Normal CF main sirf ek prediction aati thi. Naive CF tells ~~most~~ all probablities of ratings user can give. Naive CF in this ways shows relative ratings that agar 3 nahi hoti tou 2 hoti ya phir 4 hoti - CF normal wala bolta hai sirf 3 hogi. This is called Confidence interval. Normal CF doesn't tell confidence interval.

Q1) Short logical questions.

Q2) CF (exercise)

Q3) Content-based / Probablistic OR

## Item-based Collaborative Filtering.

$$P(Y_{ui5} = 1) = \frac{\text{No-of items active user rated 1}}{\text{Total ratings of active user.}}$$

prior probability.

$$\frac{\text{No-of items rated 1 by two users}}{\text{No-of items rated 1 by active user and not unrated by second user.}}$$

Serendipity is existing because active user k ilawa koi aur consider horaha hay neighbours wghera. likelihood fay aadhi hay.

If numbers are continuous, we use Gaussian method. CF doesn't work for continuous data.

Scanned with CamScanner

# Probabilistic Methods / Naive Bayes Classifier

- C.F has not any concept of I.D variables.

Classification here some concepts " " " .

$$P(r=1 \mid r=2)$$

Prob. of active user rating item 1.

item to be rated

Prob of (Prob. that user will rate 1)

$$P(y=1 \mid x) = P(y=1)\, P(x \mid y=1)$$

probability that user will rate 1

set of features

valid set of features if rating is given

$\alpha = 0.01$

$\beta = 0.01 \times 5 = 0.05$

no. of ratings 1 2 3 4 5.

users rating same rating as active user to items from 1 - n.

users rating desired rating of unknown item w.r.t active user

works best when users and items are less.

Scanned with CamScanner

We tokenize text to avoid TF-IDF problem.

porterstemmer does stemming

historian historic → history.

derived multiple words are converted to original words.

CountVectorizer calculates no. of words uniquely ocurred.

Scanned with CamScanner

usrprofile means normalizing the final matrix

Add 1st column values = 18.

" Total matrix values = 60

$\frac{18}{60} = 0.3$ for comedy.

usr profile [0.3 0.2 ... ] sums upto one.

then multiply with movie matrix to get weighted movie matrix, which tells how much the user could like features of movies. Then sum up them all and generate Recommendation matrix. Recomend one

Second Approach: (Cosine works best)

Calculate cosine similarity of user profile with each row vector of Movie matrix. Compare similarities. Then recommend.

① If any attribute which is new for user, it won't be recommended to the user.

② It has no serendipity.

Scanned with CamScanner

CF can either be Memory based (1) or Model based (2).

(1) uses complete data to predict/recommend.
user-user CF.

(2) AIT stores similarities b/w data and keep recommending things.
item-based CF.

## Content-based Recommendations:

Content = feature = attributes.
recommendations are upon preferences of me.

Abtak SIT users plot hotay thay ya phir items plot hoti thin tha hum cosine difference lelay thay. Ab CB RS main user ka cosine difference dekhengy saari items / features k satth.

Item profile is made by one-hot encoding from features

Movie 1: $\begin{bmatrix} f_1 & f_2 & f_3 & f_4 & f_5 \\ 1 & 0 & 1 & 0 & 1 \end{bmatrix}$ $f_n$ = features.

User profile =
Matrix of rating by user

$$\begin{array}{c} m1 \\ m2 \\ m3 \end{array} \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$ $m_n$ = movies.

Scanned with CamScanner

Long tail :-

long tail products also need to be recommended, products which are never purchased.

We go for content based. CF fails to solve this because long tails distributions are very hard to solve by CF.

Scaling:

CF has many many computations. It is time consume overhead.

Solution

Factorize matrix and make it dense. and dimensions are reduced to apply CF.

(Item-based me zaden masla aata hai)

similarities calculated by CF are not thrown away for further use. (Dynamic Programming or Offline training)

Scanned with CamScanner

Input → output

take inputs → process a model → generate classification label.

a classification which returns a continuous output. used in calculus. $y = mx + c$ also.

Classification main dependent variables (labels) ko predict karte hain independent (inputs) dekh kar.

CF me independent variables ka concept nahi hota.

When two users are rating less items and some ratings, then we significance rating (weightages)

$\beta$ = threshold, meaning minimum must no. of items rated by both.

$\min(U_1 \cap U_2)$ = means no. of items rated by both users.

$$\text{Sim}(u_1, u_2) \times \beta \cdot \left(\min\left(\overline{(I_1 \cap I_2)}, \beta\right)\right)$$

correlation between two users.

Crowd knowledge: using other people's opinions to recommend one person.

↳ used in user-based CF.

Scanned with CamScanner

bias removal : remove concepts of optimistism and pessimistism.

optimistic = user always rate high } baseline
pessimistic = " " " low } average

Item - based CF. another example.

Adj Cosine ( $I_1$, $I_3$ ) = - - -
= 0.912
baseline average

Comparing Item 1 and 3 above.

If baseline is origin, bias add hoga.
If baseline is average, bias remove hoga.
(CF me bias remove karni chahiye)

Whenever small matrix, item - item tends to similar with user - user predictions. Difference is observed when users or ~~items~~ items surpass each

Serendipity : diversity. It has no link with similar algo
recommend something out of taste.
User-based CF has serendipity as its advantage.
Item-based has comparatively low serendipity
It must be controlled. Not much not less.
If we consider two similar users to one, serendipity is low
" " " 1000 " " " " " " high.

Scanned with CamScanner

Item based collaborative 7i

one lecture missed!

## Item-Based Collaborative Filtering.

Why it is more stable? because users can change preferences but items do not change anyway!

Compare unknown items with known and recommend

one lecture missed, Wania pasriaje!

NO

Cold Start: problem of CF.

initial case mein koi data nai hota. Data ab kia karein? That problem is Cold Start.

Random values lagao then us data py recommend karo.

Jab user ratings dey, tab update karo or recommend karo

Now we are predicting,

using prediction formulae, we predicts:

Pearson shortcoming is when a predictive rating is less than mean value, it suggest not to recommend even if user likes.

When any user rates uniformly (same rating) to all items, then pearson similarity will have zero in denominator which means, infinity. So, we will not use it for correlation and similarity. We will have to ignore that user.

So, pearson fails on uniform distribution.

The formation of prediction $\sum_{u=1}^{n} R_u / n$.

doesnot use mean.. (Not good)

Example is User-Based CF: Another example.

Correlation is strong but user 1 and 2 are giving high ratings to middle 2-5 items, but user 3 is-r

Second prediction formula, which is mean centered.

Different formulae of predictions are available.

active users jisky liye prediction ki jaye.

Metric: measuring similarities

Method: process to predict

Explicit rating: rating by self (3-star on App Store) → hard to get/user is lazy/rate in hurry

Implicit rating: purchased or not (0 or 1). → easy to get

not user friendly

more meaningful

## Collaborative Filtering:

User based: User - User similarity

Hyper-para decision: human decision without any calculations. Choose least distant user for similarity calculations.

Now, predict after calculating distances and similarities.

Euclidean distance doesn't considers slopes (angles) It only checks magnitude. Therefore, we use Cosine similarity which checks magnitude and angle both.

Cosine doesn't check effects. Doesn't check similarity intensity. Cosine angles donot give correlation b/w two or more objects. For this issue, we use Pearson Correlation. It also identifies patterns between rating items among users. Cosine only shows distance between vectors and not correlations.

Pearson is special case of Cosine relationship because correlation checks for mean points distances between two vectors. This is also called Mean-Centered Cosine similarity.

Scanned with CamScanner

# Paradigms of Rec

① Collaborative filtering :
- User based
- Item Based

② Content Based
e.g, Item itself type
Neural ~~Based~~ Network

③ Context Aware
ML

④ Hybrid
~~Combine~~ Combination of more than one type.

⑤ Demography Based
living areas, age group, profession domains,

⑥ Social
Based upon social connections. Not based upon similar ~~com~~ people like user based.

~~User based prediction is calculated~~

# Recommendation System (RS)

Manual recommendation are nearly impossible because its hard to choose few items out of millions and show to different user ~~users~~ nature.

**Benifits of RS**

Increased Sales

good customer experience

## Problem formulation:

$$f : U \times i \longrightarrow R$$

$f$ → function

$U$ → user

$i$ → items

$R$ → ordered list

## RS are based upon:

1. past behaviours (patterns)
2. other users
3. Item itself
4. Context

based on features ratings etc ← (3)

ek lia to dusra bhi lo much ← If Else based ← (4)

↓

Purchase XBOX so you might purchase controller.

Scanned with CamScanner