Matrix factorization for Netflix recommendation

Yassine Nabou

University Politehnica Bucharest
Faculty of Automatic Control and Computer science

December 2022

Outline

Introduction

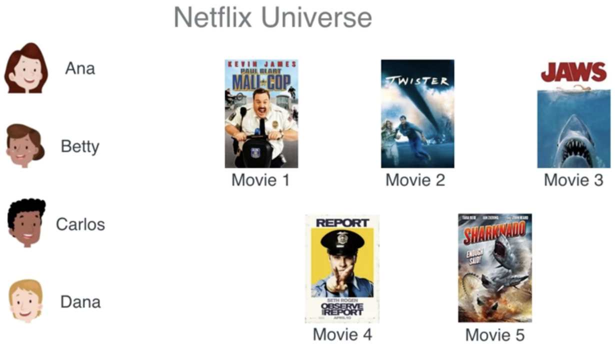
- In this presentation, we will answer to the following question:

How does Netflix recommend movies ?

Netflix universe

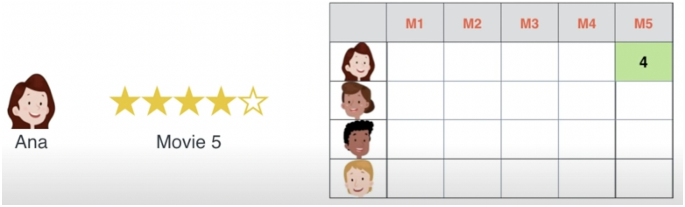
- Netflix universe: A virtual space that contains x number of users and y number of movies.

→ Consider the following example of a Netflix universe:

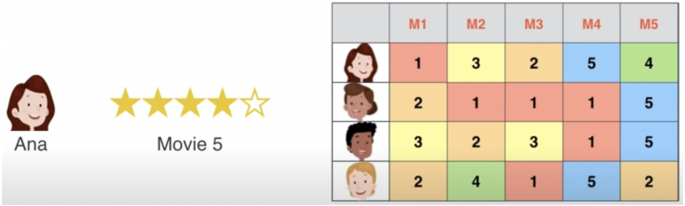


Netflix universe

- Users are going to assign ratings to the movies from 1 to 5 stares:



	M1	M2	M3	M4	M5
					4

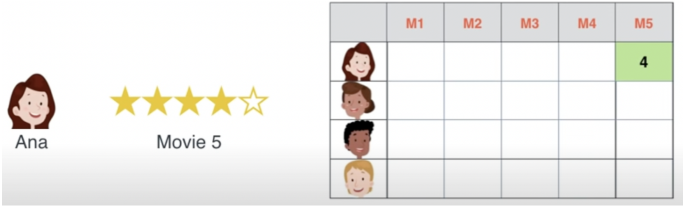


	M1	M2	M3	M4	M5
	1	3	2	5	4
	2	1	1	1	5
	3	2	3	1	5
	2	4	1	5	2

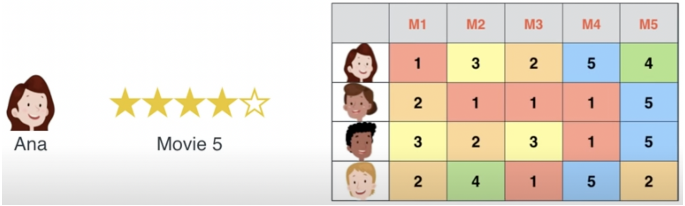
- Goal: predict these ratings.

Netflix universe

- Users are going to assign ratings to the movies from 1 to 5 stares:



	M1	M2	M3	M4	M5
					4

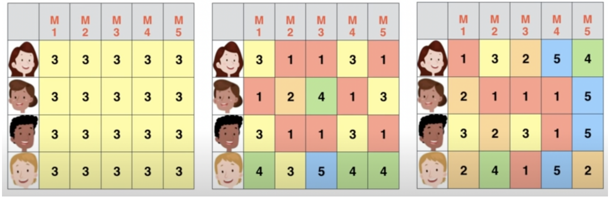


	M1	M2	M3	M4	M5
	1	3	2	5	4
	2	1	1	1	5
	3	2	3	1	5
	2	4	1	5	2

- Goal: predict these ratings.

How do Humans behave ?

- Consider the following three examples of rating:

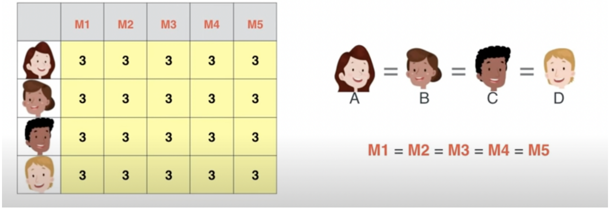


	M1	M2	M3	M4	M5
	3	3	3	3	3
	3	3	3	3	3
	3	3	3	3	3
	3	3	3	3	3

	M1	M2	M3	M4	M5
	3	1	1	3	1
	1	2	4	1	3
	3	1	1	3	1
	4	3	5	4	4

	M1	M2	M3	M4	M5
	1	3	2	5	4
	2	1	1	1	5
	3	2	3	1	5
	2	4	1	5	2

- First table: every user gave every movie a rating of 3:



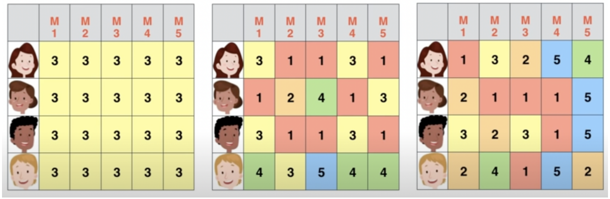
	M1	M2	M3	M4	M5
	3	3	3	3	3
	3	3	3	3	3
	3	3	3	3	3
	3	3	3	3	3

A = B = C = D
M1 = M2 = M3 = M4 = M5

Not realistic: not all human have the same references and all movies are rated he same by very human.

How do Humans behave ?

- Consider the following three examples of rating:

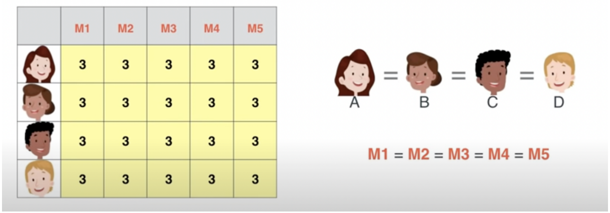


	M1	M2	M3	M4	M5
	3	3	3	3	3
	3	3	3	3	3
	3	3	3	3	3
	3	3	3	3	3

	M1	M2	M3	M4	M5
	3	1	1	3	1
	1	2	4	1	3
	3	1	1	3	1
	4	3	5	4	4

	M1	M2	M3	M4	M5
	1	3	2	5	4
	2	1	1	1	5
	3	2	3	1	5
	2	4	1	5	2

- First table: every user gave every movie a rating of 3:



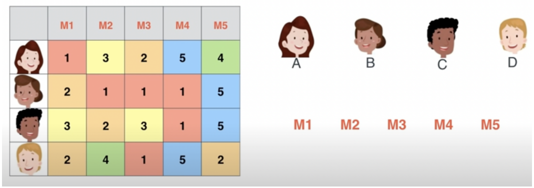
	M1	M2	M3	M4	M5
	3	3	3	3	3
	3	3	3	3	3
	3	3	3	3	3
	3	3	3	3	3

A = B = C = D
M1 = M2 = M3 = M4 = M5

Not realistic: not all human have the same references and all movies are rated he same by very human.

How do Humans behave ?

- Third table: all humans are different in terms of preferences and all the movies are different in terms of ratings by humans

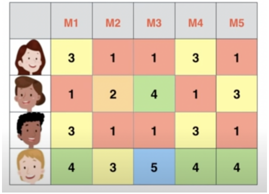


	M1	M2	M3	M4	M5
	1	3	2	5	4
	2	1	1	1	5
	3	2	3	1	5
	2	4	1	5	2

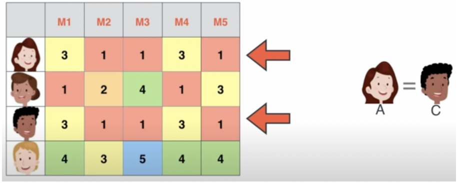
A
B
C
D
M1
M2
M3
M4
M5

Not realistic: Humans sometimes are similar to each other (movies also).

- Second table: more realistic and has a lot of dependency:

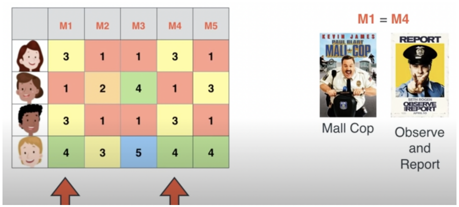


	M1	M2	M3	M4	M5
	3	1	1	3	1
	1	2	4	1	3
	3	1	1	3	1
	4	3	5	4	4



	M1	M2	M3	M4	M5
	3	1	1	3	1
	1	2	4	1	3
	3	1	1	3	1
	4	3	5	4	4

- Ana and Carlos have similar preferences.

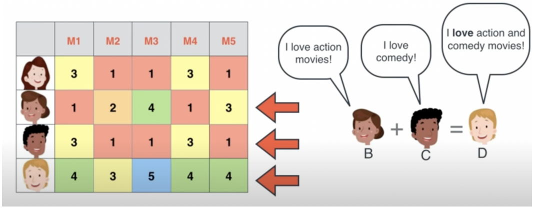


	M1	M2	M3	M4	M5
	3	1	1	3	1
	1	2	4	1	3
	3	1	1	3	1
	4	3	5	4	4

M1 = M4
Mall Cop
Observe and Report

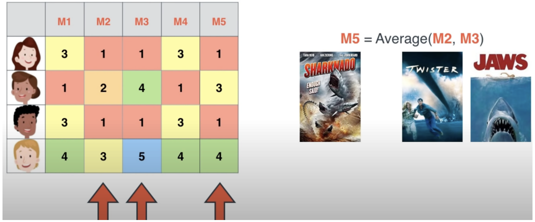
- Movies 1 and 4 are very similar (not the same).

- second row + third row = fourth row:



	M1	M2	M3	M4	M5
	3	1	1	3	1
	1	2	4	1	3
	3	1	1	3	1
	4	3	5	4	4

- We can have also dependencies in columns:

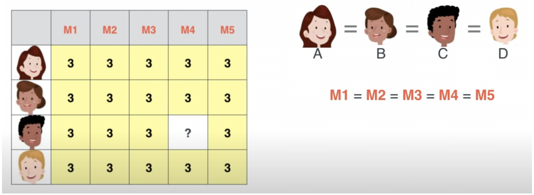


	M1	M2	M3	M4	M5
	3	1	1	3	1
	1	2	4	1	3
	3	1	1	3	1
	4	3	5	4	4

Conclusion: This table has a lot of dependencies → use these dependencies to guess ratings.

Guess ratings

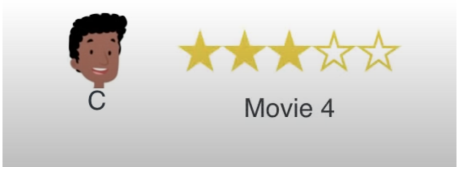
How to use dependencies to guess ratings ? Consider the simple example:



	M1	M2	M3	M4	M5
	3	3	3	3	3
	3	3	3	3	3
	3	3	3	?	3
	3	3	3	3	3

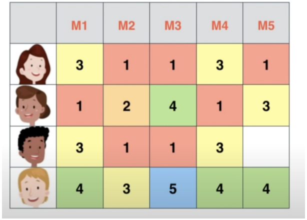
A = B = C = D
M1 = M2 = M3 = M4 = M5

Users have the same references and every movie s the same, but Carlos hasn't watched movie 4. Since all the users are the same, then:



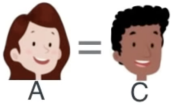
Guess ratings

Consider the following table of rating:

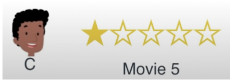


	M1	M2	M3	M4	M5
	3	1	1	3	1
	1	2	4	1	3
	3	1	1	3	
	4	3	5	4	4

Since we have:



Then, it is safe to assume that Carlos will gave the rate:

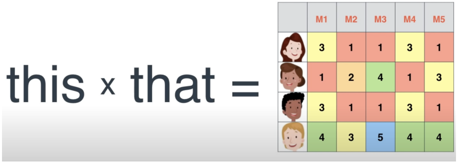


Question: How do we figure out all these dependencies ?

Answer: Matrix factorization.

Matrix factorization

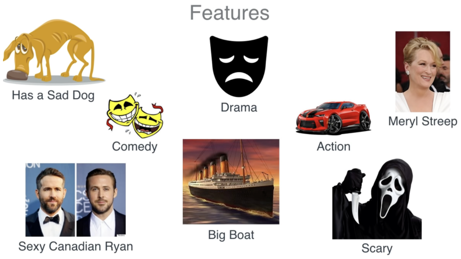
- The goal is to have two small matrices that multiply to the big one:



this x that =

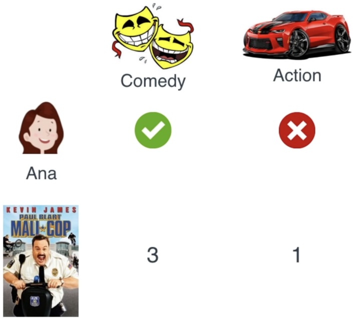
	M1	M2	M3	M4	M5
	3	1	1	3	1
	1	2	4	1	3
	3	1	1	3	1
	4	3	5	4	4

- In order to find this two matrices, we will use features:



Dot product

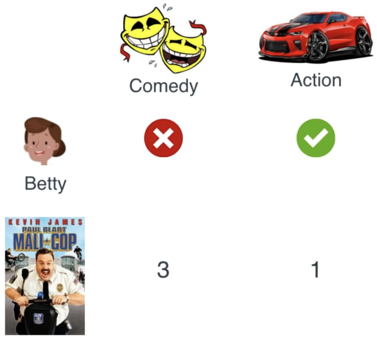
- For example, consider two features: comedy and action.
- A dot product is a way to guess rating based on how much user likes comedy and action and how much movie contains comedy and action.
- First example: assume that we figured out that Ana likes comedy and she doesn't like action, and also we managed to rate every movie based on how much comedy and how much action they have (mockup has 3 in comedy and 1 in action):



Dot product

We will figure out how many stares Ana gave mockup: since Ana likes comedy and doesn't like action, then we will say that Ana gives 3 stares to mockup.

- Second example:

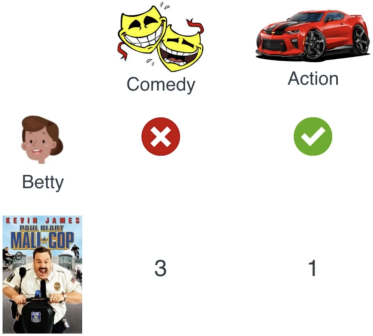


Betty gives 1 start to mockup

Dot product

We will figure out how many stares Ana gave mockup: since Ana likes comedy and doesn't like action, then we will say that Ana gives 3 stares to mockup.

- Second example:



Betty gives 1 start to mockup

Dot product

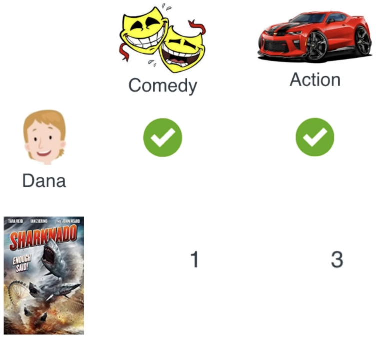
- Third example:

	Comedy	Action
Dana	✓	✓
	1	3

Dana gives $1 + 3 = 4$ start to sharknado.

Dot product

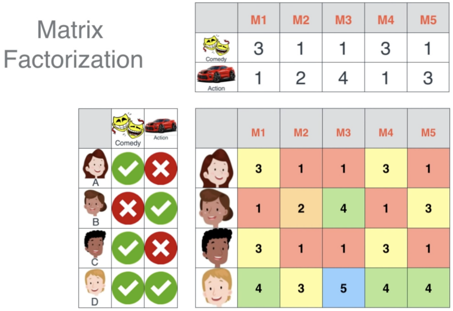
- Third example:



Dana gives $1 + 3 = 4$ start to sharknado.

Dot product

- Thus, using the dot product we have the following factorization:



Matrix Factorization

	M1	M2	M3	M4	M5
Comedy	3	1	1	3	1
Action	1	2	4	1	3

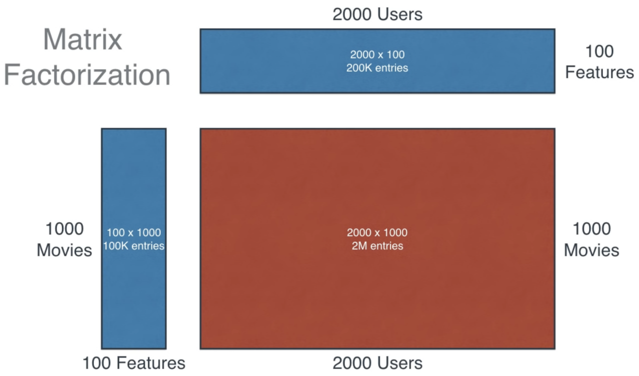
	Comedy	Action
A	✓	✗
B	✗	✓
C	✓	✗
D	✓	✓

	M1	M2	M3	M4	M5
	3	1	1	3	1
	1	2	4	1	3
	3	1	1	3	1
	4	3	5	4	4

where in top we have columns are movies and rows are features and on the left we have one with columns re features and rows are users.

- All the dependencies that we found previously among the rows and columns are found here too in an easier way: can you find the dependencies ?

Matrix factorization and storage



- huge saving of space.

The right factorization

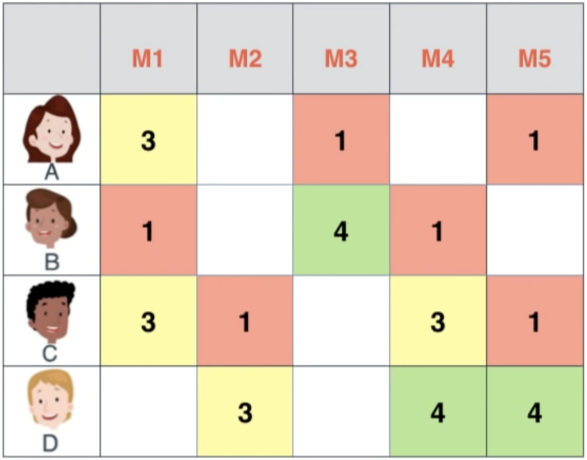
- In order to find the right factorization, we need to solve the following optimization problem:

$$\min_{U,V} \frac{1}{2}\|\mathcal{P}(UV - X)\|_F^2,$$

  where $\mathcal{P}(UV - X)_{i,j} = (UV - X)_{i,j}$ if $X_{i,j}$ is known and 0 else, $\|A\|_F = \sqrt{\text{trace}(A^T A)} = \sqrt{\sum_{i,j} A_{i,j}^2}$ and X is the given matrix of rating.
- Several algorithm can solve the optimization problem above: Gradient descent, alternating minimization,...

Predict the ratings

- Matrix factorization helps us with storage, but also predict rating. The Netflix matrix X is very sparse (not every user rates every movie):



	M1	M2	M3	M4	M5
A	3		1		1
B	1		4	1	
C	3	1		3	1
D		3		4	4

Predict the ratings

- Assume that, using gradient descent, we manage to fine U, V:

	M1	M2	M3	M4	M5
F1	3	1	1	3	1
F2	1	2	4	1	3

	F1	F2
A	1	0
B	0	1
C	1	0
D	1	1

	M1	M2	M3	M4	M5
A	3		1		1
B	1		4	1	
C	3	1		3	1
D		3		4	4

Predict the ratings

- Using the dot product we can fill in the matrix X:

	M1	M2	M3	M4	M5
F1	3	1	1	3	1
F2	1	2	4	1	3

	F1	F2
A	1	0
B	0	1
C	1	0
D	1	1

	M1	M2	M3	M4	M5
A	3	1	1	3	1
B	1	2	4	1	3
C	3	1	1	3	1
D	4	3	5	4	4

Now, if Dana comes in and she hasn't seen movie 1 and 3, which one do we recommend ? From the table, the highest rating is 3, then we recommend movie 3 to Dana.

Thank You!

This presentation is based on the following resource:
https://www.youtube.com/watch?v=ZspR5PZemcs