Lecture 4 - Modern Recommender Systems

Advanced Machine Learning

Miroslav Čepek, Zdeněk Buk, **Rodrigo da Silva Alves**,
Vojtěch Rybář, Petr Šimánek

FIT CTU

16. 3. 2023

RECAP: Explicit feedback

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

RECAP: 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

RECAP: Matrix Factoriazation

- From a rating matrix R:

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

- 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, such as

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

Example

- The resulting approximation is

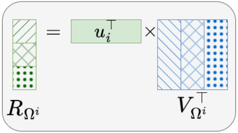
$$\mathbf{UV}^\top = \begin{pmatrix} 0.3 & 0.7 \\ 0.3 & 0.5 \\ 0.2 & 0.4 \\ 0.2 & 0.1 \end{pmatrix} \begin{pmatrix} 1 & 10 & 11 & 10 & 4 & 20 \\ 1 & -1 & -2 & -1 & 1 & -4 \end{pmatrix} =$$

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

where the red numbers are the desired predictions!

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

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 last class 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:

$$\min_{U,V} \sum_{i,j} C_{ij}(P_{ij} - u_i^\top v_j)^2 + \lambda||u_i||^2 + \lambda||v_j||^2$$

- 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

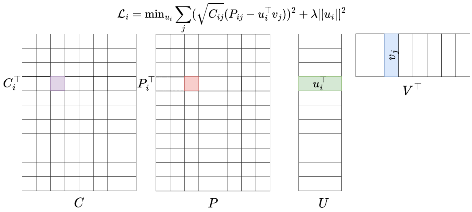
$$\mathcal{L}_i = \min_{u_i} \sum_j C_{ij}(P_{ij} - u_i^\top v_j)^2 + \lambda ||u_i||^2$$

That is the same of:

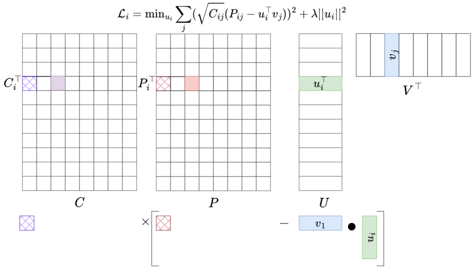
$$\mathcal{L}_i = \min_{u_i} \sum_j (\sqrt{C_{ij}}(P_{ij} - u_i^\top v_j))^2 + \lambda ||u_i||^2$$

Exercise: Find the closed form.

Alternating least squares (ALS)

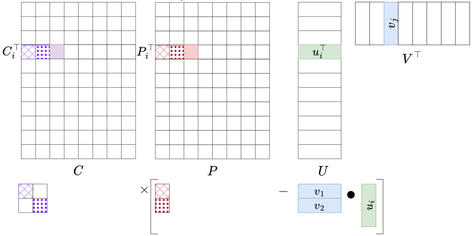


Alternating least squares (ALS)



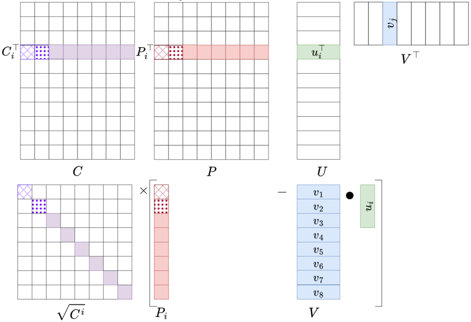
Alternating least squares (ALS)

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



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

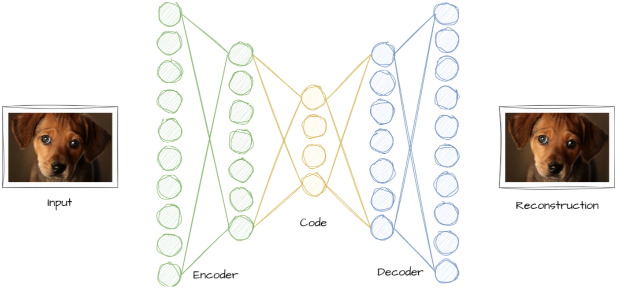
- An autoencoder is a type of feed-forward neural network
- It is designed to reconstruct its input x_i ad output x_i
- To prevent trivial solutions, the network includes a bottleneck (or code) layer
    - Significantly smaller dimension than the input

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:

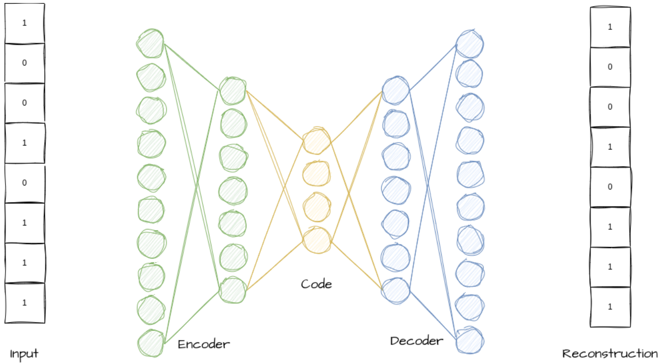
$$\min_{\mathcal{E},\mathcal{D}} \sum_i ||x_i - \mathcal{D}(\mathcal{E}(x_i))||$$

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

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

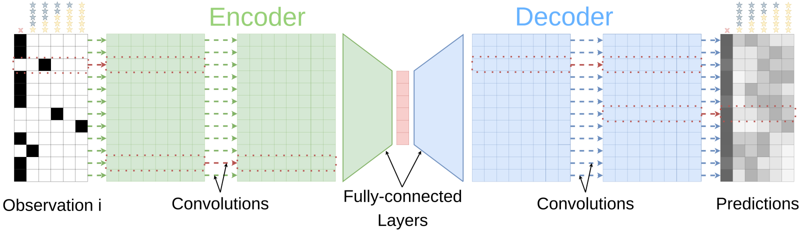
- Why do we need the constraint $\text{diag}(B) = 0$?
- EASE has closed form solution! See here
- Is this a good method?

EASE Results

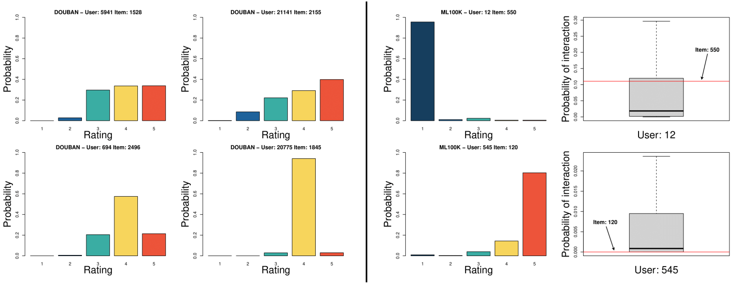
Table 1: Ranking accuracy (with standard errors of about 0.002, 0.001, and 0.001 on the *ML-20M*, *Netflix*, and *MSD* data, respectively), following the experimental set-up in [13].

(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

Mixing Implicit and Explicit



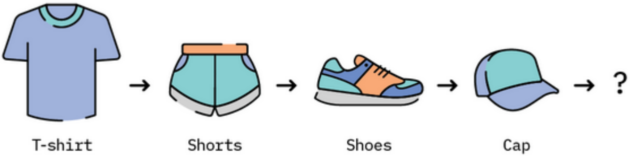
Mixing Implicit and Explicit



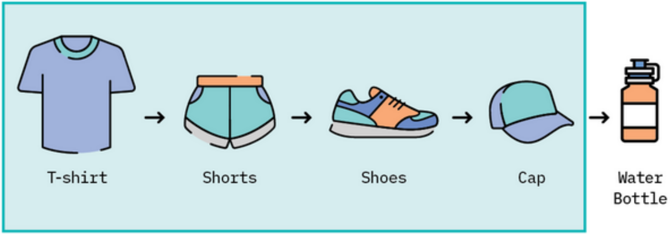
Sequential Recommendation

- 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

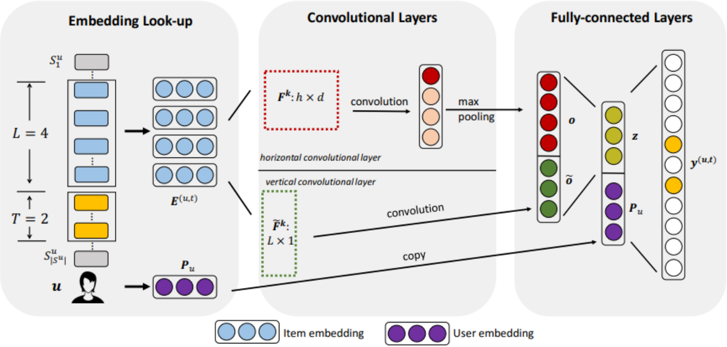
Sequential Recommendation



Sequential Recommendation



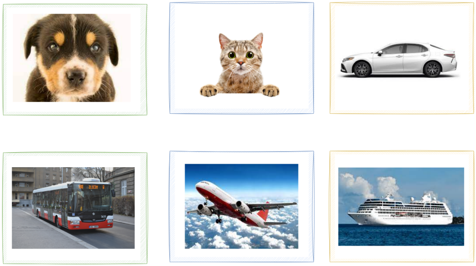
Sequential Recommendation



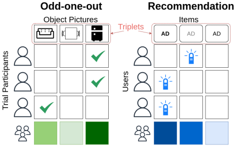
Triplets problem

Triplets problem

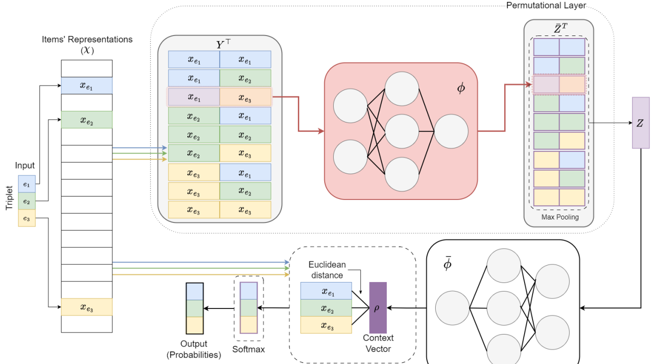


Triplets problem to Recommenders



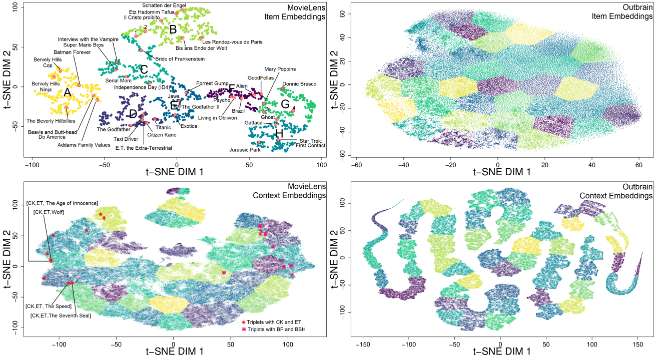
- The items we show to user can influence their decision
- Based on neuroscience
- Sometimes the position we show does not matter significantly
- Context embedding: summarizes the context of the recommendation
- Provide not just accurate recommendation but also interpretability

Care Model

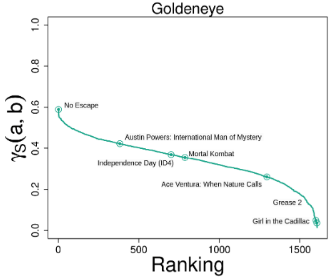


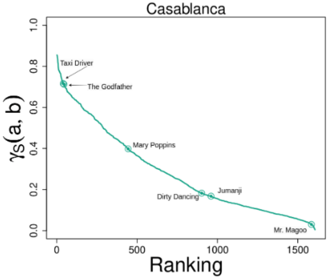
Permutational Layer
Items' Representations
(X)
$Y^\top$
ϕ
$\bar{Z}^T$
Max Pooling
Z
Input
Triplet
e_1
e_2
e_3
x_{e_1}
x_{e_2}
x_{e_3}
Euclidean distance
ρ
Context Vector
$\bar{\phi}$
Output (Probabilities)
Softmax

Care Model


MovieLens
Item Embeddings
Outbrain
Item Embeddings
MovieLens
Context Embeddings
Outbrain
Context Embeddings
t–SNE DIM 1
t–SNE DIM 2
Triplets with CK and ET
Triplets with BF and BBH

Care Model

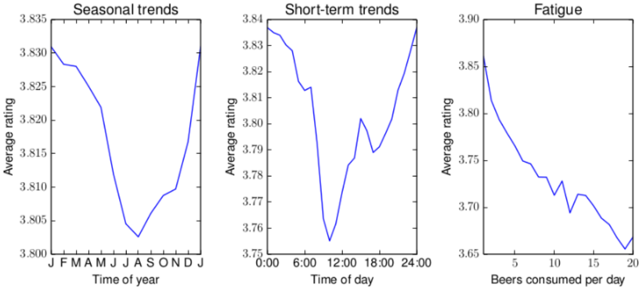

Goldeneye
γs(a, b)
Ranking
No Escape
Austin Powers: International Man of Mystery
Independence Day (ID4)
Mortal Kombat
Ace Ventura: When Nature Calls
Grease 2
Girl in the Cadillac
0
500
1000
1500
0.0
0.2
0.4
0.6
0.8
1.0


Casablanca
γs(a, b)
Ranking
Taxi Driver
The Godfather
Mary Poppins
Dirty Dancing
Jumanji
Mr. Magoo
0
500
1000
1500
0.0
0.2
0.4
0.6
0.8
1.0

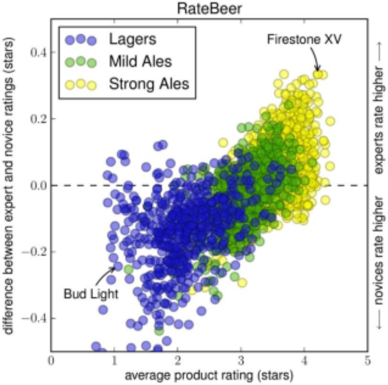
The Time Dimension in Recommendations

- 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

Pivo Recommendation



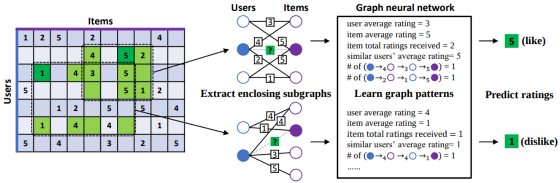
Pivo Recommendation



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



Social 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

Social 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

Social 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