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:

1. Timeit
2. Numpy
3. 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:

$$sim(x,y) = \frac{\sum_{s \in S_{xy}} (r_{xs} - \overline{r_x})(r_{ys} - \overline{r_y})}{\sqrt{\sum_{s \in S_{xy}} (r_{xs} - \overline{r_x})^2} \sqrt{\sum_{s \in S_{xy}} (r_{ys} - \overline{r_y})^2}}$$

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

$$\boxed{r_{xi} = \frac{\sum_{j \in N(i;x)} s_{ij} \cdot r_{xj}}{\sum_{j \in N(i;x)} s_{ij}}}$$

s_{ij}… similarity of items ***i*** and ***j***
r_{xj}…rating of user ***u*** on item ***j***
N(i;x)… set items rated by ***x*** similar to ***i***

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

- Estimate rating r_{xi} as the weighted average:

$$r_{xi} = b_{xi} + \frac{\sum_{j \in N(i;x)} s_{ij} \cdot (r_{xj} - b_{xj})}{\sum_{j \in N(i;x)} s_{ij}}$$

baseline estimate for r_{xi}

$$b_{xi} = \mu + b_x + b_i$$

- μ = overall mean movie rating
- b_x = rating deviation of user x = (avg. rating of user x) – μ
- b_i = rating deviation of movie i

Singular Value decomposition:

Given matrix is decomposed to these components.

$$\boxed{A_{[m \times n]} = U_{[m \times r]} \Sigma_{[r \times r]} (V_{[n \times r]})^T}$$

A: Input data matrix
- *m* x *n* matrix (e.g., *m* users, *n* movies)

U: Left singular vectors
- *m* x *r* matrix (*m* users, *r* concepts)

Σ: Singular values
- *r* x *r* diagonal matrix (strength of each 'concept') (*r* : rank of the matrix **A**)

V: Right singular vectors
- *n* x *r* matrix (*n* movies, *r* concepts)

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:

- **Sampling columns (similarly for rows):**

Total length of all the columns

Input: matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$, sample size c
Output: $\mathbf{C}_d \in \mathbb{R}^{m \times c}$

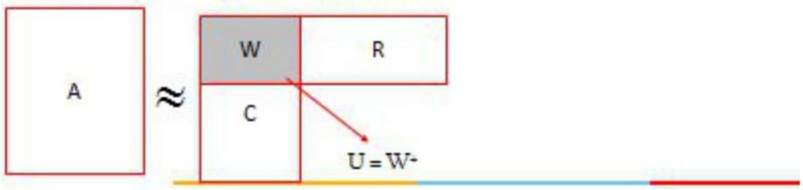
```
for x = 1 : n        [column distribution]
    P(x) = Σ_i A(i,x)^2 / Σ_{i,j} A(i,j)^2
for i = 1 : c        [sample columns]
    Pick j ∈ 1 : n based on distribution P(j)
    Compute C_d(:,i) = A(:,j) / sqrt(c P(j))
```

$$P(x) = \sum_i \mathbf{A}(i,x)^2 / \boxed{\sum_{i,j} \mathbf{A}(i,j)^2}$$

$$\mathbf{C}_d(:,i) = \mathbf{A}(:,j)/\sqrt{cP(j)}$$

Note this is a randomized algorithm, same column can be sampled more than once

- Let **W** be the "intersection" of sampled columns **C** and rows **R**
  - Let SVD of **W** = **X Z Y**^{T}
- **Then: U = W$^+$ = Y Z$^+$ X^T**
  - **Z$^+$: reciprocals of non-zero singular values: $Z^+_{ii} = 1/Z_{ii}$**
  - W$^+$ is the "**pseudoinverse**"



dense but small

CUR: A = C U R

Huge but sparse

Big but sparse

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):

- **Root-mean-square error** (RMSE)
  - $\sqrt{\sum_{xi}(r_{xi} - r^*_{xi})^2}$ where $\boldsymbol{r_{xi}}$ is predicted, $\boldsymbol{r^*_{xi}}$ is the true rating of $\boldsymbol{x}$ on $\boldsymbol{i}$

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:

- Spearman's *correlation* between system's and user's complete rankings

$$\rho = 1 - \frac{6\sum d_i^2}{n(n^2-1)}$$

where d_i = absolute difference in user and predicted ratings
n = number of ratings

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.