Non-Personalised Recommendations

- Usually recommendations are personalised
- But there are examples of non personalised recommendations
- Product reviews web sites rank items according to the opinions of all the users
- Example: The IMDb Top 250 is intended to be a listing of the top 'rated' 250 films, based on ratings by the registered users of the website

$$W_i = \frac{R_i v_i + Cm}{v_i + m}$$

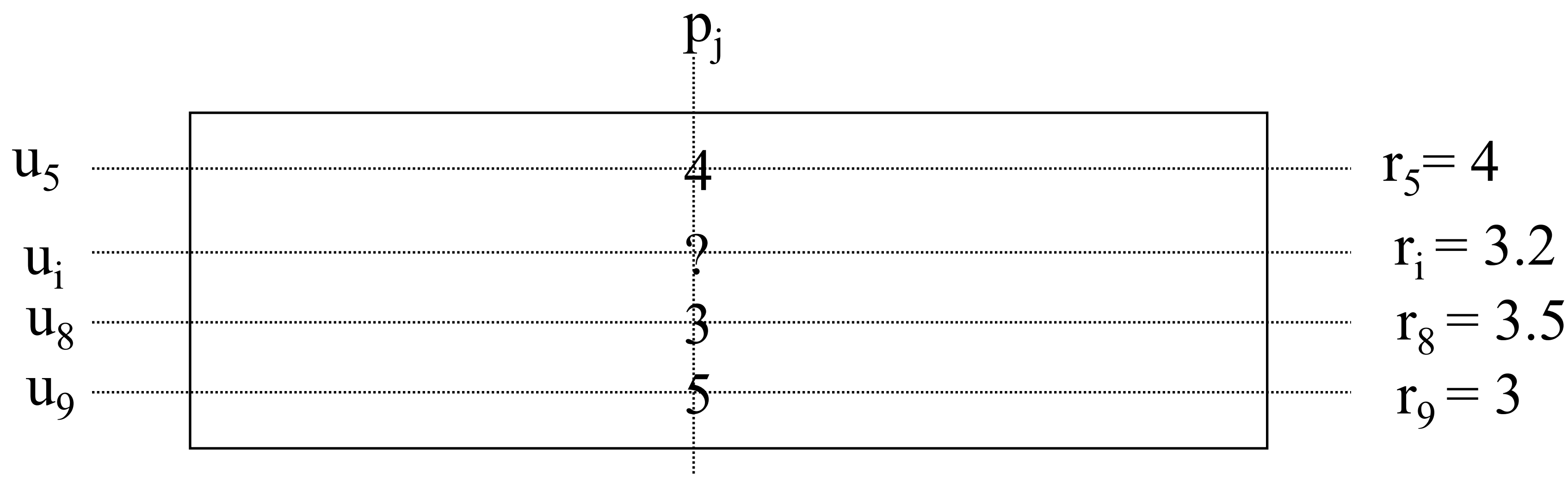
- W_i is the final score of movie i, v_i is the number of reviews for movie i, R_i is the average rating of i, m is a threshold (the number of reviews that a movie must have to enter the top 250 – it is 1300), and C is the average rating of top 250.

http://en.wikipedia.org/wiki/Imdb

IMDb non personalized score

- Compute the non personalized IMDb score for the two movies
  - Casablanca 8.8 (average rating) 132,907 votes
  - Star wars: 8.8 (average rating) 259,593 votes
- Then do it again for the movies
  - Rocky 7.9 (average rating) 63,943 votes
  - Barry Lindon: 7.9 (average rating) 29,259 votes
- C = 8.2452, m=1300
- Write the formula in excel
- Has *Casablanca* a higher score than *Star Wars*? Has *Barry Lindon* a higher score than *Rocky*?

Compute the prediction for r_{ij} in the 3 cases



Users' similarities: $w_{i5} = 0.5$, $w_{i8} = 0.5$, $w_{i9} = 0.8$

$$r^*_{uj} = r_u + K \sum_{v \in N_j(u)} w_{uv}(r_{vj} - r_v)$$

$$K = 1 \Big/ \sum_{v \in N_j(u)} |w_{uv}|$$

$$r^*_{uj} = K \sum_{v \in N_j(u)} w_{uv} r_{vj}$$

$$r^*_{uj} = 1 \Big/ |N_j(u)| \sum_{v \in N_j(u)} r_{vj}$$

$N_j(u)$ is a neighbor of users similar to u who have rated j

$N_j(u_i) = \{u_5, u_8, u_9\}$

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
1	a			1		4	5			4		3					2			4		2			
0,0541	b			4							3							5	1		3				
0	c		5		4			4						3		5					4		5		
0,3212	d								3				5				3			4		2			3
0,132	e		3					5			4	5				5					1			5	4
0,3424	f			4				1		3	5		4	1		5	4	4		4				3	
0,4714	g	2	4				4		2			5		1	4	5		4	2	4		5			4
0,2115	h			2		1		4		3	5		4	2		5	4	5						5	
0,2463	i		1					3			5				5		4	4		5			4		3
0,3664	j			4			4				5			1		5		4		4				4	
0,4584	k		5				4			2		5		1	5		4		2		4				2
0,2996	l					3			3				4	1		4		4	2	4					3
0,4054	m	5		3					5	3		5	4		5	5	3			4	4	5	4		4
0,5151	n			1		4	5				4	5		1	5		4		3		4		4	3	
0,1891	o			4			4				5		4		5			4	2		5		5		3
0,2792	p				4			5								5	4		2	4	4	5	4		2
0,3489	q					3			3					1	5		4	4		4			4		3
0,3023	r		4			1	4		2					2		5		4				5	4		4
0,4446	s			2		4		4			5			1			4		2	4		4		5	
0	t		1		4			3					4		5	5		4			4				3
0,4727	u			2		1		4		3				1		5	4		2	4		5	4		
0,546	v					4	5				4	3		5			2					2			5
0,3145	w				[illegible]			[illegible]		[illegible]			[illegible]		[illegible]	[illegible]	[illegible]		[illegible]	[illegible]		[illegible]	[illegible]		
0	x	4																							
0,2894	y																								

1) Identify the top 3 “most frequent item recommendation” for user a.
The numbers on the left are the similarities with user a.
Base the recommendations on the 4 nearest neighbors.

2) Identify the most reliable neighbors among the 4 nearest neighbor (max overlap)

Evaluating Recommenders

- Recall the definition of Mean Absolute Error MAE
  - Compute MAE for a test set of 100 predictions where the true rating is always 5 but the system predicts: 5 for 60% of the cases, 4 for 30%, and 3 for 10%.
- Recall the notion of precision and the recall of a recommender system
  - Let us assume that in a movie catalogue there are 200 movies that are relevant for user *u*
  - What is the precision and recall of the top-5 recommendations if 4 are relevant for user *u*
  - What is the precision and recall of the top-10 recommendations if 6 are relevant for user *u*
  - Would it be better to show the top-5 or top-10 recommendations? Discuss it.
  - Compare F1 in the two cases.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
a			1		4	5			4		3					2			4		2				
b			4							3							5	1		3					
c		5		4			4						3		5					4		5			
d								3				5				3			4		2			3	
e		3					5			4	5				5					1			5	4	
f			4				1		3	5		4	1		5	4	4		4				3		
g	2	4				4		2			5		1	4	5		4	2	4		5			4	
h			2		1		4		3	5		4	2		5	4	5						5		
i		1					3			5				5		4	4		5			4		3	
j			4			4				5			1		5		4		4				4		
k		5				4			2		5		1	5		4		2		4				2	
l					3			3				4	1		4		4	2	4					3	
m	5		3					5	3		5	4		5	5	3			4	4	5	4		4	
n			1		4	5				4	5		1	5		4		3		4		4	3		
o			4			4				5		4		5			4	2		5		5		3	
p				4			5								5	4		2	4	4	5	4		2	
q					3			3					1	5		4	4		4			4		3	
r		4			1	4		2					2		5		4				5	4		4	
s			2		4		4			5			1			4		2	4		4		5		
t		1		4			3					4		5	5		4			4				3	
u			2		1		4		3				1		5	4		2	4		5	4			
v					4	5				4	3		5			2					2			5	
w				2			2		3			5			4	5		4	2		3	4			
x	4			5				3		3				4	5					1					
y			1			3				2	3						3	3		5		4			

Assume we want to compute the item-to-item CF rating prediction for the product 1 for user n

We must compute the similarity of product 1 with the products bought by user n but we can do that only with products: 11, 14, and 20 (items that have at least 2 users that rated it and prod 1).

Sim(1,11)=0.4, Sim(1,14)=0.3, Sim(1,20)=0.3

What is the final prediction for product 1?

Precision at k

- Suppose to test your RS computing the **precision at k**, i.e., the precision of the top k items retrieved (over a query or averaged over n queries)
- Assume that k = 10, and that in the test set you have 20 users, 10 of them have 5 relevant items in the test set and, the others 10 have 20 relevant items in the test set
- Assume that your RS is "perfect" and always ranks the recommendations such that the relevant items are on the top
- What is the **precision at 10** of your RS? (average the precision at 10 for each user)
- Why it is not 1?

R-precision

- Consider the following definition of **R-precision**
- R-precision for a query (target user) is defined as the precision of the top r elements where r is equal to the number of relevant items for that query
- R-precision for a set of queries is the average R-precision for the queries in the set
- Compute R-precision for the experiment described in the previous slide
- How it scores?
- Is R-precision solving the problem of the precision measure identified in the previous example?
- What is the main difficulty in applying R-precision?

Matrix Factorization with SVD

- Consider the rating matrix shown on slide 34 (part 13)
- Split the available ratings into training and test by assigning to the test set the following ratings: $r_{1\ 1}$, $r_{2\ 3}$, $r_{3\ 4}$, $r_{4\ 5}$, $r_{5\ 11}$, $r_{6\ 3}$
- The remaining ratings are in the training set
- Compute the SVD rank 2 approximation of the matrix with the training data – replace missing values with the item average rating
- Use http://www.bluebit.gr/matrix-calculator/
- Estimate MAE for this model on the test set.

Information Gain

long doc	.edu	Relevant
yes	yes	no
no	no	no
no	no	yes
no	no	yes
no	yes	yes
yes	no	no
no	yes	yes
no	no	no
no	yes	yes
yes	no	yes
no	yes	yes
yes	yes	yes
no	no	yes
yes	no	no

- Consider the data shown in the table
- Select the most informative feature based on Information Gain
- See slide 15 (part 14) for the definitions.

Demographic Methods (more sophisticated)

	gender	age	area code	education	employed	Dolce
Karen	F	15	714	HS	F	+
Lynn	F	17	714	HS	F	–
Chris	M	35	714	C	T	+
Mike	F	40	714	C	T	–
Jill	F	10	714	E	F	?

- Compute the **odds** that Jill will like Dolce given these data using the Naïve Bayes classifier (multivariate)
- Odds(Dolce=+ | Jill) = P(Dolce=+ | Jill) / P(Dolce=- | Jill)
- For estimating class apriori probabilities use frequency counts
- For estimating P(x=v | c), where x is an attribute (e.g., "gender") and c is + or –, use the following formula
- P(x=v | c) = ('# of examples in class c where x=v' + 1) / ('# of examples in c' + '# of possible values for x')
- Possible values for age are 5, for education are 3
- The above method is called "Laplace smoothing".