EASSS 2023
Recommender Systems

Rodrigo da Silva Alves

July 19, 2023

Recommender Systems

" **Recommendation:** a suggestion or proposal as to the best course of action, especially one put forward by an authoritative body."

— Oxford Languages

" **System:** 1 . a set of things working together as parts of a mechanism or an interconnecting network; a complex whole. *2. a set of principles or procedures according to which something is done; an organized scheme or method."*

— Oxford Languages

Recommendation

- Recommendations have existed since ancient times.
- Societies have benefited from following the advice of their ancestors.
- For example, priests have traditionally provided guidance to people.
- However, these examples are not based on systematic tools.

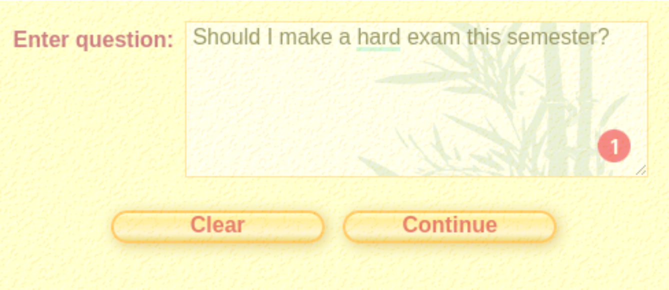
What is considered the first systematic tool created to provide advice to people about their future?

Fu Xi

- Fu Xi is a mythical emperor of China
- He taught his people how to domesticate animals, cook, fish with nets, and hunt with iron weapons over 5000 years ago.
- He is the author o I Ching (*The book of changes*), one of the oldest book in the world.
  - Guide for moral living but also a personalised oracle for one's future!
  - Instead of machine learning, he used patterns in nature and sky.

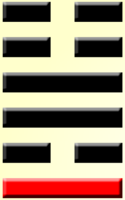
I Ching

- You can do your I Ching consult online
- I did one, let's see... I was in doubt if the exam for this course should be hard or easy. Then, I asked for these divine recommender what to do:



I Ching - Hexagram

I Ching - Recommendation



Cast Hexagram: 55

Cast Hexagram:

55 - Fifty-Five

Fêng / Abundance

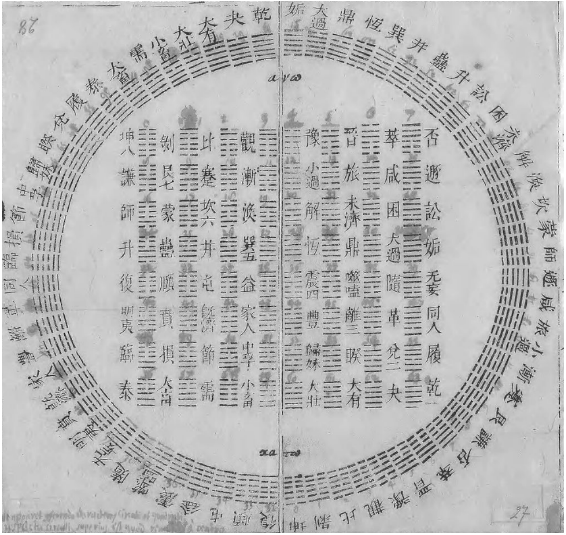
Thunder and Lightning from the dark heart of the storm:
The Superior Person judges fairly, so that consequences are just.

The leader reaches his peak and doesn't lament the descent before him.
Be like the noonday sun at its zenith.
This is success.

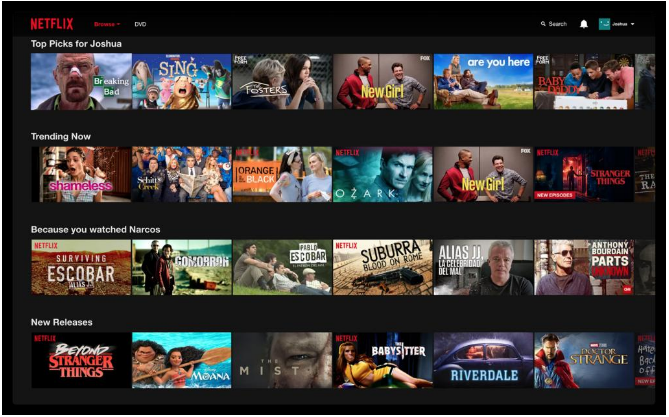
SITUATION ANALYSIS:

You are in a position of authority in this situation.
Archetypally, you are the New King, returned from your quest to claim your throne.
However, you are enlightened enough to realize that you are merely a part of a cycle, and that you must someday yield your throne to the new kid in town, the younger, faster gunslinger, the young turk, the next returning hero, the next New King.
Fretting about the inevitable descent is senseless.
For now you must play your role to the hilt and use this gift of power to govern your world as best you can.
You are the best person for the job.
That's why you were chosen.
Give it your personal best.

I Ching



Nowadays...



NETFLIX
Browse
DVD
Search
Top Picks for Joshua
Breaking Bad
SING
THE FOSTERS
New Girl
are you here
BABY DADDY
Trending Now
shameless
Schitt's Creek
ORANGE is the new BLACK
OZARK
New Girl
STRANGER THINGS
NEW EPISODES
Because you watched Narcos
SURVIVING ESCOBAR
ALIAS J.J.
GOMORRAH
PABLO ESCOBAR
SUBURRA
BLOOD ON ROME
ALIAS J.J.
LA CELEBRIDAD DEL MAL
ANTHONY BOURDAIN
PARTS UNKNOWN
New Releases
BEYOND STRANGER THINGS
MOANA
THE MIST
THE BABYSITTER
RIVERDALE
DOCTOR STRANGE

Nowadays...

Nejčastěji dohromady zakoupené zboží a příslušenství
Další příslušenství »
4,6 5x
ro Kožený kryt
ově zelený
790,-
o košíku
zdra a kryty na
obil »
4,6 5x
Apple iPhone 14 Pro Kožený kryt
s MagSafe cihlově hnědý
1 790,-
Do košíku
Obaly, pouzdra a kryty na
mobil »
4,8 479x
Apple 20W USB-C napájecí
adaptér
599,-
Do košíku
Nabíječky na mobily »
NEO
PRONÁJEM
4,9 127x
Apple AirPods Pro 2022
ZLEVNĚNO -4 %
6 990,-
7 290,-
Do košíku
Sluchátka »
NEO
PRONÁJEM
4,7 34x
Apple Watch Series 8 45mm
Temně inkoustový hliník s temn...
12 190,-
Do košíku
Apple Watch 8 »
4,7 150x
Apple Nabíječka MagSafe
949,-
Do košíku
Nabíječky na mobily »

Why recommender systems?

It is not just about youtube...

- 3,000,000 books per year (Total of 129,864,880 available)
- 137 million new tracks every year
- 500 million tweets are shared every day
- You have 38 million minutes of live
- Conclusion: we need some help

Recommender Systems

- Recommenders recommend:
  - Items to users (most common)
  - Users to Items
  - Items to Items
  - Users to Users
- Items are movies, products, news, music, books, recipes, ...

Working in pairs: try to find one example of each four recommender scenarios above.

Recommender Systems

- WLOG, we will focus on recommend **users** relevant **items**
  - Predictive modelling: predict the rate of item m by the user u
  - Retrieve modelling: learning a rank systems
- Typically based on past **interactions** and (or) **attributes** (from users and items)
- **Interactions:** normally modelled as interaction matrix
  - Explicit: a user rate a song with 4 stars in a scale from 0 to 5
  - Implicit: a user watched 80% of a movie
- **Attributes:** normally modelled as attribute matrices
  - Users: gender, age, location, $\cdots$
  - Items: text, video, meta-data, $\cdots$

Modelling Interactions: Explicit feedback

	m_1	m_2	...	m_n
u_1	?	2	...	3
u_2	5	1	...	?
u_3	?	3	...	1
...	...	...	...	...
u_m	4	4	...	?

Modelling Interactions: Implicit feedback

u_1	m_1	12/11/2021 09:01:21	Watch	25%
u_2	m_1	17/03/2021 14:27:09	Clicked	
u_2	m_4	17/03/2021 14:22:09	Clicked	Purchase
...	...	...	...	...
u_m	m_n	14/06/2020 23:14:46	Watch	100%

Modelling Interactions: Implicit feedback

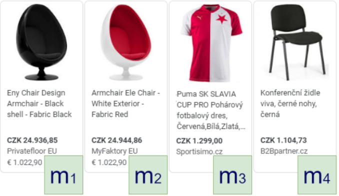
u_1	m_1	12/11/2021 09:01:21	Watch	25%
u_2	m_1	17/03/2021 14:27:09	Clicked	
u_2	m_4	17/03/2021 14:22:09	Clicked	Purchase
...	...	...	...	...
u_m	m_n	14/06/2020 23:14:46	Watch	100%

	m_1	m_2	...	m_n
u_1	1	0	...	0
u_2	1	0	...	0
u_3	0	1	...	1
...	...	...	...	...
u_m	0	1	...	1

Modelling Attributes

Modelling Attributes

	Color	Price	Category
m_1	Black	24936	Chair
m_2	Red and White	24944	Chair
m_3	Red and White	1299	T-Shirt
m_4	Black	1104	Chair



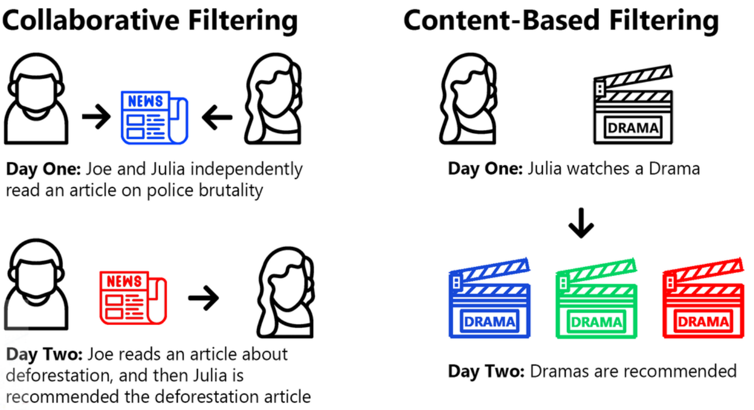
Personalised Machine Learning

- Personalisation *is not* a simple regression or classification problem
- A personalised model implies that: if the user have different interactions (or attribute) the recommendation should be different
- Suppose the vector a_u (a_m) are attribute vectors of user u (item m)
- We can use linear regression to predict how users u will like item m

$$r_{ui} = \omega^\top \times \begin{bmatrix} a_u \\ a_m \end{bmatrix}$$

- Is linear regression a personalised model for recommenders? No!

Recommendation Algorithms



Recommender as a Matrix

- As we saw, we can model recommenders as matrices.
- The ratings can be stored in a **ranking matrix** R of dimension $m \times n$ with elements from $\mathbb{R} \cup \{?\}$.
- An example of a rating matrix for $m = 4$ users and $n = 6$ items can read

$$R = \begin{pmatrix} 1 & ? & ? & 2 & ? & 1 \\ ? & 2 & 3 & ? & 2 & 1 \\ 1 & 5 & 5 & ? & ? & 5 \\ ? & ? & 2 & ? & ? & 3 \end{pmatrix}.$$

  Meaning, e.g., that user u_1 ranked items i_1 and i_6 with 1 star, item i_4 with 2 stars and had no interactions with items i_2, i_3 and i_4.
- **Our goal is to predict the unknown ratings** $r_{u,i} =?$ using the knowledge of the known ratings $r_{u,i} \neq ?$.

Idea of matrix factorization

- By **matrix factorization** we usually mean expressing a given matrix R as a matrix product of two (or more) matrices with some non-trivial properties. For example:

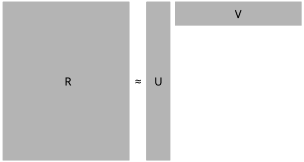
$$R = UV^\top$$

- These factorizations are a cornerstone of many algorithms and methods or are used to reach more numerically stable computations.

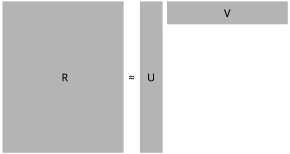
Do we need to know all the entries of a matrix R to factorize it, for example $R = UV^\top$?

Intuition of matrix factorization

- As for the recommendation systems, the inspiration comes mainly from the SVD as it can be used for constructing **latent features** or, in other words, **dimensionality reduction** using projections to lower dimensional space.
- The very basic idea of the lower dimensional approximation of an input matrix R of dimension $m \times n$ is based on this first-linear-algebra-lesson fact: Multiplying matrices U of dimension $m \times d$ and V of dimension $d \times n$ we get a matrix of dimension $m \times n$. This is true for any positive integer d.
- And this is the idea: **Given a rating matrix R, find lower dimensional matrices U and V so that the known elements of R are well approximated by the matrix $\mathbf{UV}^\top$.**



Matrix factorization for recommenders



- Let us denote:
  - The i-th row of U as u_i; the number of rows of U equals the number of users $|\mathcal{U}|$.
  - The j-th column of V as v_j; the number of columns of V equals the number of items $|\mathcal{I}|$.
  - Ω the subset of $\mathcal{U} \times \mathcal{I}$ of user-item pairs (i, j) such that $r_{i,j}$ is known, i.e., $r_{i,j} \neq ?$.
- The approximation of $r_{i,j}$ is given by the number $u_i^T v_j$, i.e., by the dot product of the two d-dimensional vectors.

Optmization Problem

- The **error of approximation** is usually measured by the squared residual:

$$(r_{i,j} - u_i^T v_j)^2.$$

- Hence, the matrices U and V are obtained by solving the optimization task:

$$\operatorname{argmin}_{\mathbf{U},\mathbf{V}} \sum_{(i,j)\in\Omega} (r_{i,j} - u_i^T v_j)^2 + \lambda(\sum_x ||u_x||^2 + \sum_y ||v_y||^2).$$

Sparsity and prediction

- The matrices U and V are optimized only by considering the known entries of R that are usually only a minority of entries.
- E.g. in the Netflix prize in 2006 there were $n = 17K$ movies and $m = 500K$ users, meaning that the matrix R had $8500M$ entries. But only $100M$ was given by Netflix!
- Still, the result of the matrix multiplication $UV^\top$ is a matrix having the same dimensions as R **with all entries known**!
- The unknown rating $r_{i,j} = ?$ is estimated as $\hat{r}_{i,j} = u_i^T v_j$..

Example

- Consider our toy example matrix from above:

$$R = \begin{pmatrix} 1 & ? & ? & 2 & ? & 1 \\ ? & 2 & 3 & ? & 2 & 1 \\ 1 & 5 & 5 & ? & ? & 5 \\ ? & ? & 2 & ? & ? & 3 \end{pmatrix}.$$

- Assume that we chose the hyperparameter $d = 2$, i.e., we look for approximation matrices U and V with dimensions 4×2 and 2×6, respectively.
- Let us pretend that the matrices resulting from the optimization are

$$U = \begin{pmatrix} 0.3 & 0.7 \\ 0.3 & 0.5 \\ 0.2 & 0.4 \\ 0.2 & 0.1 \end{pmatrix} \quad \text{and} \quad V^\top = \begin{pmatrix} 1 & 10 & 11 & 10 & 4 & 20 \\ 1 & -1 & -2 & -1 & 1 & -4 \end{pmatrix}.$$

Example

- The resulting approximation is

$$\mathbf{U}\mathbf{V}^\top = \begin{pmatrix} 0.3 & 0.7 \\ 0.3 & 0.5 \\ 0.2 & 0.4 \\ 0.2 & 0.1 \end{pmatrix} \begin{pmatrix} 1 & 10 & 11 & 10 & 4 & 20 \\ 1 & -1 & -2 & -1 & 1 & -4 \end{pmatrix} =$$

$$= \begin{pmatrix} 1 & \color{red}{2.3} & \color{red}{1.9} & 2.3 & \color{red}{1.9} & 3.2 \\ \color{red}{0.8} & 2.5 & 2.3 & \color{red}{2.5} & 1.7 & 4 \\ 0.6 & 1.6 & 1.4 & \color{red}{1.6} & \color{red}{1.2} & 2.4 \\ \color{red}{0.3} & \color{red}{1.9} & 2 & \color{red}{1.9} & \color{red}{0.9} & 3.6 \end{pmatrix},$$

where the red numbers are the desired predictions!

- E.g. the 3rd user predicted rating of the 4th item is $\hat{r}_{3,4} = 1.6$.

Supervised learning task

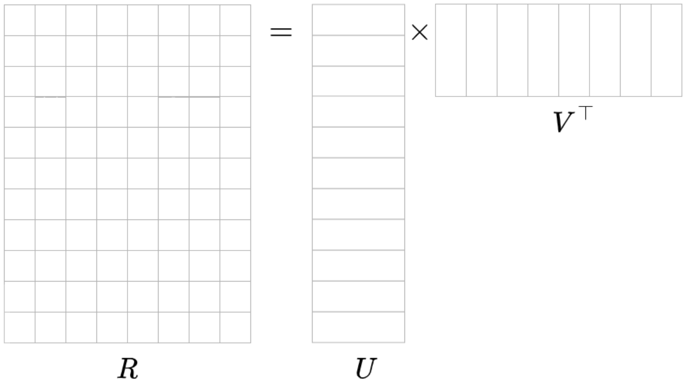
- The learning parameters: $U \in \mathbb{R}^{m \times d}$ and $V \in \mathbb{R}^{n \times d}$
- The hyperparameters:
  - the regularization constant $\lambda > 0$,
  - the matrix dimension d, which is a positive integer (significantly smaller than $\min\{m, n\}$).
- These hyperparameters can be tuned in the usual way via crossvalidation
- Therefore we would like to learn U and V, given d and λ by

$$\mathrm{argmin}_{\mathbf{U},\mathbf{V}} \sum_{(i,j)\in\Omega} (r_{i,j} - u_i^T v_j)^2 + \lambda\left(\sum_x ||u_x||^2 + \sum_y ||v_y||^2\right).$$

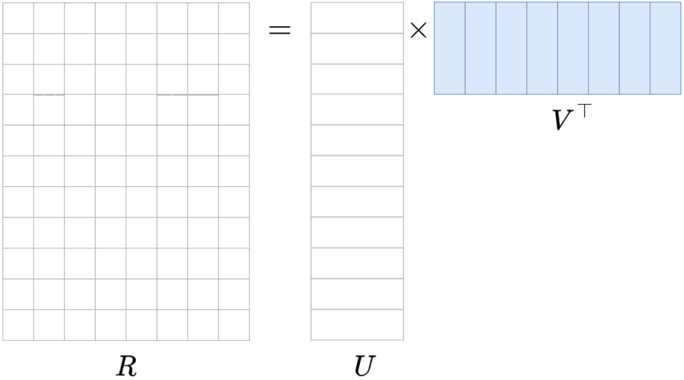
Alternating least squares (ALS)

- The idea of ALS is to fix alternately the matrix U and V. The non-fixed matrix is then considered learning variable and a subject to minimization.
- With one of the matrices fixed, the optimization problem becomes convex and very similar to the linear regression problem.
- Let's try ti understand how the mechanism works

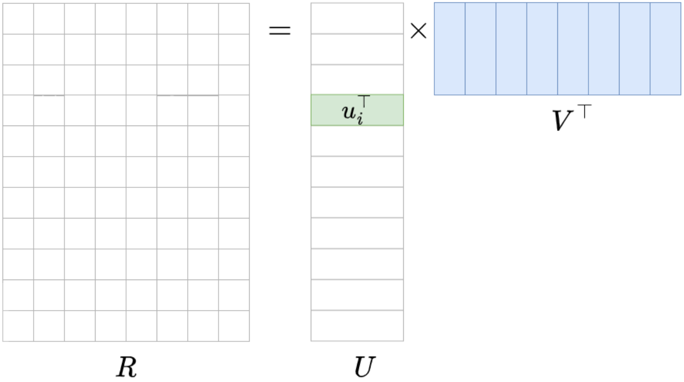
Alternating least squares (ALS)



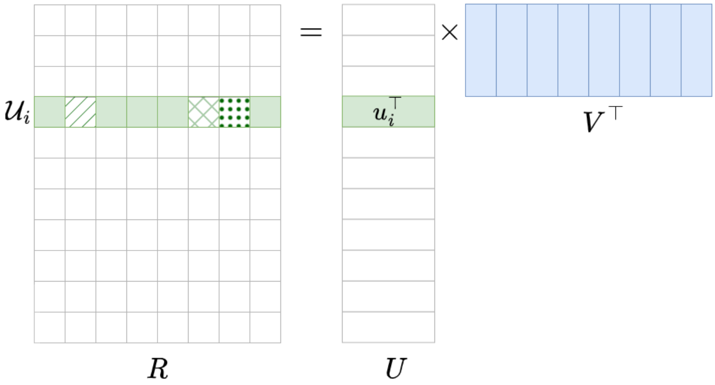
Alternating least squares (ALS)



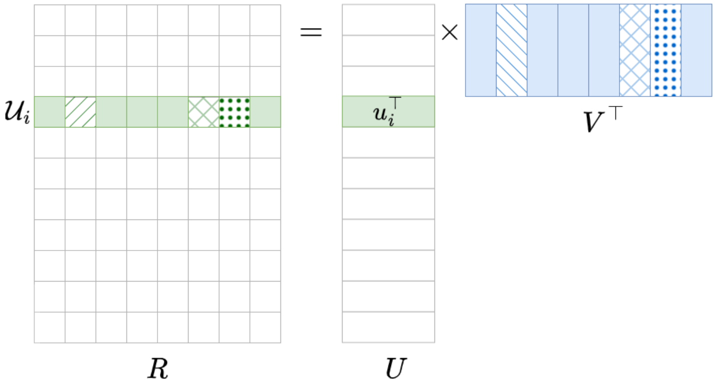
Alternating least squares (ALS)



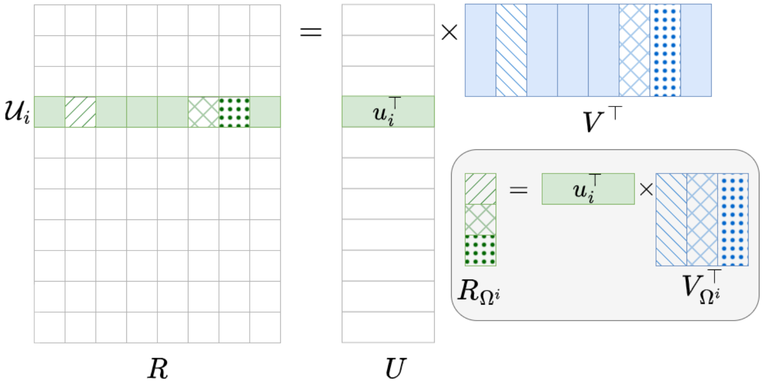
Alternating least squares (ALS)



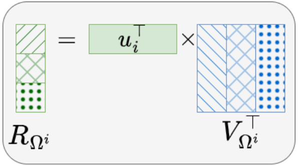
Alternating least squares (ALS)



Alternating least squares (ALS)



Alternating least squares (ALS)



- Then we have the following optimization problem

$$\min_{u_i} ||R_{\Omega^i} - u_i{}^\top V_{\Omega^i}{}^\top||^2 + \lambda ||u_i||^2$$

- Convex problem with closed-form

$$\hat{u}_i = (V_{\Omega^i} V_{\Omega^i}{}^\top + \lambda I)^{-1} V_{\Omega^i}^\top R_{\Omega^i}$$

Alternating least squares (ALS)

Randomly initialize U and V

- **WHILE** does not converge
  - $\forall i \in \mathcal{U}$, $\min_{u_i} ||R_{\Omega^i} - u_i{}^\top V_{\Omega^i}{}^\top||^2 + \lambda ||u_i||^2$
  - $\forall j \in \mathcal{I}$, $\min_{v_j} ||R_{\Omega^j} - v_j{}^\top U_{\Omega^j}{}^\top||^2 + \lambda ||v_j||^2$

MF for Implicit Feedback

- In real-world applications, we often observe more implicit feedback than explicit feedback.
- In fact, explicit feedback is sometimes considered implicit.
- Suppose user i watched 35% of movie A and 85% of movie B.

Does this mean that the user likes A more than B? If so, does it mean that the user likes A more than twice as much as B?

- The method we learned before is more appropriate for explicit feedback. Why?

Modelling Implicit Feedback

- Let's understand a more appropriate method
- Assume the binary interaction matrix P:

$$P = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}.$$

- That is, if user-i interact with item-j, than $P_{ij} = 1$, otherwise $P_{ij} = 0$.
- Now let C be a matrix of confidence regarding the interaction:

$$C = \begin{pmatrix} 0.85 & 0 & 0 & 0.34 & 0 & 0.98 \\ 0 & 0.37 & 0.10 & 0 & 0.63 & 0.01 \\ 0.45 & 0.42 & 0.43 & 0 & 0 & 0.23 \\ 0 & 0 & 0.26 & 0 & 0 & 0.88 \end{pmatrix}.$$

Collaborative Filtering for Implicit Feedback

- Then we propose the following optimisation problem:

$$\text{min}_{U,V} \sum_{i,j} C_{ij}(P_{ij} - \boldsymbol{u}_i^\top \boldsymbol{v}_j)^2 + \lambda||\boldsymbol{u}_i||^2 + \lambda||\boldsymbol{v}_j||^2$$

- Two main differences from previous MF method:
  - We need to account for the varying confidence levels
  - Optimization should account for all possible j, j pairs, rather than only those corresponding to observed data.
- We can use gradient descent to solve it.
- And ALS? By fixing V, can we find u_i?

Closed form

- Assume V being fix and let's find u_i.
- Then we need to minimize the following loss

$$\mathcal{L}_i = \text{min}_{u_i} \sum_j C_{ij}(P_{ij} - u_i^\top v_j)^2 + \lambda ||u_i||^2$$

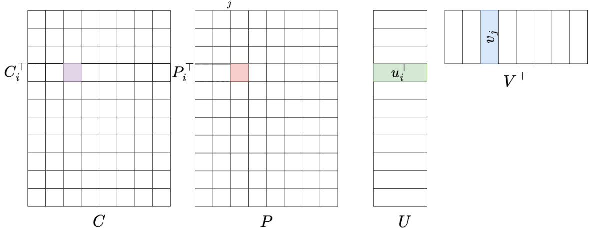
That is the same of:

$$\mathcal{L}_i = \text{min}_{u_i} \sum_j (\sqrt{C_{ij}}(P_{ij} - u_i^\top v_j))^2 + \lambda ||u_i||^2$$

Exercise: Find the closed form.

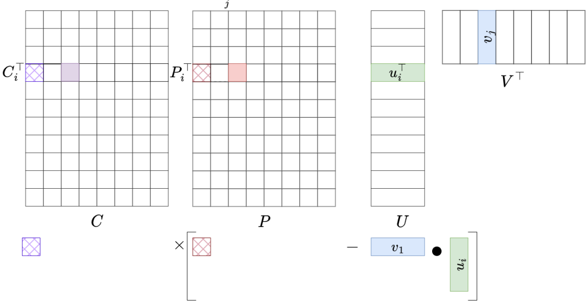
Alternating least squares (ALS)

$$\mathcal{L}_i = \min_{u_i} \sum_j (\sqrt{C_{ij}}(P_{ij} - u_i^\top v_j))^2 + \lambda ||u_i||^2$$



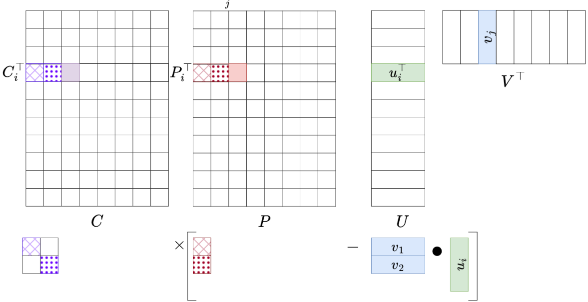
Alternating least squares (ALS)

$$\mathcal{L}_i = \min_{u_i} \sum_j (\sqrt{C_{ij}}(P_{ij} - u_i^\top v_j))^2 + \lambda ||u_i||^2$$



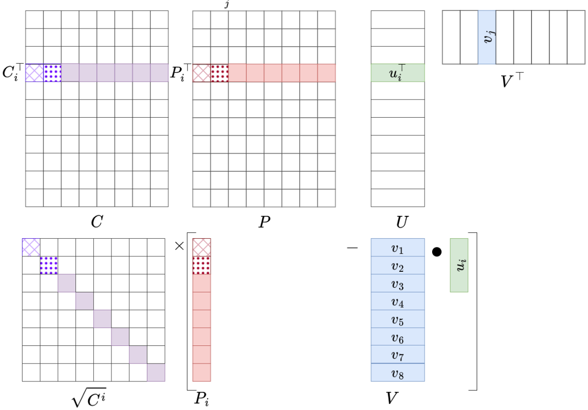
Alternating least squares (ALS)

$$\mathcal{L}_i = \min_{u_i} \sum_j (\sqrt{C_{ij}}(P_{ij} - u_i^\top v_j))^2 + \lambda ||u_i||^2$$



Alternating least squares (ALS)

$$\mathcal{L}_i = \min_{u_i} \sum_j (\sqrt{C_{ij}}(P_{ij} - u_i^\top v_j))^2 + \lambda ||u_i||^2$$



Closed form

- Therefore is the same of solving:

$$\mathcal{L}_i = ||\sqrt{C^i}P_i - \sqrt{C^i}Vu_i||^2 + \lambda + ||u_i||^2$$

- Taking the derivative

$$\nabla u_i = -2(\sqrt{C^i}V)^\top(\sqrt{C^i}P_i - \sqrt{C^i}Vu_i) + 2\lambda u_i$$

- Remind if D is diagonal $D = \sqrt{D} \times \sqrt{D}$ is trivial and $D = D^\top$
- Therefore, with just some algebraic derivations

$$u_i = (V^\top C^i V + \lambda I)^{-1} V^\top C^i P_i$$

RECAP: Autoencoder

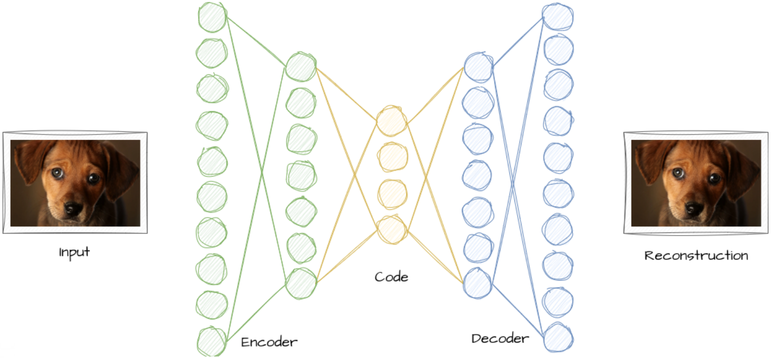
- An autoencoder is a type of feed-forward neural network
- It is designed to reconstruct its input x_i ad output x_i
- To prevent trivial solutions, the network includes a bottleneck (or code) layer
  - Significantly smaller dimension than the input

RECAP: Autoencoder

- An autoencoder is also composed by a encoder/decoder
- The encoder and the decoder have normally similar structure
- More formally: let $\mathcal{E}()$ be a encoder and $\mathcal{D}()$ be a decoder. Our optmization problem can be described as:

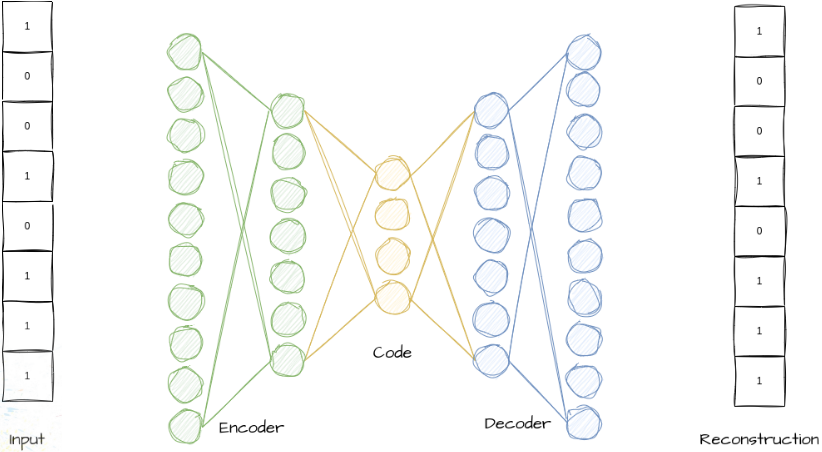
$$\min_{\mathcal{E},\mathcal{D}} \sum_i ||x_i - \mathcal{D}(\mathcal{E}(x_i))||$$

Autoencoder



How can we use autoencoders to predict implicit feedback?

Autoencoders for CF



Autoencoders for CF

- Autoencoders are frequently used for collaborative filtering.
- They are very accurate in predicting rankings.
- They can also be used to find clusters with the code.
- Empirical results show that the best architecture is often not very deep.
- What would be the shallowest autoencoder for Collaborative Filtering?

EASE

- EASE is the shallowest auto-encoder as possible
- It aims to solve the following problem

$$\text{min}_B ||X - XB||^2 + \lambda ||B||^2 \text{ s.t } \text{diag}(B) = 0$$

- Why do we need the constraint $\text{diag}(B) = 0$?
- EASE has closed form solution! See here
- Is this a good method?

EASE Results

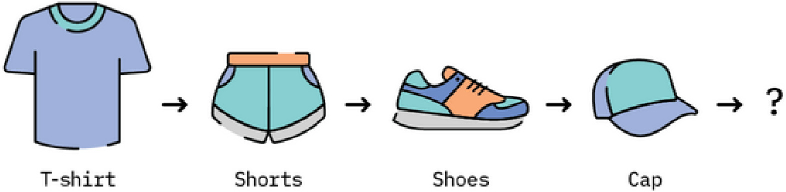
Table 1: Ranking accuracy (with standard errors of about 0.002, 0.001, and 0.001 on the *ML-20M*, *Netflix*, and *MSD* data, respectively), following the experimental set-up in [13].

(a) *ML-20M*	Recall@20	Recall@50	NDCG@100
popularity	0.162	0.235	0.191
EASE^R	0.391	0.521	0.420
EASE$^R \geq 0$	0.373	0.499	0.402
results reproduced from [13]:			
SLIM	0.370	0.495	0.401
WMF	0.360	0.498	0.386
CDAE	0.391	0.523	0.418
MULT-VAE ^{PR}	0.395	0.537	0.426
MULT-DAE	0.387	0.524	0.419
(b) *Netflix*			
popularity	0.116	0.175	0.159
EASE^R	0.362	0.445	0.393
EASE$^R \geq 0$	0.345	0.424	0.373
results reproduced from [13]:			
SLIM	0.347	0.428	0.379
WMF	0.316	0.404	0.351
CDAE	0.343	0.428	0.376
MULT-VAE ^{PR}	0.351	0.444	0.386
MULT-DAE	0.344	0.438	0.380
(c) *MSD*			
popularity	0.043	0.068	0.058
EASE^R	0.333	0.428	0.389
EASE$^R \geq 0$	0.324	0.418	0.379
results reproduced from [13]:			
SLIM	— did not finish in [13] —		
WMF	0.211	0.312	0.257
CDAE	0.188	0.283	0.237
MULT-VAE ^{PR}	0.266	0.364	0.316
MULT-DAE	0.266	0.363	0.313

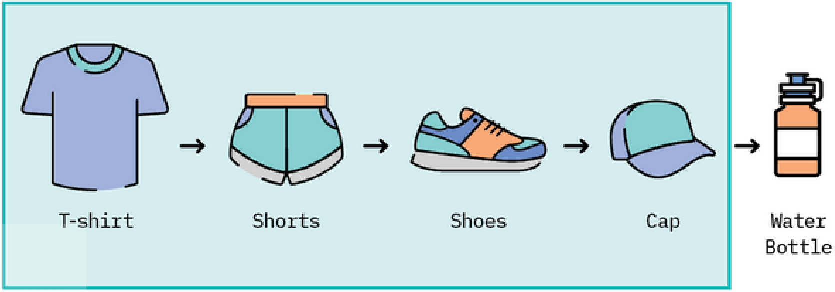
Sequential Recommendation

- Sequential recommendation is the task of predicting the next item that a user will interact
- There is extensive sequential recommendation algorithms
  - Markov chains
  - Recurrent neural networks (RNNs)
  - Long short-term memory (LSTM) networks
  - Embedding-base Neural Networks
- The models should learn patterns in a user's behavior over time

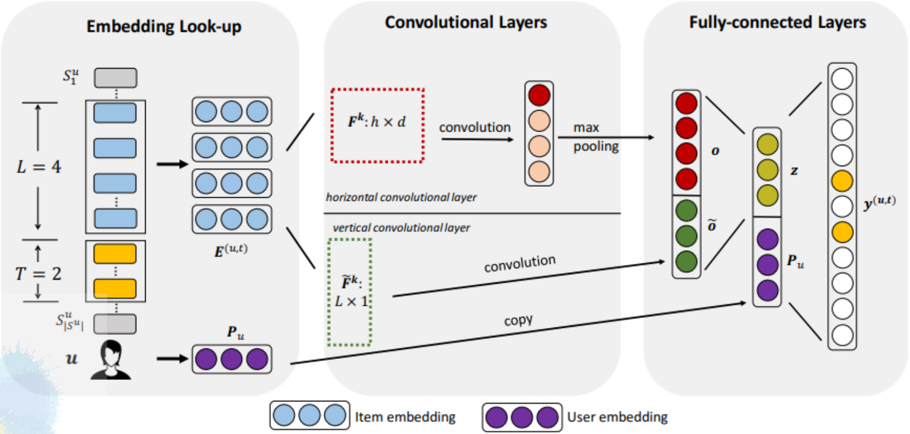
Sequential Recommendation



Sequential Recommendation



Sequential Recommendation



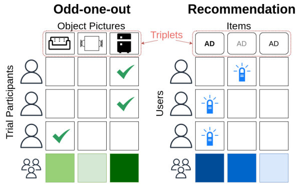
Triplets problem

Triplets problem



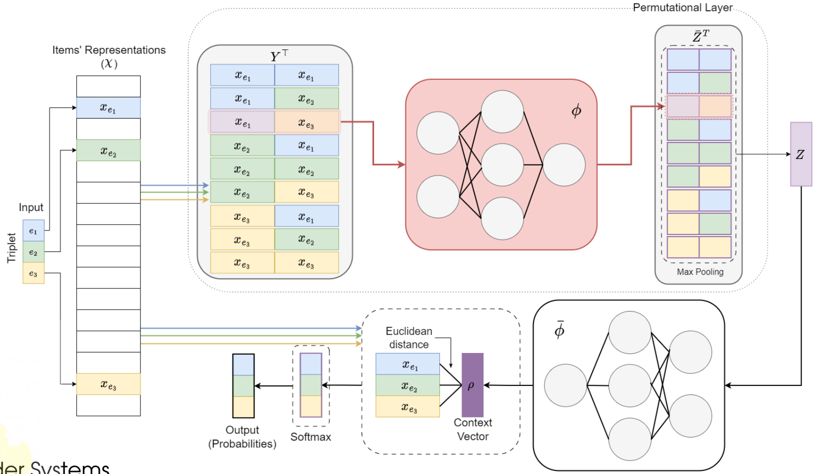



Triplets problem to Recommenders



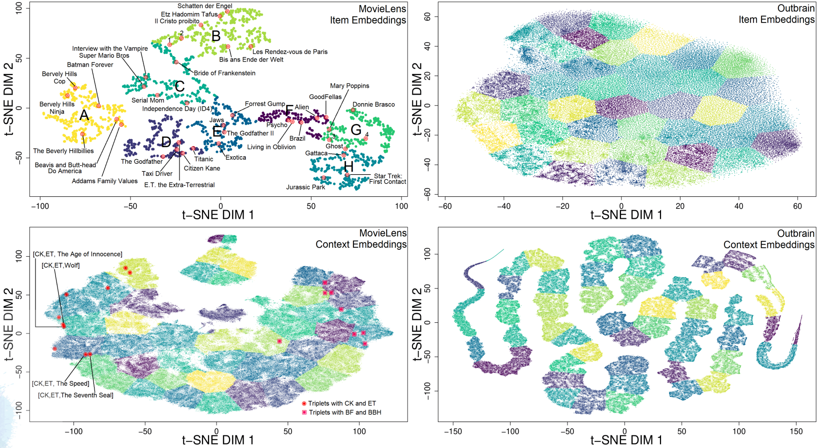
- The items we show to user can influence their decision
- Based on neuroscience
- Sometimes the position we show does not matter significantly
- Context embedding: summarizes the context of the recommendation
- Provide not just accurate recommendation but also interpretability

Care Model

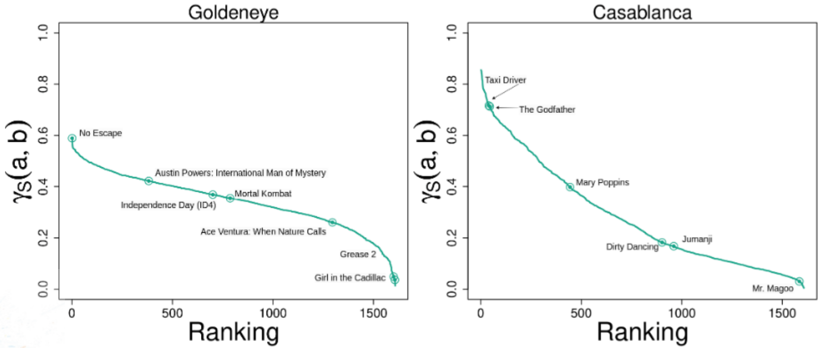


Permutational Layer
Items' Representations
$(\mathcal{X})$
$Y^\top$
ϕ
$\bar{Z}^T$
Max Pooling
Z
Input
Triplet
e_1
e_2
e_3
x_{e_1}
x_{e_2}
x_{e_3}
Euclidean distance
ρ
Context Vector
$\bar{\phi}$
Softmax
Output (Probabilities)

Care Model


MovieLens
Item Embeddings
Schatten der Engel
Etz Hadomim Tafus
Il Cristo proibito
Interview with the Vampire
Super Mario Bros
Batman Forever
Bervely Hills Cop
Bervely Hills Ninja
Les Rendez-vous de Paris
Bis ans Ende der Welt
Bride of Frankenstein
Serial Mom
Independence Day (ID4)
Forrest Gump
Alien
Mary Poppins
GoodFellas
Donnie Brasco
Jaws
Psycho
The Godfather II
Brazil
Living in Oblivion
Ghost
Gattaca
The Beverly Hillbillies
Beavis and Butt-head Do America
Addams Family Values
The Godfather
Taxi Driver
Titanic
Citizen Kane
Exotica
E.T. the Extra-Terrestrial
Jurassic Park
Star Trek: First Contact
Outbrain
Item Embeddings
MovieLens
Context Embeddings
[CK,ET, The Age of Innocence]
[CK,ET,Wolf]
[CK,ET, The Speed]
[CK,ET,The Seventh Seal]
Triplets with CK and ET
Triplets with BF and BBH
Outbrain
Context Embeddings
t–SNE DIM 1
t–SNE DIM 2

Care Model


Goldeneye
γs(a, b)
Ranking
No Escape
Austin Powers: International Man of Mystery
Mortal Kombat
Independence Day (ID4)
Ace Ventura: When Nature Calls
Grease 2
Girl in the Cadillac
Casablanca
Taxi Driver
The Godfather
Mary Poppins
Jumanji
Dirty Dancing
Mr. Magoo
0.0
0.2
0.4
0.6
0.8
1.0
0
500
1000
1500

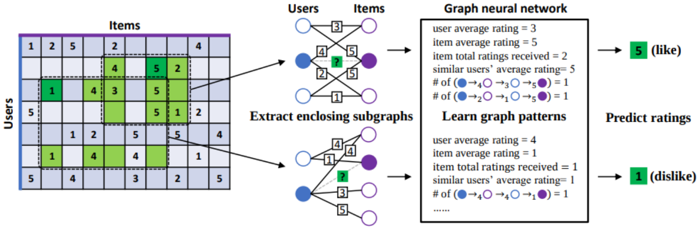
The Time Dimension in Recommendations

- Do you like the same things, morning and evenings?
- For example, the playlist recommendations on Spotify should change based on the time of day and day of the week.
  - Rarely do people have the same mood on Monday morning as they do on Friday evening.
- Taste and preferences change over time, so recommendations should adapt accordingly.
- The envoroment of RS is dynamic

Addressing the Cold-Start Problem

- Recommender systems typically require millions of interactions
- However, new systems often have limited interaction data available
- Attribute-based recommendations can provide valuable information
  - Normally less significant than interactions themselves

Transfer Learning in RS



Tandem (Marriage) Problem

1	C	B	E	A	D
2	A	B	E	C	D
3	D	C	B	A	E
4	A	C	D	B	E
5	A	B	D	E	C

A	3	5	2	1	4
B	5	2	1	4	3
C	4	3	5	1	2
D	1	2	3	4	5
E	2	3	4	1	5

Greedy Czech Algorithm

Rogue Combination

Rogue Combination

1	C	B	E	A	D
2	A	B	E	C	D
3	D	C	B	A	E
4	A	C	D	B	E
5	A	B	D	E	C

A	3	5	2	1	4
B	5	2	1	4	3
C	4	3	5	1	2
D	1	2	3	4	5
E	2	3	4	1	5

1		2		3		4		5	

Gale & Shapley Algorithm

Gale & Shapley Algorithm

Gale & Shapley Algorithm

1	C	B	E	A	D
2	A	B	E	C	D
3	D	C	B	A	E
4	A	C	D	B	E
5	A	B	D	E	C

A	3	5	2	1	4
B	5	2	1	4	3
C	4	3	5	1	2
D	1	2	3	4	5
E	2	3	4	1	5

1	C	2	A	3	D	4		5	

Gale & Shapley Algorithm

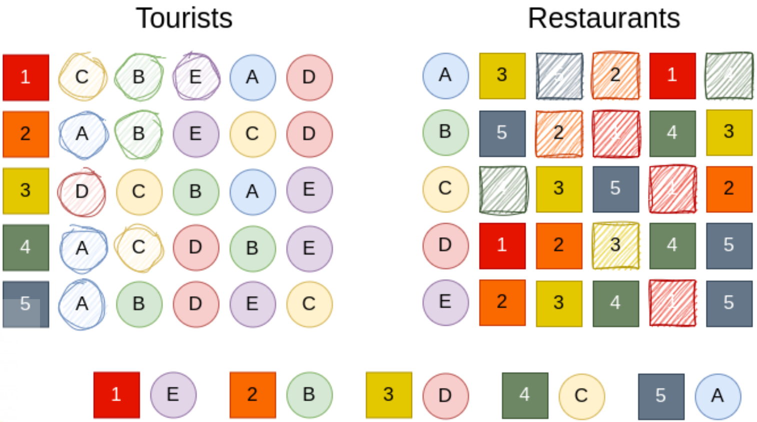
Gale & Shapley Algorithm

Gale & Shapley Algorithm

Gale & Shapley Algorithm...

What about Recommender Systems?

What about Recommender Systems?



Tourists and Restaurants

- Restaurants has working times
- Tourists have visiting period
- Restaurants has limited number of tables
- Tourists has diet restrictions
- A new restaurant can be added to the item list
- A tourist might not have interactions for the visiting city

Recommendation are often a challenging task in a Tandem Problem scenario.

Multi-agents in RS

- Opportunity for research development
- More applied previous works with almost no theory
- Not restrict to the Tandem problem
- Not restrict to the user and items as agents

Modern matters in RS

- Fairness
  - Recommender systems have the potential to perpetuate or even amplify bias
  - Unequal treatment of different groups of users
- Filter Bubbles
  - Common problem on RSs that rely heavily on personalization
  - Recommendations that align with a user's pre-existing preferences
  - Negative consequences for both individual users and society

Modern matters in RS

- Challenging to evaluate
  - Lack of ground truth
  - Changes over the time
  - Diversity of user preferences
- Scalability
  - When terabytes of memory is not enough
  - Can result in increased computational costs and reduced performance

Modern matters in RS

- Privacy concerns
  - Recommender systems often rely on user data to provide accurate recommendations
  - Legislation (GDPR)
  - Lack of interpretability
- Dynamic preference
  - User preferences and item characteristics can be highly dynamic
  - Item availability
  - Difficult to provide accurate and up-to-date recommendations

Obrigado :) - Faculty of Information Technology