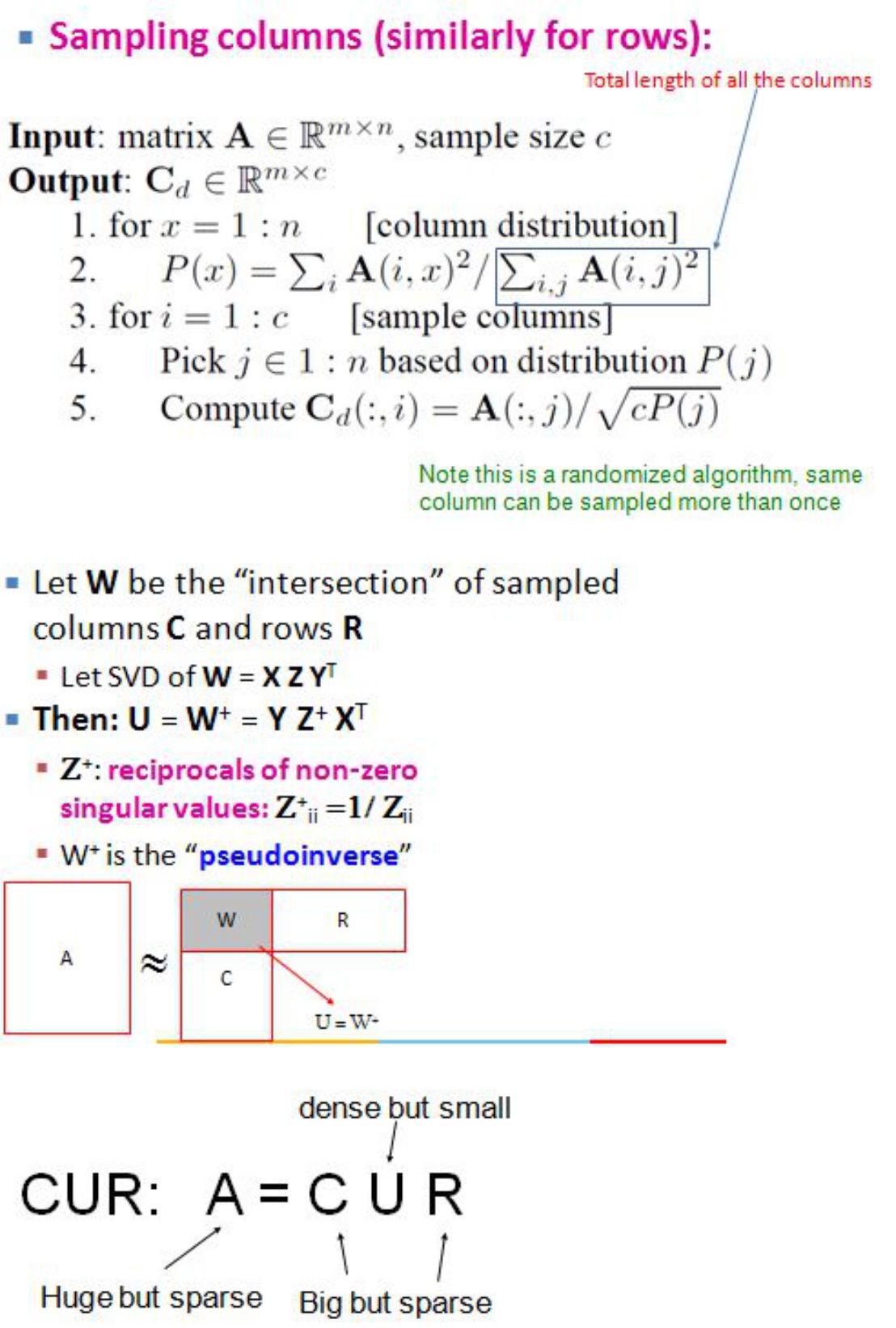
Given​ ​matrix​​ is ​​decomposed​​ to ​​these ​​components**.**​



**SVD ​​with ​​90% ​​energy:**

To​​ find​​ out​​ this​​, the​​ squares​​ of ​​the ​​diagonal ​​elements ​​of ​​Sigma​​ are summed, until​​ the ​​sum​ ​doesn’t​ ​exceed ​​90%​ ​of ​​the​ ​total​ ​energy(which ​​is​​sum ​​of squares​​ of ​​all ​​the ​​diagonal​ ​elements).At​ ​the​​ point​​ it ​​breaks, the​​index​​ is ​​noted and​ ​dimensionality​ ​reduction​​ Is ​​done ​​accordingly.

**CUR:**

****

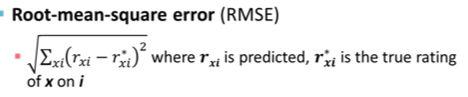
CUR​​ with ​​90%​ ​energy:

To​ ​compute​​ this​​ U=​​pseudoinverse(W)

Where ​​W ​​is ​​the ​​SVD ​​with ​​90% ​​energy​​ of ​​the ​​matrix​​which​​ is intersection ​​of ​​C,R.

**Error-​calculations:**

**Root​ ​mean ​​square ​​error (rmse):**

****

**Precision​​ on​​ top​ ​k:**

k=50

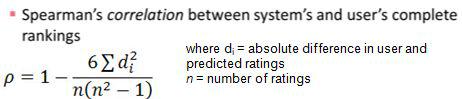
Relevance​​ = 3

If​​ the​​ number​​ of​ ​user ​​ratings ​​considered​ ​are ​​less ​​than ​​k

Value ​​for ​​each ​​user ​​= ​​(Number​​ of ​​predicted​​ user​​ ratings​​ greater​​than relevance)/(Number​​ of ​​user​​ ratings​​ actually ​​greater ​​than​​relevance)

Precision ​​on ​​top ​​k​​ = Mean ​​of ​​the ​​values​ ​for ​​all ​​​relevance

**Rank​​ Correlation:**

****

### 

### Comparisons:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Recommender System | RMSE | Precision on Top k(k=3, rel=2) | Spearman Rank Correlation | Time taken |
| Collaborative | 1.31 | 0.0010604453870625664 | 0.99 | 66.46347651415039 |
| Collaborative with Baseline  Approach | 1.024 | 0.003534817956875221 | 0.99 | 68.66220501824604 |
| SVD | 0.423 | 0.3716861081654297 | 0.99 | 236.37811649303268 |
| SVD with 90% Energy | 0.24 | 0.3891834570519618 | 0.99 | 236.53716842969106 |
| CUR | Depends on c,r but average is around 100 | 0.1378579003181332 | 0.98 | 15.053157166368827 |
| CUR with 90% energy | Depends on c,r but average is around 100 | 0.10763520678685047 | 0.98 | 11.98424341847331 |

### 

### Data Set:

Small: 100,000 ratings applications applied to 1046 movies by 943 users. Last updated 10/2016.