Information Retrieval(CS F469)
Design Document
Recommender Systems

By

Kaustubh Rajendra Welankar 2016A7PS0095H
Gaurab Das Gupta 2016A3PS0255H
Aman Kumar Jain 2016A7PS0009H

Various Techniques for Implementing the Recommender System

Collaborative Filtering

Collaborative filtering filters information by using the recommendations of other people. It is based on the idea that people who agreed in their evaluation of certain items in the past are likely to agree again in the future.

Formulation:

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

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***

R_{xI} :Rating of item i by user x

Neighbours taken: 3

Collaborative Filtering with Baseline Approach

The Baseline approach is used to take care of the cold start problem. The baseline is the avg rating + deviation of user + deviation of the movie. The CF gives the deviation from the baseline .

We solve the problem of strict and generous raters by using the centered cosine similarity.

Formulation:

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

baseline estimate for r_{xi}

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

- $\boldsymbol{\mu}$ = overall mean movie rating
- $\boldsymbol{b_x}$ = rating deviation of user $\boldsymbol{x}$ = *(avg. rating of user* $\boldsymbol{x}$*)* $- \boldsymbol{\mu}$
- $\boldsymbol{b_i}$ = rating deviation of movie $\boldsymbol{i}$

Singular Value Decomposition(SVD)

SVD is a matrix factorization technique that is usually used to reduce the number of features of a data set by reducing space dimensions from N to K where K < N.

Formulation:

$$\mathbf{A}_{[m \times n]} = \mathbf{U}_{[m \times r]} \, \Sigma_{[r \times r]} \, (\mathbf{V}_{[n \times r]})^{T}$$

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)

$$\mathbf{A} \approx \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T = \sum_i \sigma_i \mathbf{u}_i \circ \mathbf{v}_i^T$$

σ_i ... scalar
u_i ... vector
v_i ... vector

rows of $\mathbf{V}^t$ are eigenvectors of $\mathbf{D}^t\mathbf{D}$ = basis functions

Σ is diagonal, with δ_{ii} = **sqrt(λ_i)** (*i*th eigenvalue)

rows of **U** are coefficients for basis functions in **V**

(here we assumed that **m > n**, and **rank(D) = n**)

CUR Decomposition

CUR matrix decomposition, an alternative to SVD, is a low-rank matrix decomposition algorithm that is explicitly expressed in a small number of actual columns and/or actual rows of data matrix.

Formulation:

$$\|A\text{-}CUR\|_F \leq \|A\text{-}A_k\|_F + \varepsilon\|A\|_F$$

Where A is Original Matrix CUR is Matrix obtained by CUR multiplication A_k is matrix obtained by retaining k dimensions

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

Here the selected block represent the Frobenius norm of entire matrix

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

$$\mathbf{W = X\ Z\ Y^T}$$

$$\mathbf{U = W^+ = Y\ Z^+\ X^T}$$

In the code 470 rows/colums were selected

Results

Users: 943
Movies: 1682
Number of ratings: 100,000

The algorithms were evaluated based on 3 factors:
1. Root Mean Square Error
2. Spearman Correlation Coefficient
3. Top k Precision

Recomender System Technique	RMSE	Precision on top K	Sperman Rank Correlation	CPU time taken for prediction
User-User Collaborative Filtering (without handling strict and generous raters)	1.0872568586175646	1.0	0.9999999955877837428	5.966830218
User-User Collaborative Filtering(handling strict and generous raters)	1.1505162127509043	0.9859142857142857	0.99999999505941967456	7.495272851999999

Item-Item Collaborative Filtering (without handling strict and generous raters)	0.5353389052320383	1.0	0.9999999989303293208	8.589506576999998
Item-Item Collaborative Filtering(handling strict and generous raters)	0.30995527697619024	0.9593714285714285	0.999999999641416297	10.281696169
Item-Item Collaborative Filtering with Baseline Approach	0.9714891356713724	0.6992857142857143	0.99999999647735908617	10.893180438000002
SVD with 100% energy	0.11631906097256046	0.9452285714285714	0.9999999999494996656	24.193614435000008
SVD with 90% energy	0.28683458021289965	0.9583142857142857	0.9999999996929171983	11.88517726100001
CUR with 100% energy	0.4793803153369837	0.9955714285714286	0.99999999914226551316	12.032446805000006
CUR with 90% energy	0.4854692045928863	0.9942	0.9999999991203379617	12.269587239000003

Packages Used

Here are the following python packages used:

1. numpy
2. math
3. pandas
4. time