Recommender Systems

SIR ZAIN

Project 10%.
Finals 50%.
S1 , S2 30%.
Assignments 5%.
Quiz 5%.

have to use any 2 techniques
show rating predictions from both

Soch kay purchase ✗
recomm purchase ✓

Information Overload

↳ too much info

↓

filtering

↓

IDEA BEHIND RS

million flavours of icecream → HOW TO CHOOSE TOP 5

TOO MANY OPTIONS ↓ MAKE IT LESSER ↓ EASIER TO CHOOSE

↓

Ratings

Implicit

MORE USEFULL

many people just scroll, dont like

↳ is collective ??

eg purchasing a product

↳ likes is collective

↳ has no dislike

con

↳ harder to collect

pro

↳ not everyone rates

↳ -ve comments are implicit ratings

↳ useful

Explicit

IMDb

↳ on a scale

↳ 1 star, i dont like it

↳ 5 star, i love it

pro

↳ easy to collect

con

↳ not everyone rates

↳ assumes no rating is positive

↳ not useful

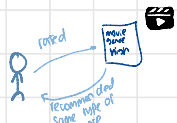
↓

Genere

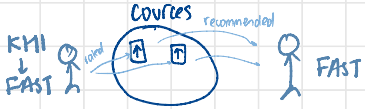
↳ normally rated high by active user

1. Similarity b/w items
2. Similarity b/w users

1.



2.



Reccomendation

↳ A suggestion to help user in decision making

use to improve user experience

Adv of Reccomendation

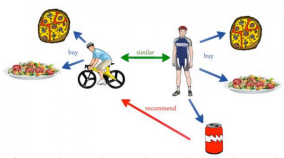
↳ To increase sales

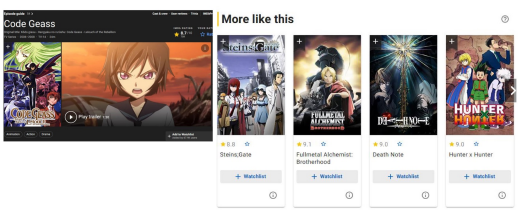
↳ To improve user experience

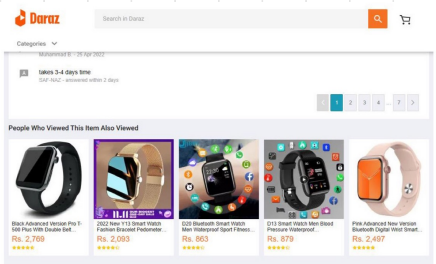
↳ To maximise productivity

Example

If **cyclist A** always orders *pizza* and *salad* at our food joint, and **cyclist B** eats *pizza* and *salad* as well. Then recommend cola to **cyclist A** if **cyclist B** had been ordering it lately







Recommender system (RS)

↳ helps match users with items

↳ It eases info overload

↳ helps improve user experience

↳ helps increase sales

↳ suggest novel items

How does RS work?

↳ It bases reccomendation on

↳ Past behavior pattern of that user

↳ Similarity with other users

↳ Item similarity

↳ context

$$f : U \times I \rightarrow R$$

U: users; I: items; R: Recommended items → ordered list

- For each user u, what we want to do is to choose the item i that maximizes f

$$S_u = \underset{i}{\text{argmax}}\, f(u,i)$$

PARADIGMS OF RECOMMENDER SYSTEM

1. Collaborative based

Most common

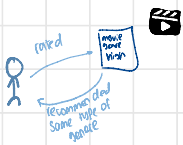
↳ Collaborative Filtering (CF)

1. User based
   ↳ find users similar to me
   ↳ then recommend me what they like
2. Item based
   ↳ recommend me an item similar to the ones I normally like

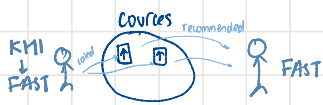
2. Content based approach

↳ doesnt req other users data
↳ only users own history patterns
↳ minimal collaboration

1.

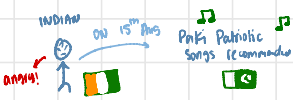


2.



3. Demographic approach

↳ eg Spotify

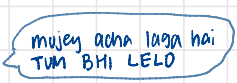


4. Social Tagging → aka Connection based

↳ connection based

mutual friends suggested
friend liked Pages suggested
→ on social media
↳ insta
↳ facebook

5. Trust based approach



↳ normally in news recommendation
eg stack overflow, social media posts
↳ reliability is important

similarity based ✗
trust based ✗✓ → mixed

based on what people in a users connection like
eg Fb group recommendation

6. Hybrid approach

↳ combination of 2 or > of the above mentioned approaches

Collaborative Filtering (CF)

* collaborative and similarity are SAME THING

↳ uses similarities to recommend items to active user → Needs ratings! → no features, only raw ratings

↳ uses similarity, ratings

↳ It has 2 approaches

1. User based CF

↳ find users similar to me
↳ then recommend me what they like
  ↳ if most similar user rated it high then it should be recommended

↳ users are vectors

↳ similarity b/w users

PROS

↳ more diverse recommendations

↳ better choice if no of items > no of users → this is common in practice

CONS

↳ not stable → e.g likeness changes over item. I rated something high but I don't like it as much anymore → after a year users preferences might change

↳ can't provide indepth analysis of a user

2. Item based CF → more accurate

↳ recommend me an item similar to the ones I normally like

↳ items are vectors

↳ Similarity b/w items

PROS

↳ provides more accurate recommendations

↳ better choice if no of users > no of items

↳ is more stable → major adv

↳ can provide indepth analysis on users

CONS

↳ is prone to shilling attacks → major disadv → Sab mil kay movie ko kam rating dayengay → Campaign to pay to give high or low rating

↳ Provides much less diversity than user based CF

TERMINOLOGIES OF CF

↳ Active user → the user that's given recommendation

↳ Active item → deciding whether it should be recommended or not

↳ Interactive Table/Matrix

↳ Metric → to measure similarity between users

↳ Method → for subset consisting of closest neighbour

INPUT → sparse matrix

	I_1	I_2	I_3	I_4
U_1	4		3	2
U_2		3	2	5
U_3	1	2	3	
U_4		2	4	

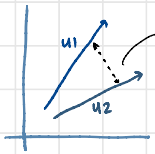
rows tall → users

Similarity is measured using

Vector Space

↳ Interactive Table/matrix as vectors

VECTOR SPACE MODEL



Distance measure
↳ Eudielean
↳ Cosine
↳ manhattan
↳ Corelation
↳ Raw distance

1.1 User based CF APPROACH

Q) U_1 is active user
Predict for I_5 of U_1

STEPS

1. Choose a K (no of similar users — hyperparameter we choose)
   - taking K=2
2. measure similarity
   1. Euclidean = U_2, U_3
   2. Cosine = U_2, U_3
   3. Pearson = U_2, U_4 → uses mean
   4. Adj Cosine Similarity → uses mean
3. Choose Prediction Method
   1. Average
   2. Weighted Average Item 5 of U_1 = 3
   3. Mean Adjusted Weighted Avg → uses mean
   4. Significance Weighting

		I_1	I_2	I_3	I_4	I_5
Active user →	U_1	5	3	4	4	?
	U_2	3	1	2	3	3
	U_3	4	3	4	3	5
	U_4	3	3	1	5	4
	U_5	1	5	5	2	1

dim = 5, only 4 compared as we dont count active item

no of users = 5, only 4 compared as we dont count active user

STEP 1. Choose K

↳ Take K = 2

STEP 2. Measuring User Similarity

Similarity = 1 − Distance ⟶ anti-distance

1. Find distance b/w AC and all other users
2. Find k nearest neighbour of AC

↳ Some similarity measures

1. Euclidean

good to start from not the best

↳ lesser distance = most similar

$$d: \sqrt{(x_2 - x_1) + (y_2 - y_1)^2}$$

eg $V_1 = (3,10)$ $V_2 = (7,13)$

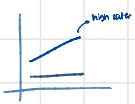
$d: \sqrt{(7-3)^2 + (13-10)^2}$

$= 5$

CONS

↳ dosent consider patterns

eg pattern: U_2 rates low, U_3 rates high → dosent take this into consideration



↳ dosent consider slope, only distance

$\sqrt{(5-3)^2 + (3-1)^2 + (4-2)^2 + (4-3)^2}$

		I_1	I_2	I_3	I_4	I_5	$d_{\text{with } U_1}$	Similarity with U_1
Active user	U_1	5	3	4	4	?	0	
✓	U_2	3	1	2	3	3	3.60	−2.6 → 1−3.6
✓	U_3	4	3	4	3	5	1.41	−0.41 → 1−1.41
	U_4	3	3	1	5	4	3.74	−2.74 → 1−3.74
	U_5	1	5	5	2	1	5	−4 → 1−5

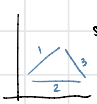
for K=2, U_2 and U_3

2. Cosine Similarity

sparse matrix as not everyone rates

↳ considers slope and distance

↳ low rater
↳ high rater
↳ indifferent



slope identifies rating pattern

↳ treats each item seperately

$$\cos\theta = \frac{A \cdot B}{\|A\| \cdot \|B\|}$$

		I_1	I_2	I_3	I_4	I_5	Cosθ Similarity with U_1
Active user	U_1	5	3	4	4	?	
✓	U_2	3	1	2	3	3	0.97
✓	U_3	4	3	4	3	5	0.99
	U_4	3	3	1	5	4	0.89
	U_5	1	5	5	2	1	0.79

$\frac{(5\times3)+(3\times1)+(4\times2)+(4\times3)}{\sqrt{5^2+3^2+4^2+4^2}\sqrt{3^2+1^2+2^2+3^2}}$

$\frac{(5\times4)+(3\times3)+(4\times4)+(4\times3)}{\sqrt{5^2+3^2+4^2+4^2}\sqrt{4^2+3^2+4^2+3^2}}$

for K=2, U_2 and U_3

CON

↳ cant measure corelation b/w 2 users

user 2 likes something else
user 3 likes something else

3. Pearson Correlation Coefficient (r)

↳ measures magnitude and orientation b/w data points

↳ strength and r/s is given by -1 to 1

- -ve → strong -ve correlation → if U1 likes this, U2 hates it → inverse
- 0 → no correlation
- +ve → strong +ve correlation → if U1 likes this, U2 likes it just as much

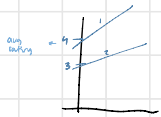
↳ treats each item as a whole collection

↳ uses magnitude

↳ what is my general rating pattern →

↳ which is most similar user

cosine w.r.t. origin



correlation w.r.t.

$$r = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \cdot \sum (y_i - \bar{y})^2}}$$

(mean)

		I_1	I_2	I_3	I_4	I_5	Mean	r
Active user	U_1	5	3	4	4	?	4	
✓	U_2	3	1	2	3	3	2.4	0.85
	U_3	4	3	4	3	5	3.8	0
✓	U_4	3	3	1	5	4	3.2	0.70
	U_5	1	5	5	2	1	2.8	-0.76

$\frac{(5-4)(3-2.4)+(3-4)(1-2.4)\ldots}{\sqrt{(5-4)^2+(3-4)^2+(4-4)^2+\ldots}\sqrt{\ldots}}$

$\frac{1+5+5+2+1}{5}$

for k=2, U_2 and U_4

CON

↳ if uniform distribution the line will be flat

there might be a user that rated everyone 3

$\sqrt{(3-3)^2 \ldots 000}$ → UNDEFINED

↓ SOLUTION

↳ add another item and give it another value → kinda like smoothing → but not for everyvalue

↓ How many times will we do this

↓ too many calculations!

CF

↳ high resource

↳ high Processing power

↳ high storage

↳ will work for < 100 users

* Exam advice

↳ make subtraction column with mean

↳ as many people make this mistake

Pearson Correlation Coefficient: Issues

- ❑ Underlying assumption is that users dislike what they rated below average
- ❑ This is not true in practice (we rate only what we liked or highly disliked)
- ❑ The correlation *flattens* in case of uniformly distributed ratings

Deviation from average rating on shared items

$$sim(a,b) = \frac{\sum_{p \in P}(r_{a,p} - \bar{r}_a)(r_{b,p} - \bar{r}_b)}{\sqrt{\sum_{p \in P}(r_{a,p} - \bar{r}_a)^2} \sqrt{\sum_{p \in P}(r_{b,p} - \bar{r}_b)^2 + \varepsilon}}$$

!!! Will be zero in case of uniform rating !!!

*interaction table must never change

STEP 3. Prediction Method

* example below are based on using Pearson correlation as a similarity measure

1. Average

		I_1	I_2	I_3	I_4	I_5
Active user	U_1	5	3	4	4	4
	U_2	3	1	2	3	3
	U_4	3	3	1	5	4

round up

$\frac{3+4}{2} = 3.5$

CON

↳ Outliers effect

E1 Q) U_1 is AU, find I_5

	Item 1	Item 2	Item 3	Item 4	Item 5	Mean	Pearson Correlation Similarity with User 1
User 1	5	3	4	4	?	4	1
User 2	3	1	2	3	3	2.4	≈ 0.85
User 3	4	3	4	3	5	3.8	≈ 0
User 4	3	3	1	5	4	3.2	≈ 0.70
User 5	1	5	5	2	1	2.8	≈ -0.76

K=2

E2 Q) U_3 is AU, find I_1 and I_6

	Item 1	Item 2	Item 3	Item 4	Item 5	Item 6	Mean	Pearson Correlation Similarity with User 3
User 1	7	6	7	4	5	4	5.5	0.894
User 2	6	7	?	4	3	4	4.8	0.939
User 3	?	3	3	1	1	?	?	1
User 4	1	2	2	3	3	4	2.5	-1
User 5	1	?	1	2	3	3	2	-0.817

K=2

2. Weighted Average

↳ multiply with AI

$$R_U = (\sum_{u=1}^{n} R_u)/n$$

E1 ⇒

		I_1	I_2	I_3	I_4	I_5	r
Active user	U_1	5	3	4	4	3	
	U_2	3	1	2	3	3	0.85
	U_4	3	3	1	5	4	0.70

$$R_{15} = \frac{3 \times 0.85 + 4 \times 0.7}{|0.85| + |0.7|} = 3.45 \xrightarrow{\text{round}} 3$$

User 1 Item 5

Sb no -ve value

$R_{15} = 3$ → Should we recommend or not?

Depends on cutoff YOU CHOOSE

taking 3 as cut off ← hyper parameter based on domain knowledge / business decision

E2 ⇒

$$R_{31} = \frac{(7 \cdot 0.894) + (6 \cdot 0.939)}{|0.894| + |0.939|} \approx 6.49$$

$$R_{36} = \frac{(4 \cdot 0.894) + (4 \cdot 0.939)}{|0.894| + |0.939|} = 4$$

CON

↳ incorrect assumption based on correlation

SOLUTION

3. Mean Adjusted Ratings

↳ independent of whether user is generous or not

↳ mean centered Prediction function

↳ removes bias

$$R_U = \overline{r_a} + \frac{\sum_{b \in N} sim(a,b) * (r_{b,p} - \overline{r_b})}{\sum_{b \in N} sim(a,b)}$$

E2 ⇒

		I_1	I_2	I_3	I_4	I_5	I_6	Mean	r
	U_1	7	6	7	4	5	4	5.5	0.894
	U_2	6	7	?	4	3	4	4.8	0.939
Active user	U_3	?	3	3	1	1	?	2	

mean of active user

$$R_{31} = 2 + \frac{(1.5 * 0.894) + (1.2 + 0.939)}{|0.894| + |0.939|} \approx 3.35 = 3$$

$$R_{36} = 2 + \frac{(-1.5 * 0.894) + (-0.8 * 0.939)}{|0.894| + |0.939|} \approx 0.86 \approx 1$$

Question 3 (CLO: 1) **10 points**

Consider the given interaction matrix for Crunchyroll website users.

	Attack on Titan	My Hero Academia	Code Geass	Vinland Saga	Mean
User 1	9	7	2	8	6.5
User 2	?	7	5	7	6.33
User 3	10	4	9	5	7
User 4	8	5	8	6	6.75

0.958 ✓ → User 2 ; 0.881 User 3 ; 0.943 ✓ User 4

a) Find the Cosine similarity between all users.

a) Cosine Similarity

U1, U2 = 0.958 U2, U3 = 0.881 U3, U4 = 0.986

U1, U3 = 0.839 U2, U4 = 0.943

U1, U4 = 0.833

b) Predict the missing rating $R(U2, I1)$ using mean-centered prediction function for $k=2$.

b) K=2, closest are U1, U4

$$R_{U2,U1} = 6.33 + \frac{(9-6.5)0.958 + (8-6.75)(0.943)}{|0.958| + |0.943|} = 8.21 \approx 8$$

Item based collaborative filtering

same formulas → more accurate

↳ same as user based but similarity b/w items → not users

↳ items are vectors

"Things don't change as much as people do." — Made-up quote

↳ more stable → after years it stays the same

Q) U_3 is AU, Predict for I_1 and I_6 (AU = Active User)

doesn't matter what's in rows or column → can transpose for own self

	I_1	I_2	I_3	I_4	I_5	I_6	Mean
U_1	7	6	7	4	5	4	5.5
U_2	6	7	?	4	3	4	4.8
Active User → U_3	?	3	3	1	1	?	2
U_4	1	2	2	3	3	4	2.5
U_5	1	?	1	2	3	3	2

Step 2: Adjusted cosine similarity

almost always used for item based CF

$$sim(\vec{a},\vec{b}) = \frac{\sum_{u\in U}(r_{u,a}-\bar{r}_u)(r_{u,b}-\bar{r}_u)}{\sqrt{\sum_{u\in U}(r_{u,a}-\bar{r}_u)^2}\sqrt{\sum_{u\in U}(r_{u,b}-\bar{r}_u)^2}}$$

for I_1, calculate Adj cosine with all other items

↓ after subtracting mean

	I_1	I_2	I_3	I_4	I_5	I_6	Mean
U_1	1.5	0.5	1.5	-1.5	0.5	-1.5	5.5
U_2	1.2	2.2	?	-0.5	-1.8	-0.8	4.8
U_3	?	1	1	-1	1	?	2
U_4	-1.5	-0.5	-0.5	0.5	0.5	1.5	2.5
U_5	-1	?	-1	0	1	1	2

Adj cosine

$$(1,3) = \frac{(1.5)(1.5)+(-1.5)(-0.5)+(-1)(-1)}{\sqrt{1.5^2+(-1.5)^2+(-1)^2}\cdot\sqrt{1.5^2+(-0.5)^2+(-1)^2}} = 0.912$$

$$(1,4) = \frac{(1.5)(-1.5)+(1.2)(-0.5)+(-1.5)(0.5)}{\sqrt{1.5^2+1.2^2+1.5^2+(-1)^2}\sqrt{(-1.5)^2+(-0.5)^2+0.5^2}} = -0.824$$

$$(1,2) = \frac{(1.5)(0.5)+(1.2)(2.2)+(-1.5)(-0.5)}{\sqrt{1.5^2+1.2^2+1.5^2}\sqrt{0.5^2+2.2^2+0.5^2}} = 0.735$$

∴ do the rest

use only dim commonly present

K=2, so I_2 and I_3

Step 3: Prediction function

↳ using weighted avg

$$R_{31} = \frac{(3)(0.735)+(3)(0.912)}{|0.735|+|0.912|} = 3$$

(I_2: 0.735, I_3: 0.912) → Predicted rating of I_1 of U_3

* Do the same steps for I_6

Collaborative Filtering (Active user) VS Classification

↳ no distribution b/w dependent and independent variables

↳ similar to missing value analysis but with a much larger matrix

	I_1	I_2	I_3	I_4	
U_1	5	?	?	3	
U_2	5	1	?	?	= 1
U_3	3	?	2	5	

unfair advantage ↑ common rating

↓ SOLUTION

Significance Weighting (optional but recommended)

↳ tweak the score based on no of items rated together by these users

↳ uses discount factor

↳ It kicks in when no of common ratings b/w 2 users < a particular threshold β

- The reliability of any similarity function ***sim(u, v)*** between two users **u** and **v** is often affected by the number of common ratings between **u** and **v** *i.e.* ($I_u \cap I_v$)
- When the two users have only a small number of ratings in common, the similarity function ***sim(u, v)*** should include a discount factor to **de-emphasize** the importance of that particular user pair

$$\text{DiscountSim}(u,v) = \text{Sim}(u,v) \cdot \frac{\min\{(I_u \cap I_v), \beta\}}{\beta}$$

Sim(u,v): original similarity score; $(I_u \cap I_v)$: no. of common ratings b/w U_u and U_v; β: threshold value → hyperparameter

	I_1	I_2	I_3	I_4	I_5	I_6	$I_{U_5} \cap I_{\dots}$ (common items)	r	Similarity
U_1	7	6	7	4	5	4	4	0.894	1
U_2	6	7	?	4	3	4	3	0.939	0.7
Active user → U_3	?	3	3	1	1	?			
U_4	1	2	2	3	3	4	4	-1	
U_5	1	?	1	2	3	3	3	-0.817	

taking β = 4

$U_1,U_1 = 0.894 \times \frac{\min(4,4)}{4} \cdot 1$

$U_5,U_2 = 0.939 \cdot \frac{\min(3,4)}{4} \cdot 0.7$

Collaborative Filtering

CONS

Issues
- highly dependent on data
- not good for sparse matrix → less data
- not everything is rated

↳ Serendipity

basically randomness → so no overfitting

- ↳ expand users taste into neighbouring areas
- ↳ measure of recomending something new
- ↳ personilization is indirectly proportional to srendipity

  large value of K → high serendipity

it is good to recommend something new to the user

↳ Cold start user

- ↳ new user with no info
- ↳ hard to recommend

SOLUTION
- ↳ make them rate popular items in the begining

Cold start item

- ↳ items with no rating
- ↳ so no similarity can be calculated

SOLUTION
- ↳ randomly pop up those items and make user rate it

	CoU (cold start user)	CoI (cold start item)
U-CF	X (dont have enough info)	X (no one rated so cant do anything)
I-CF	X (no common item)	X

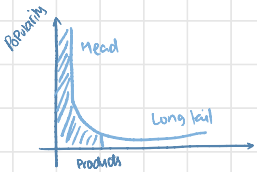
↳ Long tail

- ↳ A large no. of items will be unraked → hence cannot be recommended easily
- ↳ often leads to sparsity → not having enough data to make a prediction
- ↳ products with less views

↓SOLUTION

can never be fully resolved → but can work with CF

- ↳ Prioritize unwated movies more



↳ Scalling

- ↳ NOT scable
- ↳ as if too many users or items the too high resourse which is hard to provide

↓solution

- ↳ use offline training
  - ↳ Pre computed calculations can be stored offline → save CPU resource
  - ↳ matrix factorization → save storage

CF
- ↳ high resource
  - ↳ high Processing power
  - ↳ high storage

will work for < 100 users

*recommender system needs to be simple and resource req is low → as too much data

CRITERIA TO JUDGE RECOMMENDER SYSTEMS

Memory Based

↳ uses entire data everytime rating is predicted — gets data calculates formula EACH TIME

↳ User based CF

Model Based

↳ uses data once to create a model then makes new prediction w/o using the entire data again → pre stores traversed data

↳ Item based CF → get similarity b/w items and store them offline

↳ Content based RS

Content based RS

↳ match user with items that are similar to what they highly rated in the past — 0 serendipity, high personalization, features

↳ history based → uses your own past ratings

↳ uses item features, not similarity b/w users → there are no other users

↳ good for personalized recommendations

Collaborative based RS

only ratings

↳ uses similarity, using ratings

N amazon

USES

↳ movie, blogs, websites news articles, wiki pages

Content

↳ explicit features of the item

eg for movie

↳ Genre: Action/Superhero

↳ Actor: Robert Downey Jr.

↳ Year: 2019

- It can also be textual content (title, description, table of content, *etc.*)
  - Several techniques to compute the distance between two textual documents
  - Can use NLP techniques to extract content features
- Can be extracted from the signal itself (audio, image)

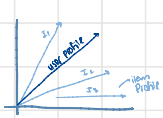
Vector Space Model (VSM)

↳ each item is saved as a vector of its features

↳ in n-dimensional space

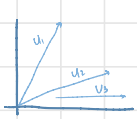
↳ similarity: angle b/w 2 vectors

↳ every user has a user profile vector

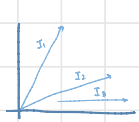


dim: no. of features

Content Based



dim: no. of users



dim: no. of items

Collaborative Filtering

Content based Filtering APPROACH 1

Steps

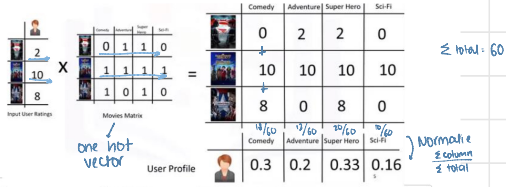
1. User Profile (user vector of rated items)
   - ↳ one hot vector (rated) x Input user ratings
   - ↳ Normalise = $\frac{\Sigma \text{column}}{\Sigma \text{total}}$
2. Reccomend
   1. Weighted Avg
      - ↳ one hot vector (unrated) x User Profile
      - ↳ Row Sum
   2. Cosine Similarity

- A lazy approach is to recommend an item containing content that is most similar to that present in user's highly rated items
  - Also called **lazy approach**
- More novel approaches involve training a classifier *(decision trees, neural networks etc.)* to predict a rating given a set of features as input

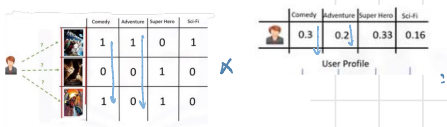
What if another feature
↳ Then add another dim
then take avg

Q) features → 1 : Genre

- We first create a user profile based on already rated items (and attributes)



- Which one of the yet unrated movies should we be recommending?



	G_1	G_2	G_3	G_4	Weighted Avg
M_1	0.3	0.2	0	0.16	0.66 → Reccomend this movie
M_2	0	0	0.33	0	0.33
M_3	0.3	0	0.33	0.16	0.63

Row sum 0.3+0.33+0.16

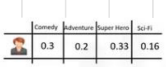
*can use cosine similarity aswell

Q) Using cosine similarity

- **Task:** Find Cosine similarity between user profile and movie matrix and compare with previous results.

0.09 0.04 0.0756

	G_1	G_2	G_3	G_4	Weighted Avg
M_1	0.3	0.2	0	0.16	0.76
M_2	0	0	0.33	0	0.64
M_3	0.3	0	0.33	0.16	0.86 → Reccomend this movie



if someone wants rating, make in scale of 0 and 10

0.66 ≃ 6.6 ≃ 7
0.33 ≃ 3.3 ≃ 3
0.63 ≃ 6.3 ≃ 6

Question 2 (CLO: 1) — **10 points**

For the given data, create user profile and item profiles. Then use these profiles to predict ratings for Item 2 and Item 4 with content-based filtering.

User	Item 1	Item 2	Item 3	Item 4
Features	*f1, f2*	*f1, f3*	*f2, f3*	*f2*
Rating	5		3	

Note: The allowed values for ratings are 1, 2, 3, 4 and 5.

Rated items

	F1	F2	F3
I_1	1	1	0
I_3	0	1	1

× User ratings

R
5
3

= User Profile

	F1	F2	F3
I_1	5	5	0
I_3	0	3	3

Total: 16

→ Normalise User Profile

F1	F2	F3
0.3	0.5	0.2

Unrated items

	F1	F2	F3
I_2	1	0	1
I_4	0	1	0

× Normalized user Profile

F1	F2	F3
0.3	0.5	0.2

=

	F1	F2	F3
I_2	0.3	0	0.2
I_4	0	0.5	0

= Row Sum weighted Avg

0.5
0.5

either I_2 or I_4 can be recommended

Content based Filtering APPROACH 2 → active item

↳ Text recommendation → TF-IDF → Making Features

$$TF_{tk,dj} = \frac{freq(tk)}{max(t_i, d_j)}$$ (row wise)

just raw frequency doesn't give imp of term →

$$IDF_{tk} = \log_{10}\frac{N}{n_k}$$ (column wise)

the more the term occurs the lesser the impt

Steps

1. TF
2. IDF
3. TF x IDF
4. Magnitude = $\sqrt{\Sigma \text{ of row}^2}$
5. Normalise = $\frac{TF \cdot IDF}{Magnitude}$
6. Cosine similarity

Q) BIOG 1 is AI (Active item)

* if active item not given choose any by yourself

$TF = \frac{29}{61} = 0.47$ (29 = freq(Bussiness, B1); 61 = max(freq(any other term B1)))

BIOG	Bussiness	Invention	Hospital	TFB	TFI	TFH
Active item → 1	31	27	0	1.14	0.87	0
2	29	61	3	0.47	2.10	0.04
3	52	99	6	0.52	1.90	0.06
4	10	29	5	0.34	2.9	0.17
5	13	41	7	0.31	3.15	0.17
DOC freq	2500	1200	800			
IDF	1.3	1.61	1.79			

$\log_{10}\frac{50000}{2500}$ $\log_{10}\frac{50000}{1200}$ $\log_{10}\frac{50000}{800}$

Corpus contains

N → 50000 blogs

2500 → Bussiness

1200 → Invention

800 → hospital

↓ TF x IDF

Blog	Bussiness	Invention	Hospital	magnitude
Active item → 1	1.48	1.40	0	2.03 → $\sqrt{1.48^2 + 1.4^2 + 0^2}$
2	0.61 ← 0.47×13	3.38	0.07	3.43
3	0.67	3.05	0.10	3.12
4	0.44	4.66	0.30	4.69
5	0.40	5.07	0.30	5.09
Doc freq	2500	1200	800	
IDF	1.3	1.61	1.79	

↓ Normalise = Divide by magnitude
using length so scale is 0-1

Blog	Bussiness	Invention	Hospital	Cosine Similarity
Active item → 1	0.73 ← 1.48/2.03	0.68	0	
2	0.17	0.98	0.02	0.79 ✓ → $\frac{(0.73)(0.17)+(0.68)(0.98)+(0)(0.02)}{\sqrt{0.73^2+0.68^2+0^2}\cdot\sqrt{0.17^2+0.98^2+0.02^2}}$
3	0.21	0.97	0.03	0.82 ✓
4	0.09	0.99	0.06	0.74
5	0.07	0.99	0.05	0.73
Doc freq	2500	1200	800	
IDF	1.3	1.61	1.79	

Blog 2 and 3 will be recommended

PROS

- scalable new users → doesn't need too many resources
- highly personalised → max personalization
- Partially solves cold start item → doesn't solve for cold start user
- recommendations are explained
  - you liked this hence it is recommended

CONS

- hard to construct features → if features are not readily available like images
  - too many calculation
- little to no serendipity
- suffers from cold start user problem

ishma hafeez notes

DESIGNING A RECOMMENDATION SYSTEM

Probabalistic CF

- aka Naive Bayes CF
- dosent have vector same so more accurate than normal CF → rating are features

Naive Bayes classifier

↳ a supervised multi class classification algorithm

prediction (Posterior) | likelihood | Prior

$$P(Y|X) = \frac{P(X|Y) \times P(Y)}{P(X)}$$

$P(X)$ → if features stay constant then remove it

Naive Bayes Collaborative Filtering

↳ Probabilities rated based on Bayes Rule

↳ assumes ratings are independent → cant use this algo if this is false → imp

↳ this approach can be bom

1. user based
2. item based

1. User based NB CF

→ no of users FOCUS

Prior

$$P(r_i = y) = \frac{(\text{no of users who rated item } i \text{ as } y) + \alpha}{(\text{no of users who rated item } i) + \beta}$$

→ smoothing → so value isn't '0'

likelihood

$$P(r_j = k \mid r_i = y) = \frac{(\text{no of users who rated item } j \text{ as } k \text{ and item } i \text{ as } y) + \alpha}{(\text{no of users who rated item } j \text{ and rated item } i \text{ as } y) + \beta}$$

item

$$\beta = R \times \alpha$$

R → no of possible ratings; α → hyperparameter

Q) How Probable is the rating 1 for item 5?

Take out for all items → rating → 1, 2, 3, 4, 5

↓ Active item

↓ x column
→ y rows

Active User		I_1	I_2	I_3	I_4	I_5
→	U_1	1	3	3	2	?
	U_2	2	4	2	2	4
	U_3	1	3	3	5	1
	U_4	4	5	2	3	3
	U_5	1	1	5	2	1

→ for $r_{is}=1$

*if item not rated dont consider the column

*if no ratin given, then dont count that user

*if β not given
α = 0.002
β = α × possible ratings

likelihood

likelihood for f_1

likelihood for f_2

likelihood for f_3

likelihood for f_4

$$P(X|r_{is}=1) = P(r_{i1}=1|r_{is}=1) \times P(r_{i2}=3|r_{is}=1) \times P(r_{i3}=3|r_{is}=1) \times P(r_{i4}=2|r_{is}=1)$$

AU: no of users who rated I_i as 1 → based on neighbourhood

AS: how many 1s in I_5

U_{i2}, U_{i3}, U_{i4}

$$= \frac{2+0.01}{2+0.05} \times \frac{1+0.01}{2+0.05} \times \frac{1+0.01}{2+0.05} \times \frac{1+0.01}{2+0.05} = 0.117$$

$$P(X|r_{is}=2) = P(r_{i1}=1|r_{is}=2) \times P(r_{i2}=3|r_{is}=2) \times P(r_{i3}=3|r_{is}=2) \times P(r_{i4}=2|r_{is}=2)$$

$$\frac{0+0.01}{0+0.05} \times \frac{0+0.01}{0+0.05} \times \frac{0+0.01}{0+0.05} \times \frac{0+0.01}{0+0.05} = 0.0016$$

$$P(X|r_{is}=3) = P(r_{i1}=1|r_{is}=3) \times P(r_{i2}=3|r_{is}=3) \times P(r_{i3}=3|r_{is}=3) \times P(r_{i4}=2|r_{is}=3)$$

no of users who rated I_i as 1 → based on neighbourhood

how many 3s in I_5

U_{i2}, U_{i3}, U_{i4}

$$= \frac{0+0.01}{1+0.05} \times \frac{0+0.01}{1+0.05} \times \frac{0+0.01}{1+0.05} \times \frac{0+0.01}{1+0.05} = 0.000000082$$

no of users who rated I_i as 3

$$P(X|r_{is}=4) = P(r_{i1}=1|r_{is}=4) \times P(r_{i2}=3|r_{is}=4) \times P(r_{i3}=3|r_{is}=4) \times P(r_{i4}=2|r_{is}=4) = 0.000000083$$

$$= \frac{0+0.01}{1+0.05} \times \frac{0+0.01}{1+0.05} \times \frac{0+0.01}{1+0.05} \times \frac{1+0.01}{1+0.05}$$

$$P(X|r_{is}=5) = P(r_{i1}=1|r_{is}=5) \times P(r_{i2}=3|r_{is}=5) \times P(r_{i3}=3|r_{is}=5) \times P(r_{i4}=2|r_{is}=5) =$$

$$\frac{0+0.01}{1+0.05} \times \frac{0+0.01}{1+0.05} \times \frac{0+0.01}{1+0.05} \times \frac{0+0.01}{1+0.05} = 0.0016$$

Active user		I_1	I_2	I_3	I_4	I_5 (Active item)
→	U_1	1	3	3	2	?
	U_2	2	4	2	2	4
	U_3	1	3	3	5	1
	U_4	4	5	2	3	3
	U_5	1	1	5	2	1

* On ans, dont write Posterior Probability but the class label

if multiply numerator by a weight → overfitting
if multiply denominator by a weight → underfitting

* if prior 0, then no need to calculate its likelihood

Prior

$$P(r_{is}=1) = \frac{2+0.01}{4+0.05} = 0.496$$

(no of users who rated I_5 as 1) / (no of users who rated I_5)

Posterior, likelihood of $r_{is}=1$, Prior of $r_{is}=1$

$$P(r_{is}=1|X) = 0.117 \times 0.496 = 0.058$$

→ gives max
Hence U_1 I_5 = 1 rating

$$P(r_{is}=2) = \frac{0+0.01}{4+0.05} = 0.002$$

$$P(r_{is}=1|X) = 0.0016 \times 0.002 = 0.000,0032$$

$$P(r_{is}=3) = \frac{1+0.01}{4+0.05} = 0.249$$

$$P(r_{is}=1|X) = 0.000,000,082 \times 0.249 = 0.00000002$$

$$P(r_{is}=4) = \frac{1+0.01}{4+0.05} = 0.249$$

$$P(r_{is}=1|X) = 0.000,000,83 \times 0.249 = 0.0000002$$

$$P(r_{is}=5) = \frac{0+0.01}{4+0.05} = 0.002$$

$$P(r_{is}=1|X) = 0.0016 \times 0.002 = 0.0000032$$

2. Item based NB CF

no of users FOCUS

Prior

$$P(r_{ui}=y) = \frac{(\text{no of items that user has given a rating } y \text{ (AU)}) + \alpha}{(\text{no of total items the user has rated}) + \beta}$$

likelihood

$$P(r_j=k \mid r_i=y) = \frac{(\text{no of items that both user } j \text{ and user } i \text{ have rated as } y) + \alpha}{(\text{no of items that user } j \text{ has given a ratings } y \text{ and for which user } i \text{ has given a rating}) + \beta}$$

AU ki existing ratings

Q) How Probable is the rating 1 for item 5?

*Tell how many decimal places you rounded off

Active User	I_1	I_2	I_3	I_4	I_5
U_1	1	3	3	2	?
U_2	2	4	2	2	4
U_3	1	3	3	5	1
U_4	4	5	2	3	3
U_5	1	1	5	2	1

likelihood

$$P(r_{uj}=1 \mid r_{u1}=1) = P(r_{u2}=1 \mid r_{u1}=1) \times P(r_{u3}=1 \mid r_{u1}=1) \times P(r_{u4}=1 \mid r_{u1}=1) \times P(r_{u5}=1 \mid r_{u1}=1)$$

no. of users U_3, U_5 rated Y / no. of items rated U_1 rated Y and U_j rated something (no question mark)

$$= \frac{0+0.01}{1+0.05} \times \frac{1+0.01}{1+0.05} \times \frac{0+0.01}{1+0.05} \times \frac{1+0.01}{1+0.05} = 0.000,0839$$

(no. of users U_3, U_5 rated 1; no. of items rated U_1 rated 1 and U_2 rated something (no question mark); no. of users U_3, U_4 rated 1; no. of users U_3, U_5 rated 1 → I_1)

$$P(r_{uj}=2 \mid r_{u1}=2) = P(r_{u2}=2 \mid r_{u1}=2) \times P(r_{u3}=2 \mid r_{u1}=2) \times P(r_{u4}=2 \mid r_{u1}=2) \times P(r_{u5}=2 \mid r_{u1}=2)$$

$$= \frac{1+0.01}{1+0.05} \times \frac{0+0.01}{1+0.05} \times \frac{0+0.01}{1+0.05} \times \frac{1+0.01}{1+0.05} = 0.001,762,4$$

$$P(r_{uj}=3 \mid r_{u1}=3) = P(r_{u2}=3 \mid r_{u1}=3) \times P(r_{u3}=3 \mid r_{u1}=3) \times P(r_{u4}=3 \mid r_{u1}=3) \times P(r_{u5}=3 \mid r_{u1}=3)$$

no. of users U_3, U_5 rated 1 / no. of items rated U_1 rated 1 and U_j rated something (no question mark)

$$= \frac{0+0.01}{2+0.05} \times \frac{2+0.01}{2+0.05} \times \frac{0+0.01}{2+0.05} \times \frac{0+0.01}{2+0.05} = 0.000,000,1$$

(no. of users U_3, U_5 rated 3; no. of items rated U_1 rated 3 and U_2 rated something (no question mark); no. of users U_3, U_4 rated 3; no. of users U_3, U_5 rated 3)

$$P(r_{uj}=4 \mid r_{u1}=4) = P(r_{u2}=4 \mid r_{u1}=4) \times P(r_{u3}=4 \mid r_{u1}=4) \times P(r_{u4}=4 \mid r_{u1}=4) \times P(r_{u5}=4 \mid r_{u1}=4)$$

$$= \frac{0+0.01}{0+0.05} \times \frac{0+0.01}{0+0.05} \times \frac{0+0.01}{0+0.05} \times \frac{0+0.01}{0+0.05} = 0.006$$

$$P(r_{uj}=5 \mid r_{u1}=5) = P(r_{u2}=5 \mid r_{u1}=5) \times P(r_{u3}=5 \mid r_{u1}=5) \times P(r_{u4}=5 \mid r_{u1}=5) \times P(r_{u5}=5 \mid r_{u1}=5)$$

$$= \frac{0+0.01}{0+0.05} \times \frac{0+0.01}{0+0.05} \times \frac{0+0.01}{0+0.05} \times \frac{0+0.01}{0+0.05} = 0.006$$

	I_1	I_2	I_3	I_4	I_5
Active User → U_1	1	3	3	2	?
U_2	2	4	2	2	4
U_3	1	3	3	5	1
U_4	4	5	2	3	3
U_5	1	1	5	2	1

Prior

$P(u_1=1) = \frac{1}{4} = 0.25$ (no of items that AU rated 1 / no. of items AU rated)

$P(r_{u_1,i_5}=1|x) = 0.0000839 \times 0.25$
$= 0.000,020,9$

$P(u_1=2) = \frac{1}{4} = 0.25$

$P(r_{u_1,i_5}=2|x) = 0.001,762,4 \times 0.25$ → MAX Hence $U_1 i_5 = 2$ rating
$= 0.000,440,6$

$P(u_1=3) = \frac{2}{4} = 0.5$

$P(r_{u_1,i_5}=3|x) = 0.000,000,1 \times 0.5$
$= 0.000,000,0$

$P(u_1=4) = \frac{0}{4}$ → ?

$P(r_{u_1,i_5}=4|x) = 0.006 \times 0$ → ?
$= 0$

$P(u_1=5) = \frac{0}{4}$

$P(r_{u_1,i_5}=5|x) = 0.006 \times 0$
$= 0$

NAIVE BAYES CF

PROS

- more accurate recommendations
- can provide ranking of predicted ratings (how likely rating 2 can come)
- can provide confidence level for a prediction → be 90% sure prediction is 4
- handles sparse data well
- simple implementation

CONS

- can be computationally intracable
- serendipity can't be controlled
- independence b/w rating is required

Provide brief answers for the following and justify:

a) Why is serendipity so important in recommender systems? How do we control serendipity in item-based Collaborative Filtering?
Answer: *By serendipity we mean recommending items to the user such that his taste is expanded. We can control serendipity in item-based collaborative filtering by adjusting the value of hyperparameter k (as in number of items to be considered in the neighborhood).*

b) How do we handle cases where the ratings are allowed to be real-valued in Naïve Bayes Collaborative Filtering?
Answer: *By assuming the ratings to be normally distributed and using the Gaussian or z-distribution to estimate the likelihood.*

c) Let us say that we have a photography website. The site contains both user-created and AI-generated images. Should our recommender system be giving more importance to user-created images over the AI-generated ones for recommendation? Provide a reason.
Answer: *Although there is no definite answer to this open issue yet, one strong argument is to give preference to AI generated images as often user feedback is important in performance evaluation and improvement of the model.*

d) Can content-based recommender systems solve cold-start problem if a new item is added?
Answer: *Yes. It can solve cold-start problem if a new item is added since this new item can also be recommended to a user based on feature similarity.*

e) If an active user has ***k*** yet unrated items in a system of ***m*** total items, how many vectors would be present in the vector space model at one time?
Answer: *k+1 (one user vector and k item vectors).*

History Based Techniques

- uses historical data to make recommendations
  - collaborative Filtering
  - Content based Filtering
- These techniques
  - require data
  - are not adaptable
  - doesn't ask user for preferences

knowledge based RS

- exploits user requirements
- Used when
  - customers have explicit requirements
  - It's difficult to get range of a specific type of item
  - when ratings are time sensitive
- Application Domain
  - expensive items → not frequently purchased, few ratings (e.g. car, rooms)
  - Time span important → technological product
  - Exploits requirements of user → buying property
- It has 2 types
  1. Constraint Based Systems
     - explicitly defined by conditions
  2. Rule Based Systems
     - similarity to specific requirements

Knowledge-based Systems: Example

- If you're looking for a house or a car online. You input price, how many rooms, how much total floor space, etc., and the website returns a list of houses based on those constraints
- **Issue**:
  - Is this recommendation or simple query search?

Knowledge base: Example

V_C = {kl_c: [expert, average, beginner] ..... /* level of expertise */
wr_c: [low, medium, high] ..... /* willingness to take risks */
id_c: [shortterm, mediumterm, longterm] ..... /* duration of investment */
aw_c: [yes, no] ..... /* advisory wanted ? */
ds_c: [savings, bonds, stockfunds, singleshares] ..... /* direct product search */
sl_c: [savings, bonds] ..... /* type of low-risk investment */
av_c: [yes, no] ..... /* availability of funds */
sh_c: [stockfunds, singlshares] ..... /* type of high-risk investment */ }

V_{PROD} = {$name_p$: [text] ..... /* name of the product */
er_p: [1..40] ..... /* expected return rate */
ri_p: [low, medium, high] ..... /* risk level */
$mniv_p$: [1..14] ..... /* minimum investment period of product in years */
$inst_p$: [text] ..... /* financial institute */ }

Knowledge base: Example

C_R = {CR_1: $wr_c = high \rightarrow id_c \neq shortterm$,
CR_2: $kl_c = beginner \rightarrow wr_c \neq high$}

C_F = {CF_1: $id_c = shortterm \rightarrow mniv_p < 3$,
CF_2: $id_c = mediumterm \rightarrow mniv_p \geq 3 \wedge mniv_p < 6$,
CF_3: $id_c = longterm \rightarrow mniv_p \geq 6$,
CF_4: $wr_c = low \rightarrow ri_p = low$,
CF_5: $wr_c = medium \rightarrow ri_p = low \vee ri_p = medium$,
CF_6: $wr_c = high \rightarrow ri_p = low \vee ri_p = medium \vee ri_p = high$,
CF_7: $kl_c = beginner \rightarrow ri_p \neq high$,
CF_8: $sl_c = savings \rightarrow name_p = savings$,
CF_9: $sl_c = bonds \rightarrow name_p = bonds$ }

C_{PROD} = {$CPROD_1$: $name_p = savings \wedge er_p = 3 \wedge ri_p = low \wedge mniv_p = 1 \wedge inst_p = A$;
$CPROD_2$: $name_p = bonds \wedge er_p = 5 \wedge ri_p = medium \wedge mniv_p = 5 \wedge inst_p = B$;
$CPROD_3$: $name_p = equity \wedge er_p = 9 \wedge ri_p = high \wedge mniv_p = 10 \wedge inst_p = B$}

Conceptual Goals of Recommender Systems

Approach	Conceptual Goal	Input
Collaborative	Give me recommendations based on a collaborative approach that leverages the ratings and actions of my peers/myself.	User ratings + community ratings
Content-based	Give me recommendations based on the content (attributes) I have favored in my past ratings and actions.	User ratings + item attributes
Knowledge-based	Give me recommendations based on my explicit specification of the kind of content (attributes) I want.	User specification + item attributes + domain knowledge