Collaborative filtering based on k-nearest-neighbours

- Rely on the joint analysis of pasts :
  - suggest to user U items liked by users similar to U (for example)
  - this involves an implicit social network and is called collaborative filtering

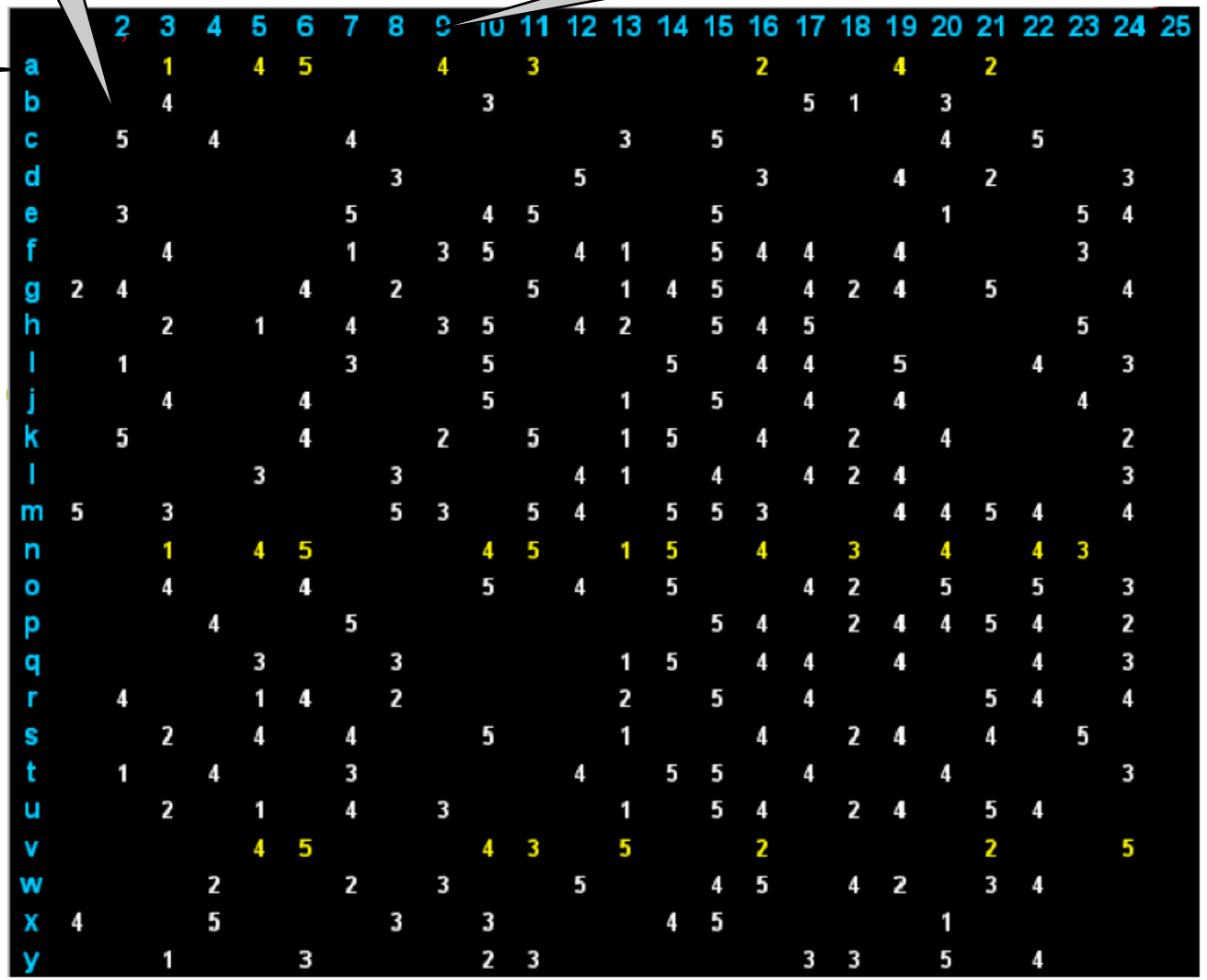
- This technique does not require content descriptors (but it would be possible to integrate them if needed)

Rating in {1,2,3,4,5}

Objects

Users

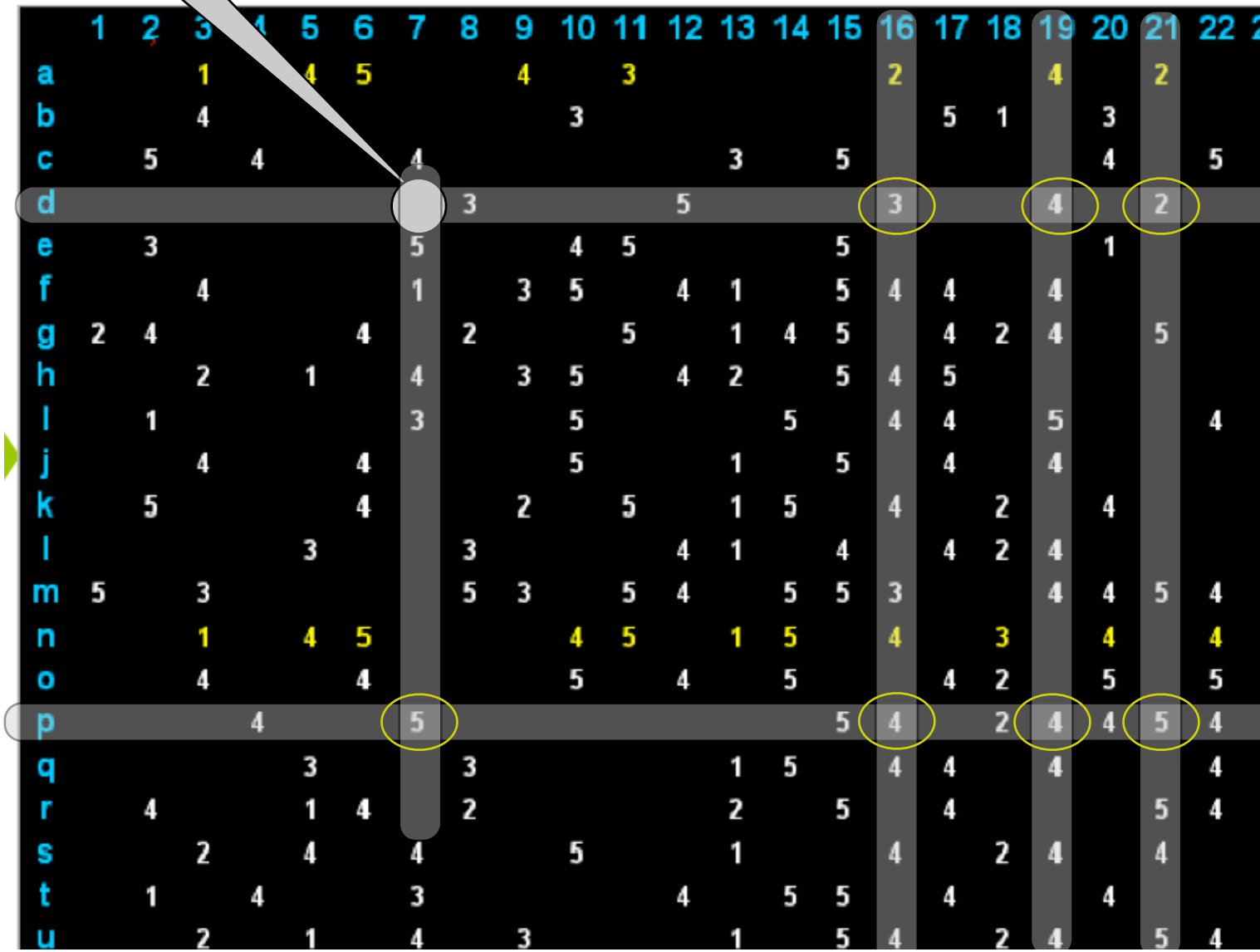
- Must make lines or column « collaborate »
- Example
  - 18 000 films
  - 500 000 users
  - 100 millions ratings
  - how sparse is the matrix ?



	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		
l		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	

Let's try to predict the rating in (d,7)

- Predict (d,7) using the opinion on film of users similar to user d



« user-based » k-nearest neighbours recommendation

Should Charles watch film F2 ?

	F1	F2	F3	F4	F5
Alice	5	1		2	2
Bob	1	5	2	5	5
Charles	2	?	3	5	4
Dédé	4	3	5	3	

Collect all potential neighbours :
=users that have at least rated one film in common with Charles, besides having F2 in common

Charles and Bob seem to have similar tastes

Bob is the nearest neighbour to Charles (using e.g. euclian distance)

So let's predict that Charles would rate F2 with 5

« user-based » k-nearest neighbours recommendation

Should Charles watch film F2 ?

	F1	F2	F3	F4	F5
Alice	5	1		2	2
Bob	1	5	2	5	5
Charles	2	?	3	5	4
Dédé	4	3	5	3	

«ask k unknown-friends instead of 1»

make a combination of these elements of advice

What would possible/suitable combinations ?

« user-based » k-nearest neighbours recommendation

	F1	F2	F3	F4	F5	
Alice	5	1		2	2	?
Bob	1	5	2	5	5	?
Charles	2	?	3	5	4	?
Dédé	4	3	5	3		?
	?	?	?	?	?	

Some users give higher/lower rating on average.
Should we compensate for this ? How ?
Any choice in the similar measures ?
What can the causes for this variability be ?

« item-based » k-nearest neighbours recommendation

	F1	F2	F3	F4	F5
Alice	5	1		2	2
Bob	1	5	2	5	5
Charles	2	?	3	5	4
Dédé	4	3	5	3	

Alternative to « user-based »

Seek film similar to F2
using similarity between columns

Make an example

Can this be combined with « user-based » ?

Exercise :
Reliability of k-nearest neighbours (user or item=objects)

POLYTECH NANTES

- Let us consider a rating matrix (u,o) containing 10000 rating, given by 1000 users on 100 objets.
  Let us assume the probability over the set of (user,object) pairs is uniform.
- Lets us consider :
  - item-based k-nearest-neighbours recommendation
  - user-based k-nearest-neighbours recommendation
- In each case :
  - Calculate the expected number of potential neighbours (i.e. having at least one rating in common)
  - Calculate the expectation of the number of rating in common
- Conclude : should we prefer item-based or user-based knn recommendation ?

Extra questions on nearest neighbors recommendation

	F1	F2	F3	F4	F5
Alice	5	1		2	2
Bob	1	5	2	5	5
Charles	2	?	3	5	4
Dédé	4	3	5	3	

- user-based or item-based for transparency ?
- user-based or item-based for serendipity ?
- how to store sparse matrices, how to make comparisons between sparse vectors ?
- how to choose k ?