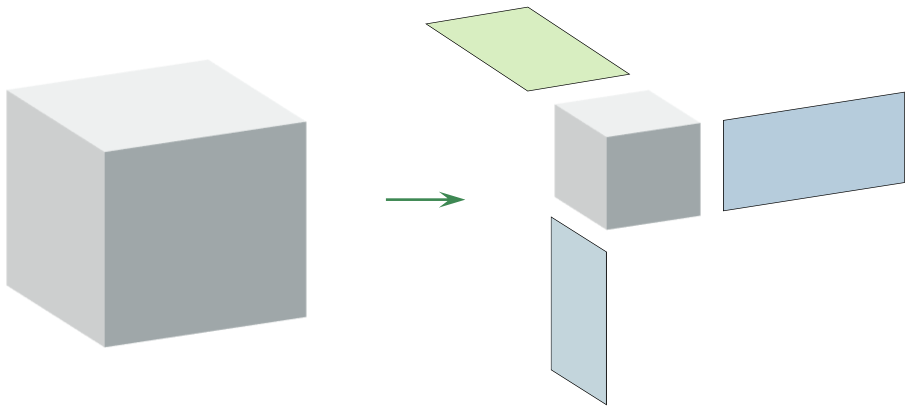
(Some) Tensor Methods in Recommender Systems



15.03.2022

For the practical part:

Python environment:

- Numpy, Scipy, Numba, Pandas (preferably via `conda`)
- Polara (https://github.com/evfro/polara#installation):

```
pip install git+https://github.com/evfro/polara.git@develop#egg=polara
```

Running the code:

- Jupyterlab/Jupyter Notebook
- or VS Code

What is a recommender system?

Examples:

- Amazon
- Netflix
- Pandora
- Spotify
- Social platforms
- etc.

Many different areas: e-commerce, news, tourism, entertainment, education...

Goal: predict user preferences given some prior information on user behavior.

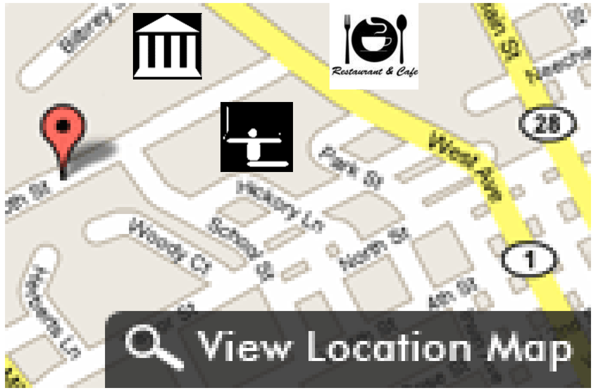
In a more general sense

Recommender Systems aim to recover partially observed relations between two or more entities.

Social Networks:
user ↔ user

Sequential data:
item → next item
(order matters)

Ternary relations:
user → action → location



General workflow

Goal: predict user preferences based on prior user feedback and collective user behavior.

collect data

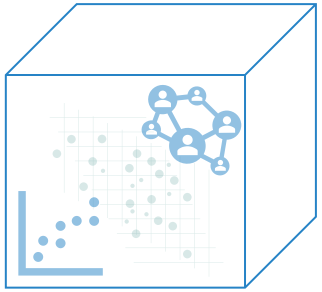
user-movie matrix A of size $M \times N$

a_{ij} is a rating of i^{th} user for j^{th} movie

? - missing (unknown) values

build model

f_U: User × Item → Relevance



generate recommendations

unknown user

Conventional techniques

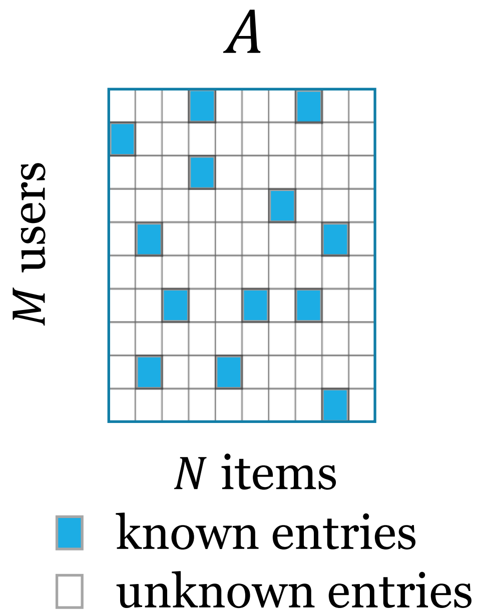
A general view on latent factors models

- **Task**: find utility (relevance) function f_R :

$$f_R\colon \text{Users} \times \text{Items} \rightarrow \text{Relevance score}$$

- As optimization problem with some *loss function* $\mathcal{L}$:

$$\mathcal{L}(A, R) \rightarrow \min$$



Components of the model:

- Utility function to generate R
- Optimization objective defined by $\mathcal{L}$
- Optimization method (algorithm)

Intuition behind MF

Assumption: observed interactions can be explained via

- a *small* number of common patterns in human behavior
- + individual variations (including random and unobservable factors)

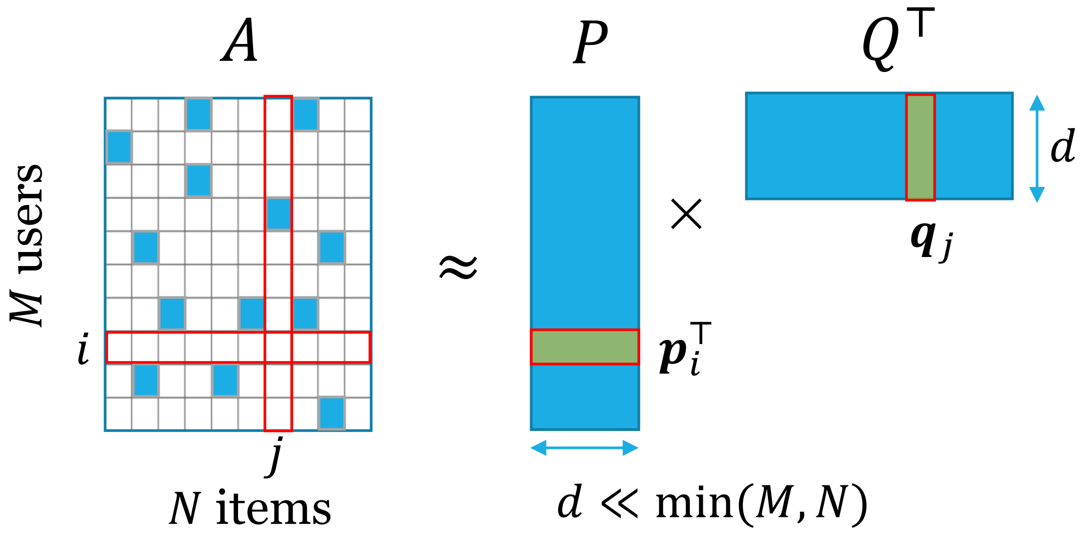
$$A_{full} = R + E, \qquad R = PQ^\top$$

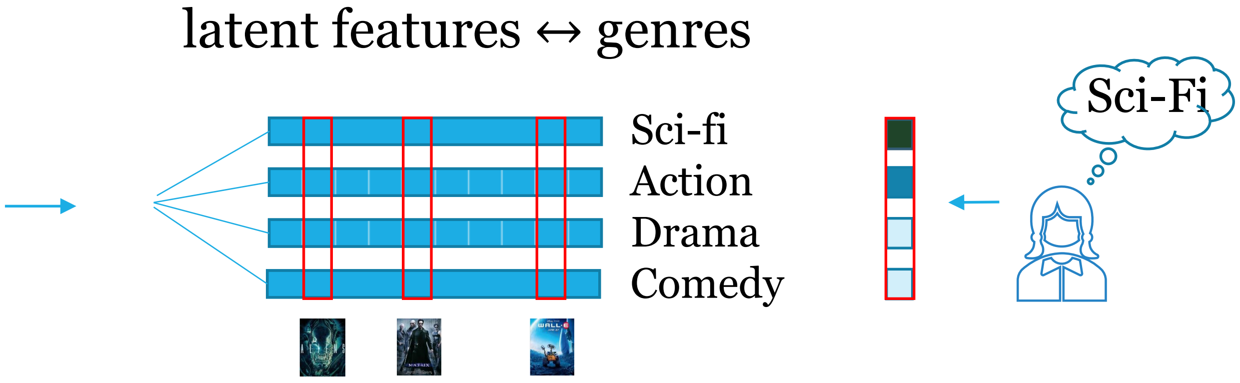
Low rank representation

?	3	5	5
4	?	5	5

4	3	?		
?	3	5		
4	?	5		
			4	5
			?	5

Simplistic view on latent features





rows of P – user embeddings
rows of Q – item embeddings

Predicted utility of item j for user i:

$$r_{ij} \approx \boldsymbol{p}_i^\top \boldsymbol{q}_j = \sum_{k=1}^{d} p_{ik} q_{jk}$$

$\boldsymbol{p}_i$ – latent factors vector for user i
$\boldsymbol{q}_j$ – latent factors vector for item j

Top-n recommendations task

Expected output: a ranked list of n most relevant items.

$$\operatorname{toprec}(i, n) := \arg\max_{j}^{n} r_{ij}$$

r_{ij} - is the predicted relevance score between user i and item j

Example:

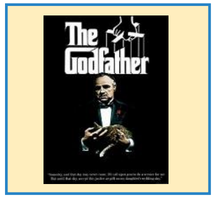
r_{ij}	0.73	0.41	0.95
item			

top-2 recommendations

leftsorted list

Evaluation or recommendations

User

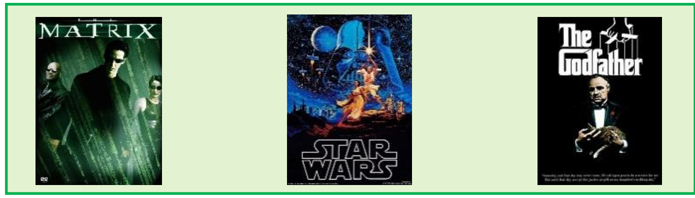


holdout

known user preferences

recommendations

Algorithm:



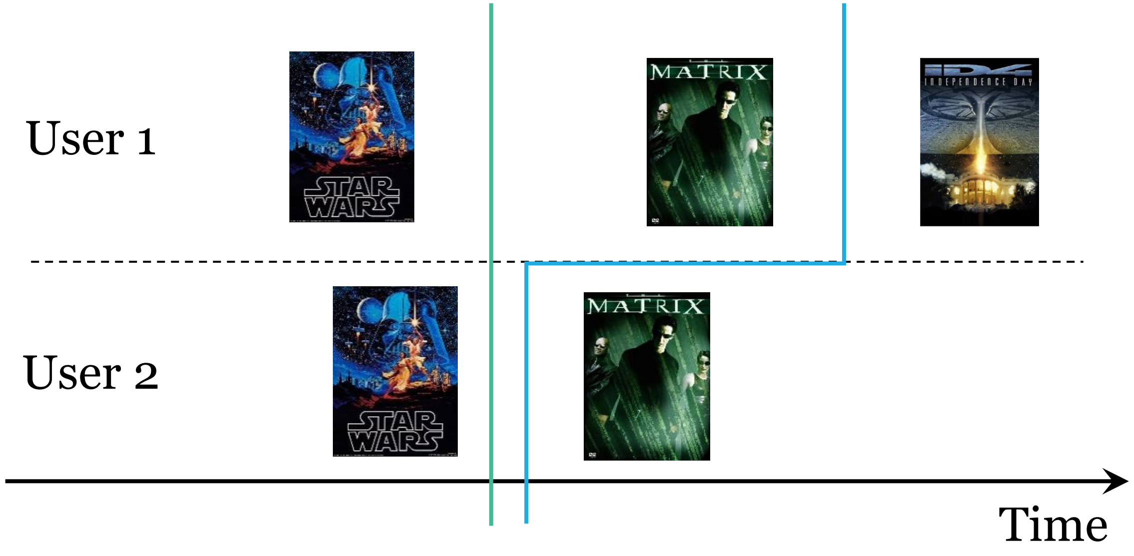
$$\mathrm{HR} = \frac{1}{\#(\text{test users})} \sum_{\substack{\text{test}\\\text{users}}} \text{hit} \qquad \text{hit} = \begin{cases} 1 & \text{if holdout item is in recommended items,} \\ 0 & \text{otherwise.} \end{cases}$$

$$\mathrm{MRR} = \frac{1}{\#(\text{test users})} \sum_{\substack{\text{test}\\\text{users}}} \frac{1}{\text{hit rank}}$$

hit rank = position of the item in the recommendations list

Typically computed: metric@n, where n = #recommended items, e.g. Recall@n, MRR@n, etc.

Temporal splits



Data split strategy	Definition of training and test instances	Local timeline	Global timeline	Data leakage
Random-split-by-ratio	Randomly sample a percentage of user-item interactions as test instances; the remaining are training instances.	No	No	Yes
Random-split-by-user	Randomly sample a percentage of users, and take all their interactions as test instances; the remaining instances from other users are training instances.	No	No	Yes
Leave-one-out-split	Take each user's last interaction as a test instance; all remaining interactions are training instances.	Yes	No	Yes
Split-by-timepoint	All interactions after a time point are test instances; interactions before this time point are training instances.	No	Yes	No

Table source: Ji, Yitong, Aixin Sun, Jie Zhang, and Chenliang Li. "A critical study on data leakage in recommender system offline evaluation." *arXiv preprint arXiv:2010.11060* (2020).

Singular Value Decomposition

Quick reminder:

$$A = U\Sigma V^\top$$

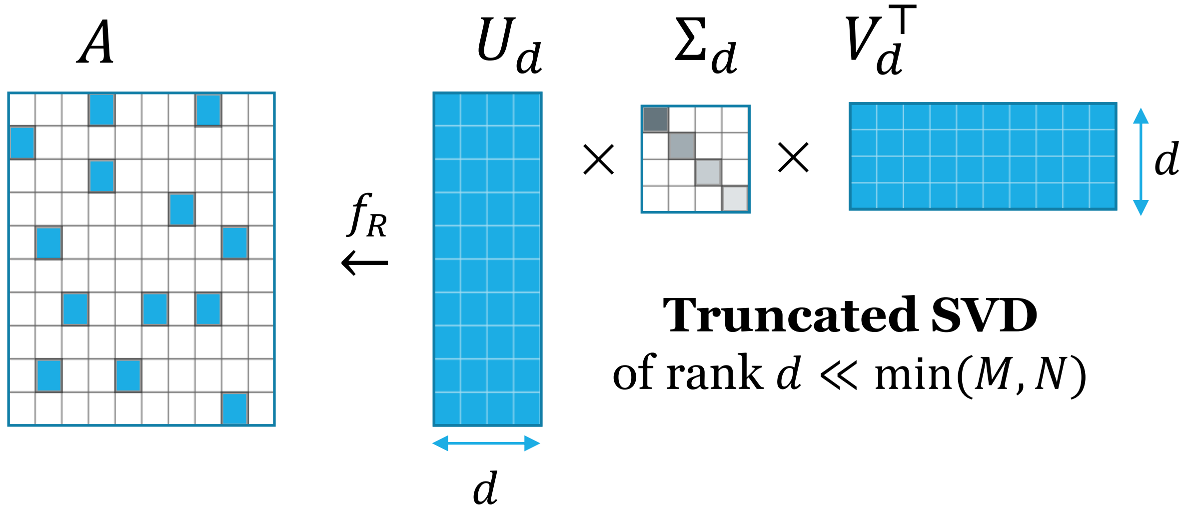
$$U \in \mathbb{R}^{M\times M}, \qquad V \in \mathbb{R}^{N\times N}$$

$$U^\top U = I_M, \qquad V^\top V = I_N$$

$\Sigma \in \mathbb{R}^{M\times N}$ - diagonal, with $[\Sigma]_{kk} = \sigma_k$:

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{\min(M,N)} \geq 0$$

$$\sigma_k(A) = \sqrt{\lambda_k(A^\top A)} = \sqrt{\lambda_k(AA^\top)}$$



Truncated SVD
of rank $d \ll \min(M, N)$

Low-rank approximation task:

$$\|A - R\|_F^2 \to \min, \ \text{s.t.}\ \operatorname{rank}(R) = d$$

$$R = U_d \Sigma_d V_d^\top$$

Undefined for incomplete matrix!

PureSVD model for CF

- Let's impute zeros in place of unknowns!

Works well in practice*!

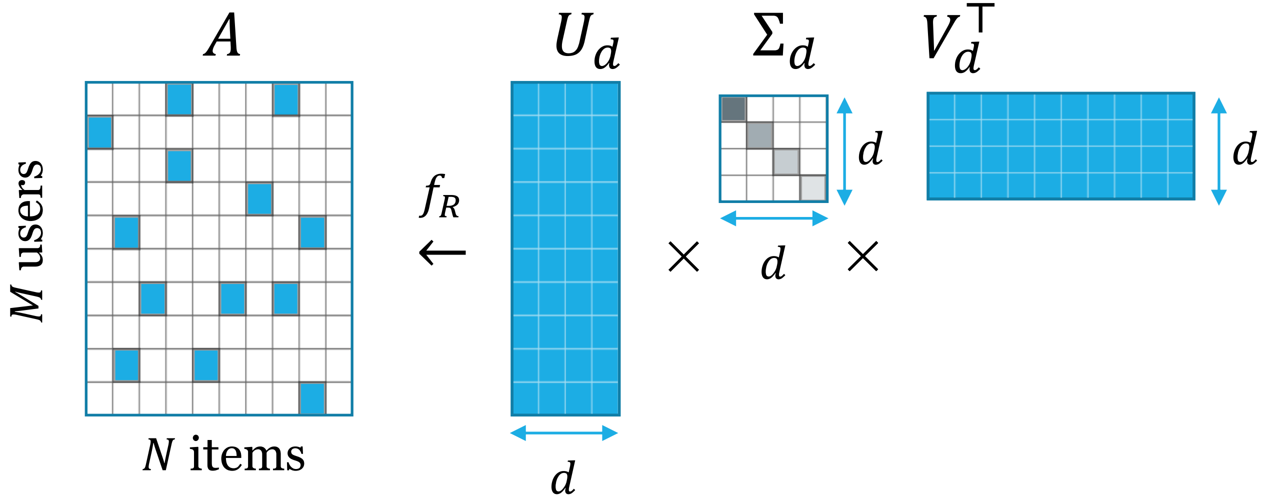
$$A_0 = U\Sigma V^\top, \qquad [A_0]_{ij} = \begin{cases} a_{ij}, & \text{if known} \\ 0, & \text{otherwise} \end{cases}$$

- Relevance score prediction:

$$R = U_d \Sigma_d V_d^\top = A_0 V_d V_d^\top$$

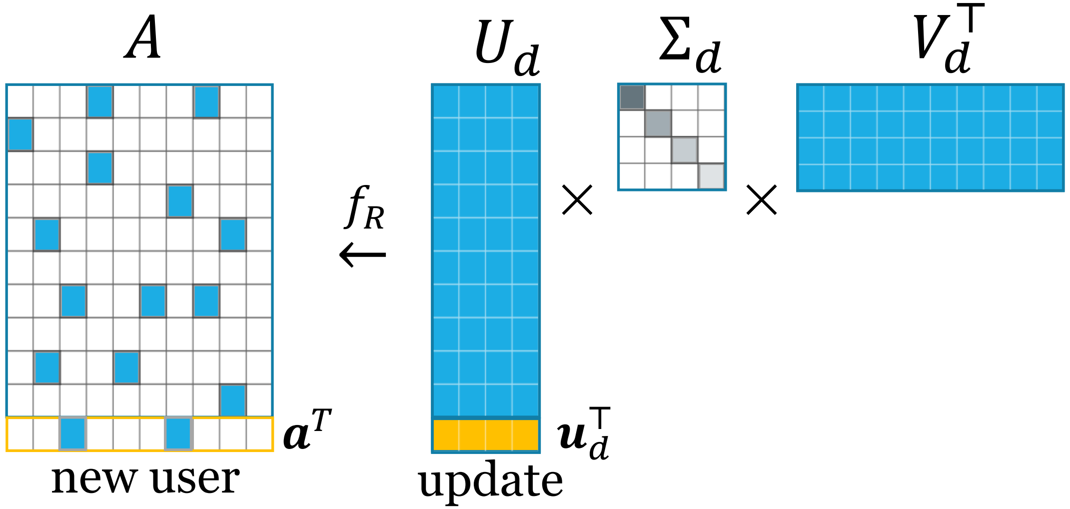
- Scores prediction for a user

$$\boldsymbol{r} = V_d V_d^\top \boldsymbol{a}_0$$



* Cremonesi, Paolo, Yehuda Koren, and Roberto Turrin. "Performance of recommender algorithms on top-n recommendation tasks." In *Proceedings of the fourth ACM conference on Recommender systems*, pp. 39-46. 2010.

PureSVD – recommending online



folding-in technique[*]

[*]G. Furnas, S. Deerwester, and S. Dumais, "Information Retrieval Using a Singular Value Decomposition Model of Latent Semantic Structure," Proceedings of ACM SIGIR Conference, 1988

Finding a warm-start user representation:

$$\left\|\boldsymbol{a}_0^\top - \boldsymbol{u}^\top \Sigma V^\top\right\|_2^2 \to \min$$

new user embedding

$$\boldsymbol{u}^\top = \boldsymbol{a}_0^\top V \Sigma^{-1}$$

Prediction:

$$\boldsymbol{r}^\top = \boldsymbol{u}^\top \Sigma_d V_d^\top = \boldsymbol{a}_0^\top V_d V_d^\top$$

$$\boldsymbol{r} = V_d V_d^\top \boldsymbol{a}_0$$

- convenient for evaluation
- complexity $\sim O(Nd)$
- enables real-time recommendations

Some practical observations

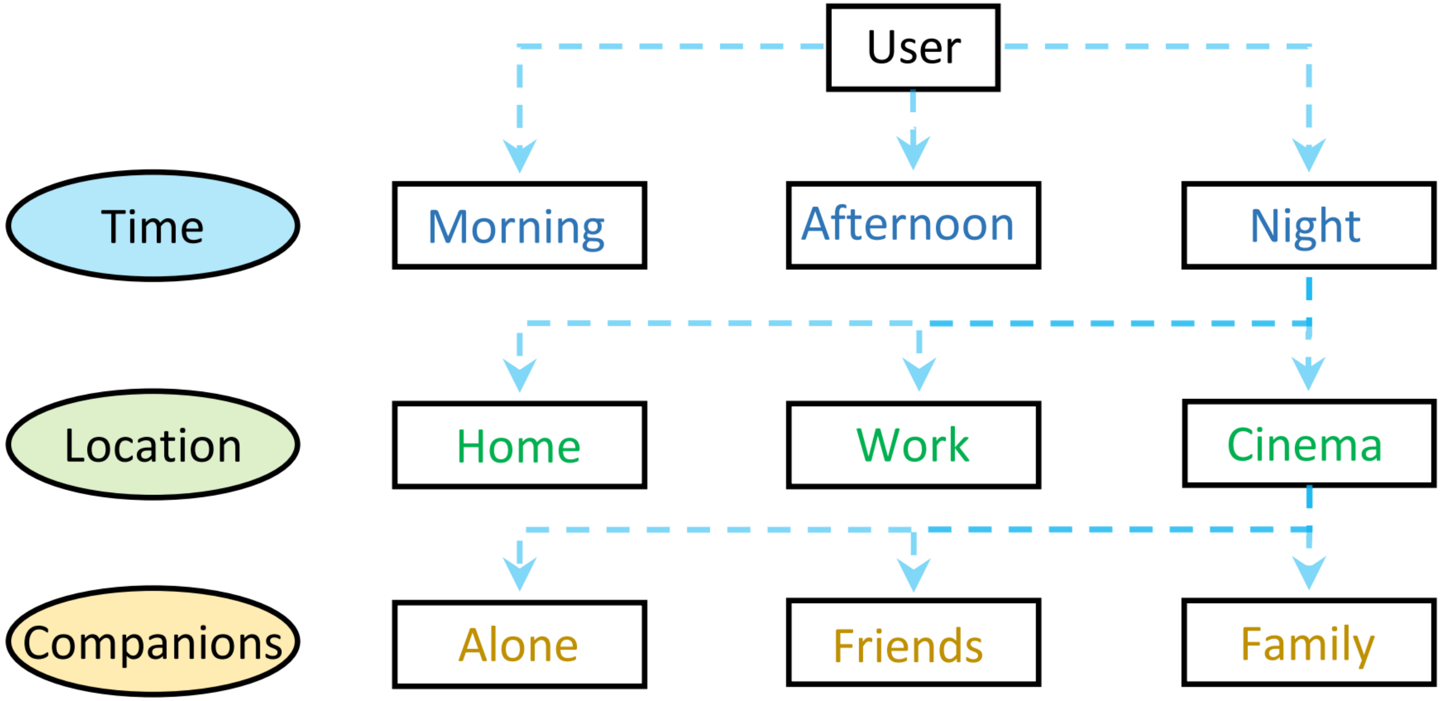
- what SVD for CF is not :
  - pure matrix completion
  - pure dimensionality reduction

- common PCA-like preprocessing may spoil data representation

- rating prediction doesn’t make sense
  - recommendations can still be good!
  - we can treat rating values more flexibly

Higher-order techniques

Context-awareness in RecSys



Also: folksonomies, cross-domain RS, temporal models, etc.

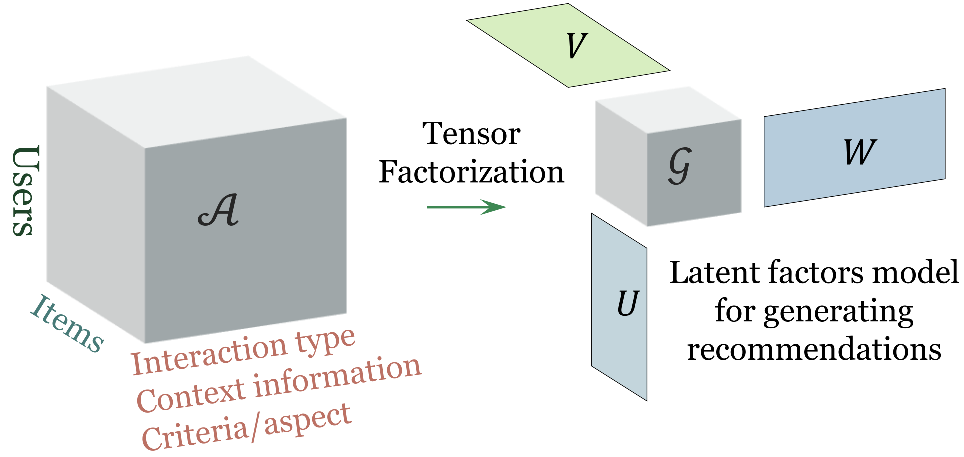
Latent factors models with TF

- **Task**: find utility (relevance) function f_R :

$$f_R\colon \text{Users} \times \text{Items} \times \text{Context} \rightarrow \text{Relevance score}$$

- As optimization problem with some *loss function* $\mathcal{L}$:

$$\mathcal{L}(\mathcal{A}, \mathcal{R}) \rightarrow \min$$



Contextual top-n recommendations

Possible scenarios:

- recommend the best items within a selected context
  - e.g., best restaurant based on location

$$\text{toprec}(u, c, n) := \arg\max_{i}^{n} r_{uic}$$

- recommend the best context for a target item
  - e.g., find best distribution channel

$$\text{toprec}(u, i, n) := \arg\max_{c}^{n} r_{uic}$$

Tensor decompositions in recsys

Properties	TD	CP
# learnable parameters	$mnd + d^m$	mnd
Uniqueness	No	Yes (under mild conditions)
Stability	stable	ill-posed*

*V. De Silva and L.-H. Lim. Tensor rank and the ill-posedness of the best low-rank approximation problem.

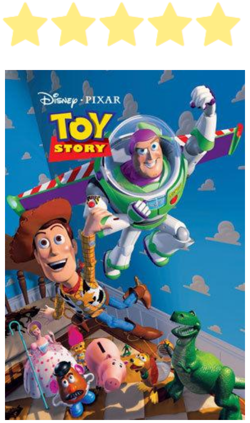
In recommender systems**:

Method	Year	Type	Algorithm
TagTR	2008	TD	HOSVD
Multiverse	2010	TD	SGD
PITF	2010	CP	SGD
TFMAP	2012	CP	SGD
GFF	2015	CP	ALS
Taper	2016	CP	SGD
CoFFee	2016	TD	HOOI

**Read more in recsys: "*Tensor methods and recommender systems*", Evgeny Frolov and Ivan Oseledets. *WIREs Data Mining Knowledge Discovery* 2017, vol. 7, issue 3.

Modeling user ratings

Subjective nature of user feedback

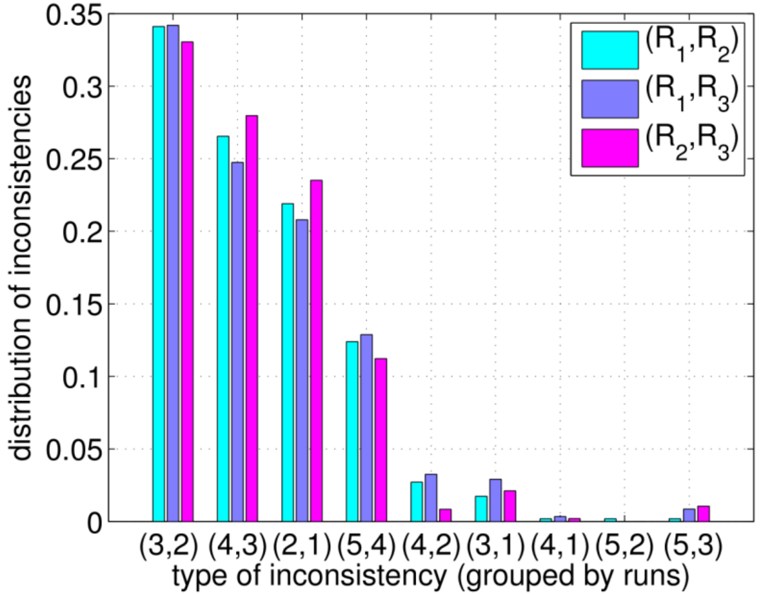


2.5x better?

From neoclassical economics:
utility is an **ordinal concept**.

Traditional recommender models treat ratings as **cardinal numbers**.

Ratings scale consist of **unequal intervals***:

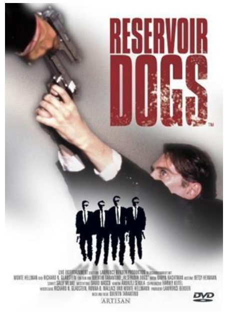


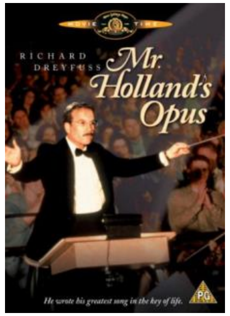
*"I like it... I like it not: Evaluating User Ratings Noise in Recommender Systems" by Xavier Amatriain, Josep M. Pujol, and Nuria Oliver

Negative feedback problem

What is likely to be recommended in this case?

User feedback is negative!
Probably he or she doesn't
like criminal movies.





Explanation

$\xrightarrow{f_U}$

recommendations are insensitive to negative feedback

predicted scores (*folding-in*):

$$\boldsymbol{r} = VV^\top \boldsymbol{p}$$

top-n recommendations task:

$$\operatorname{toprec}(\boldsymbol{p}, n) \coloneqq \arg\overset{n}{\max}\, \boldsymbol{r}$$

$$\arg\max VV^\top(0, \dots, 0, \underset{★★☆☆☆}{\mathbf{2}}, 0, \dots, 0)^\top \equiv \arg\max VV^\top(0, \dots, 0, \underset{★★★★★}{\mathbf{5}}, 0, \dots, 0)^\top$$

Rating value doesn't change ranking of the items!

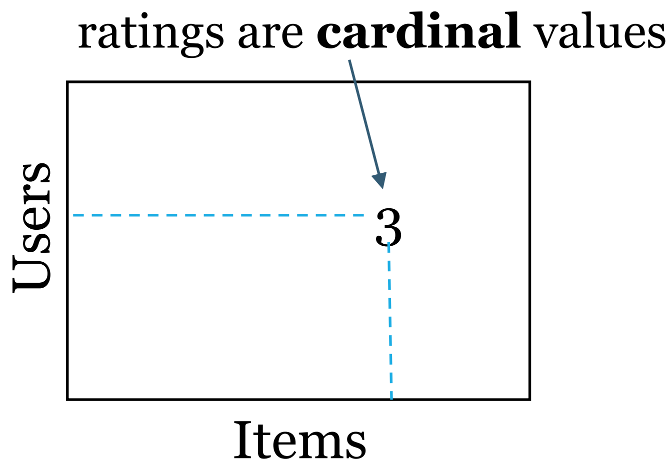
Can we deal with it?

Recommend the least similar items	unlikely to be relevant
	not clear when to switch between similar and dissimilar items
Recommend from similar users	not guaranteed to return items with “opposite” features
	doesn’t scale well
Shift rating values below zero	if predefined – not clear which values should go below zero
	even if learned (a.k.a. bias terms) - equivalent to recommending least similar
Rating elicitation	hard to peak most representative items
	increases barrier to entry (not effortless for user)
	non-personalized user experience

Redefining the utility function

Standard MF model

f_U: User × Item → Rating

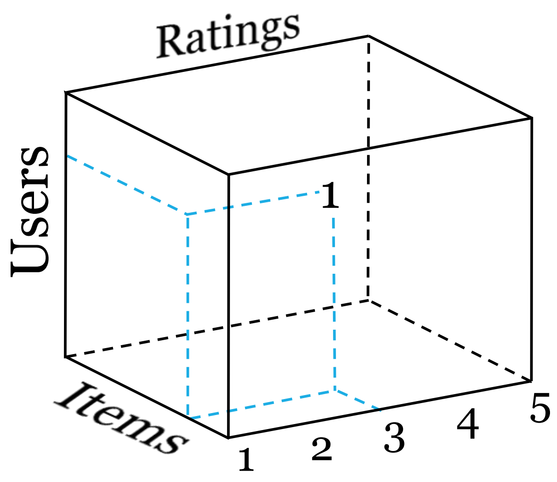


$$||A_0 - R||_F^2 \to \min$$

$$R = U\Sigma V^\top$$

Collaborative Full Feedback model – CoFFee*

f_U: User × Item × Rating → Relevance Score

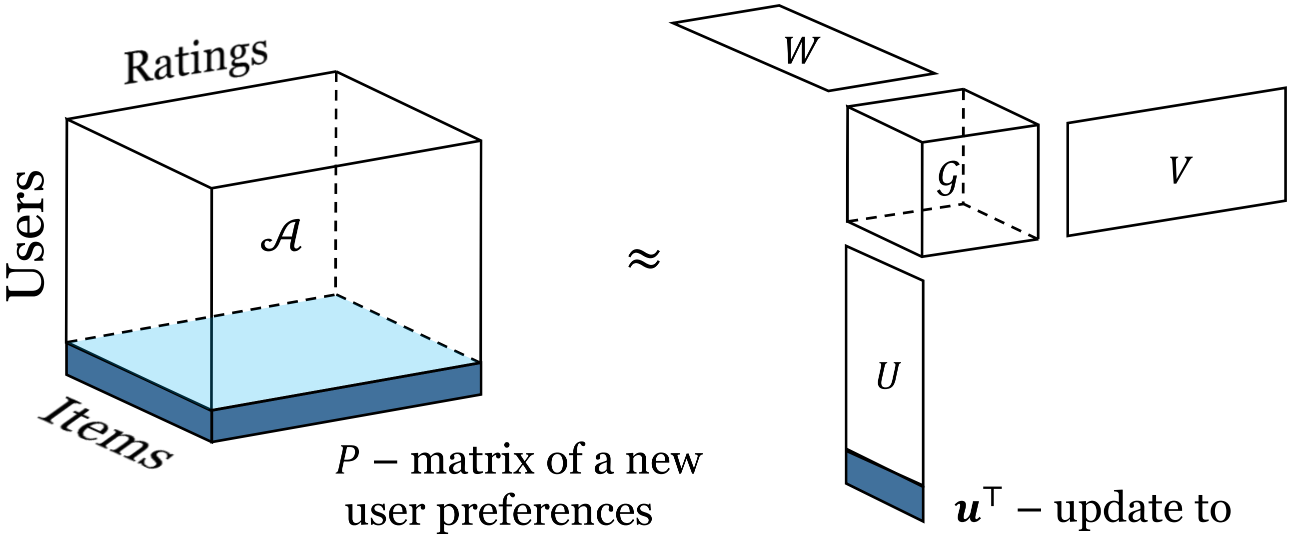


$$||\mathcal{A}_0 - \mathcal{R}||_F^2 \to \min$$

$$\mathcal{R} = \mathcal{G} \times_1 U \times_2 V \times_3 W$$

*[Frolov & Oseledets 2016]

Higher order folding-in



P – matrix of a new user preferences

$\boldsymbol{u}^\top$ – update to the Tucker model

$$R \approx VV^\top P\, WW^\top$$ **predictions matrix**

Compare to SVD:

$$\boldsymbol{r} = VV^\top \boldsymbol{p}$$ **predictions vector**

$$\mathrm{vec}(R)^T = \boldsymbol{u}^T \mathcal{G}_{(1)}\, (W \otimes V)^T$$

How to find $\boldsymbol{u}$?

$$\|P - \mathcal{G} \times_2 V \times_3 \mathrm{W} \times_1 \boldsymbol{u}\|_\mathrm{F}^2 \to \min \ \ or \ \ \left\|\mathrm{vec}(P)^T \to \boldsymbol{u}^T \mathcal{G}_{(1)}\, (W \otimes V)^T\right\|_\mathrm{F}^2 \to \min,$$

$$\mathrm{vec}(R)^T \leftarrow \mathrm{vec}(P)^T [(W \otimes V)^T]^{-1} \mathcal{G}_{(1)}^\dagger \mathcal{G}_{(1)} (W \otimes V)^T$$

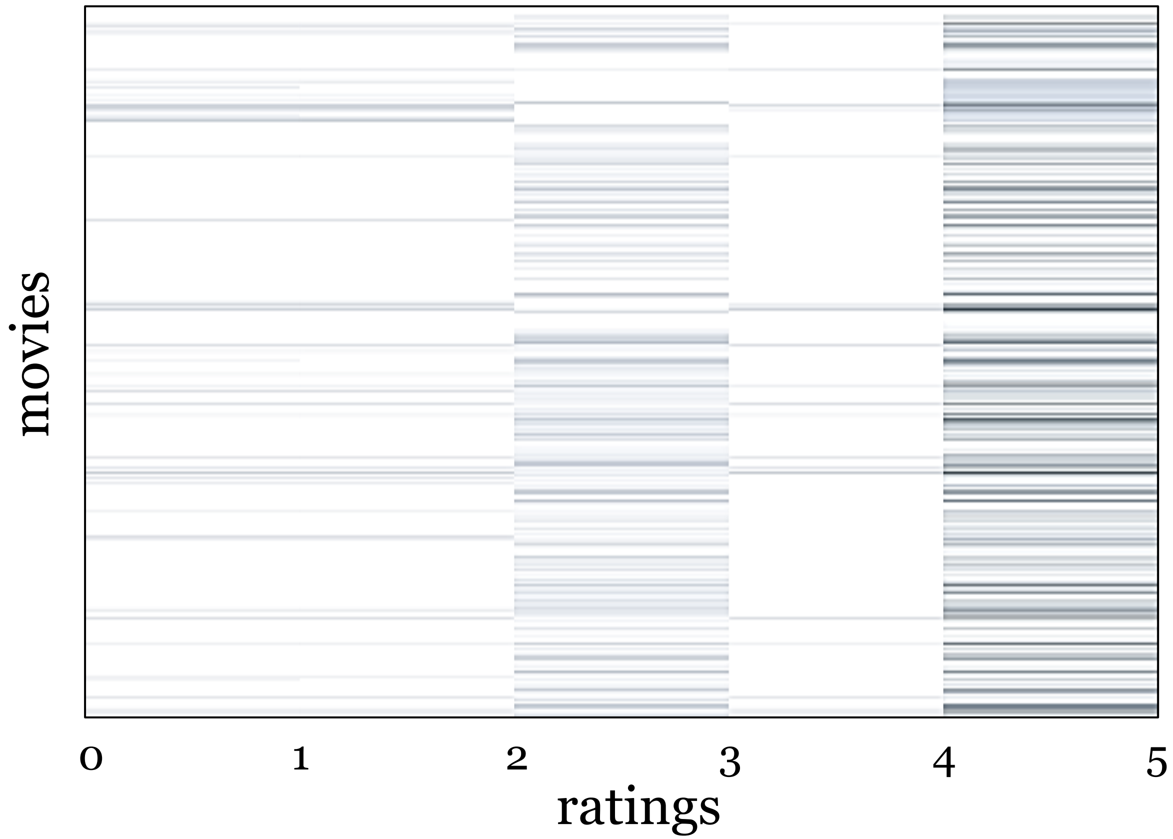
$$\mathrm{vec}(R) \approx \left(WW^T\right) \otimes \left(VV^T\right) \mathrm{vec}(P) = \mathrm{vec}(VV^T P\, WW^T)$$

“Shades” of ratings

$$R = VV^\top P\, WW^\top$$

matrix of known
user preferences

Darker colors correspond to higher relevance score.



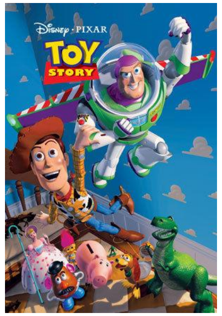
Solves both tasks:

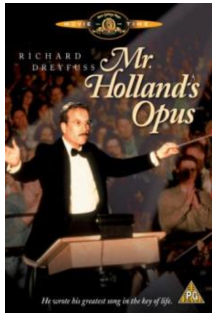
- ranking
- rating prediction

Granular view of user preferences,
concerning **all possible ratings**.

Model is **equally sensitive**
to any kind of feedback.

Recommendations with CoFFee





Uncovers new recommendation modes:

"users who like this also like..." → "users who **dislike** this, do like..."

	Scarface ★★☆☆☆	LOTR: The Two Towers ★★☆☆☆	Star Wars: Episode VII - The Force Awakens ★★★★★
CoFFee	Toy Story Mr. Holland's Opus Independence Day	Net, The Cliffhanger Batman Forever	Dark Knight, The Batman Begins Star Wars: Episode IV - A New Hope
SVD	Reservoir Dogs Goodfellas Godfather: Part II, The	LOTR: The Fellowship of the Ring Shrek LOTR: The Return of the King	Dark Knight, The Inception Iron Man

Multi-criteria ratings

Multi-criteria hotel ratings

TripAdvisor

Multiple emoji, a.k.a reactions

Facebook

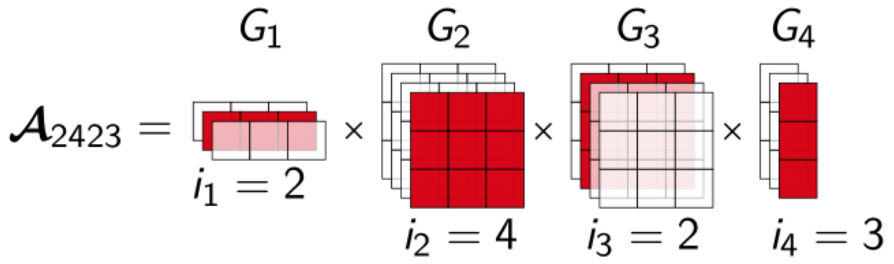
Internal micro-genres ratings*

Netflix

$$f_U\colon \text{User} \times \text{Item} \times \text{Rating}_{c_1} \times \cdots \times \text{Rating}_{c_n} \times \text{Rating} \rightarrow \text{Relevance Score}$$

TD is not efficient for ND tensors with N > 4 → Tensor Train Decomposition*:

$$\mathcal{A}_{i_1 \dots i_d} = \underbrace{G_1[i_1]}_{1\times r}\ \underbrace{G_2[i_2]}_{r\times r} \dots \underbrace{G_d[i_d]}_{r\times 1}$$



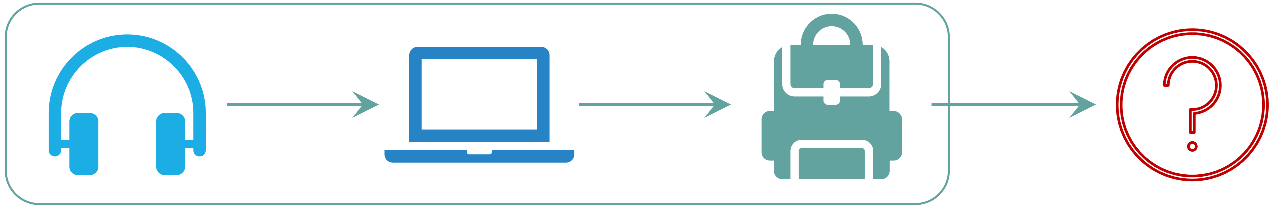
\# learnable parameters : mnd^2

*Ivan Oseledets, "*Tensor-Train Decomposition*", SIAM Journal on Scientific Computing 33, 2295-2317

Sequential modeling

Sequential recommendations

- A user's decision to consume the next item may be influenced by:
  - a few most recent items
  - consumed in a specific sequential order



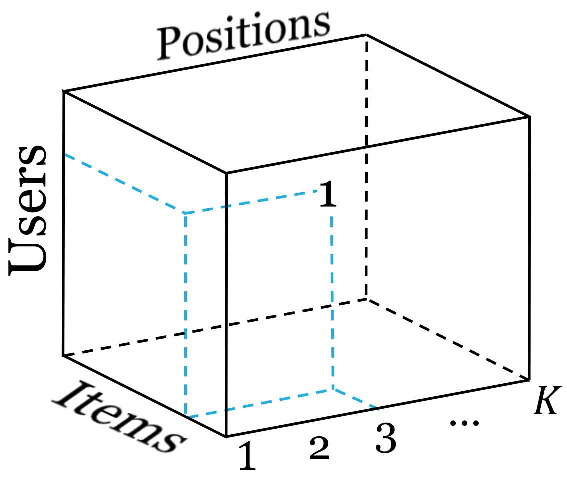
Example:

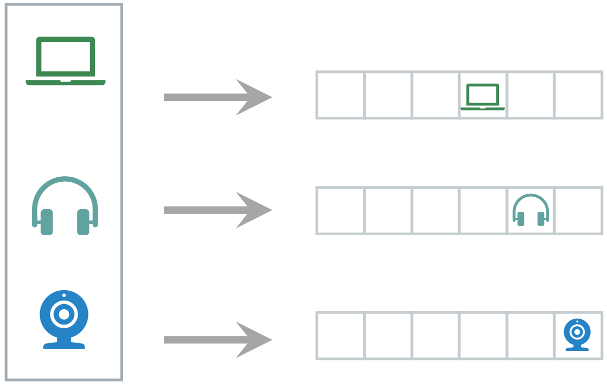
- purchasing a laptop may lead to the purchase of a backpack
- the opposite is unlikely, however
- if prior to the laptop, the user also bought headphones:
  - could we reliably predict a backpack purchase from here?

Tensor Factorization with Positional Information

$$||\mathcal{A}_0 - \mathcal{R}||_F^2 \rightarrow \min$$

$$\boxed{f_U\text{: User} \times \text{Item} \times \text{Position} \rightarrow \text{Relevance Score}}$$

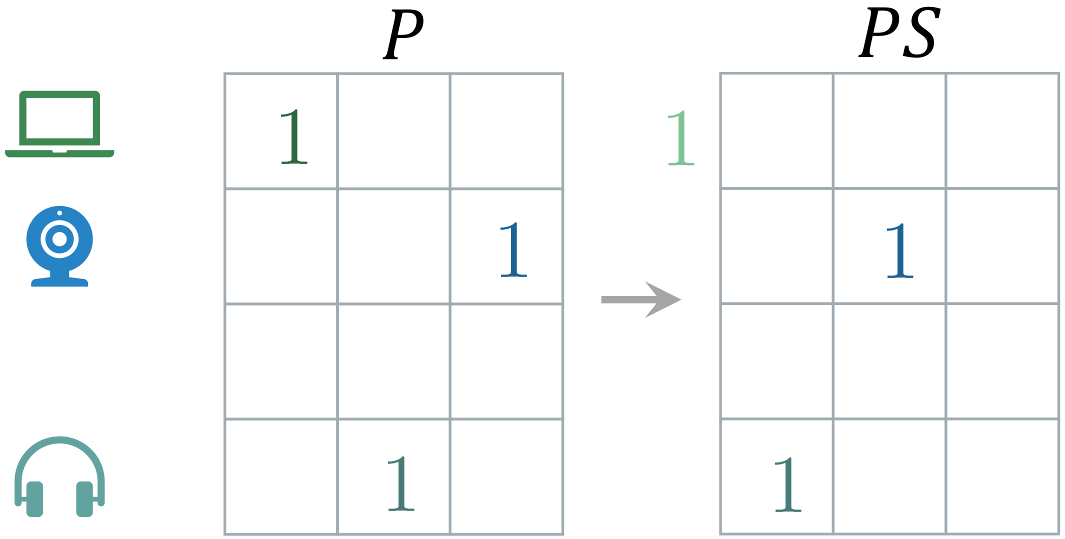


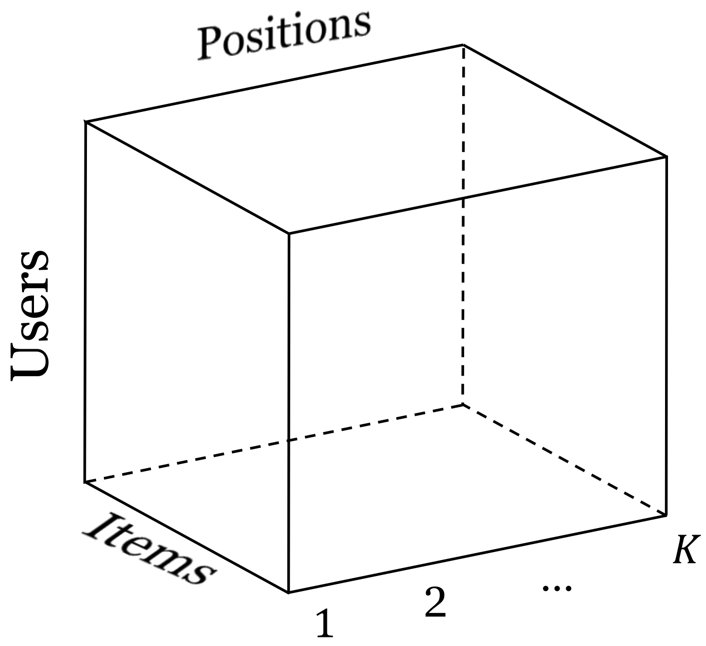


- encode positions as a third categorical entity
- need to handle user sequences of variable length
  - pad with 0
- local vs. global sequential context
  - weighting based on position
- how to generate predictions?

Predicting future interactions with TF

- there's no notion of "future item" in our tensor
- idea: treat the last position as the prediction target

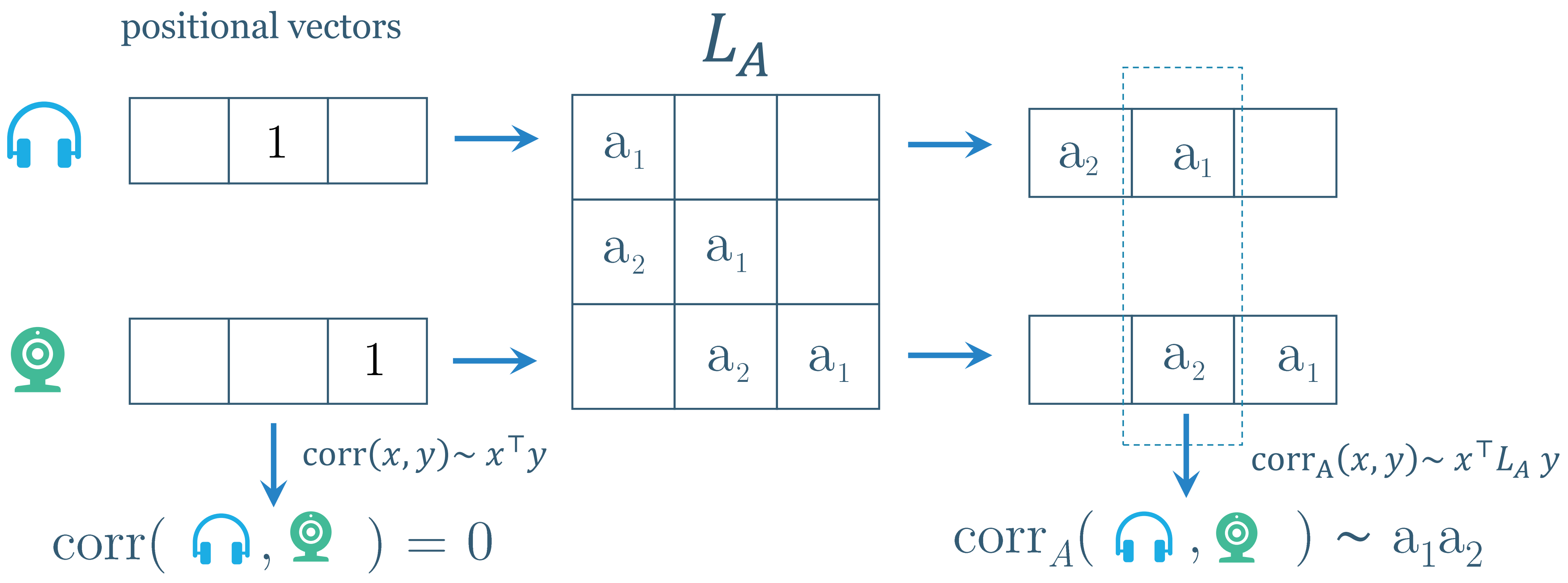




- $S = \left[\delta_{k,k'+1}\right]_{k,k'=1}^{K}$ – "shift operator"

$$\mathrm{toprec}(P, n) \coloneqq \overset{n}{\operatorname{argmax}}\, V V^\top P S\, W w_K$$

Imposing directed positional correlations



- Weighting example: $a_k = k^{-f},\ f \geq 0.$
- How to incorporate into factorization model?

Hybrid TD

Higher order generalization of HybridSVD*

An auxiliary tensor can be represented in the form ** :

$$\mathcal{A} = \mathcal{A}_0 \times_1 L_K^\top \times_2 L_S^\top \times_3 L_A^\top, \qquad L_K L_K^\top = K, \quad L_S L_S^\top = S, \quad L_A L_A^\top = A$$

Connection between the auxiliary and the original representation:

$$L_K^{-\top} U = U_0, \qquad L_S^{-\top} V = V_0, \qquad L_A^{-\top} W = W_0$$

Higher order generalization of hybrid folding-in

Matrix of predicted user preferences for item-context:

$$P \approx V V_S^\top A W_R W^\top, \qquad V_S = L_S V, \quad W_R = L_A W$$

*E.Frolov, and I.Oseledets. "HybridSVD: when collaborative information is not enough." In *Proceedings of the 13th ACM conference on recommender systems*, 2019.
**E.Frolov, and I.Oseledets. "Revealing the Unobserved by Linking Collaborative Behavior and Side Knowledge." *arXiv preprint arXiv:1807.10634* (2018).

Implementation of the hybrid HOOI

Input : Tensor $\mathcal{A}$ in sparse format
Tensor decomposition ranks d_1, d_2, d_3
Cholesky factors L_K, L_S, L_R

Output: auxiliary low rank representation $\mathcal{G}, U, V, W$

Initialize V, W by random matrices with orthonormal columns.

Compute $V_S = L_S V,\ W_A = L_A W$.

Repeat:

$U \leftarrow d_1$ leading left singular vectors of $L_K^T A^{(1)}(W_A \otimes V_S)$,

$U_K \leftarrow L_K U$,

$V \leftarrow d_2$ leading left singular vectors of $L_S^T A^{(2)}(W_A \otimes U_K)$,

$V_S \leftarrow L_S V$,

$W, \Sigma, Z \leftarrow d_3$ leading left singular vectors of $L_A^T A^{(3)}(V_S \otimes U_K)$,

$W_S \leftarrow L_A W$,

$\mathcal{G} \leftarrow$ reshape matrix ΣZ^T into shape (d_3, d_1, d_2) and transpose.

Until: *norm of $\mathcal{G}$ ceases to grow or algorithm exceeds maximum number of iterations.*

Positional TF summary

Optimization task (solved via hybrid HOOI):

$$||\mathcal{A}_0 \times_3 L_A^\top - \mathcal{R}||_F^2 \to \min$$

$$\mathcal{R} = \mathcal{G} \times_1 U \times_2 V \times_3 W$$

Scores prediction (hybrid HO folding-in):

$$R = VV^\top P L_A W \widetilde{W}^\top, \qquad \widetilde{W} = L_A^{-\top} W$$

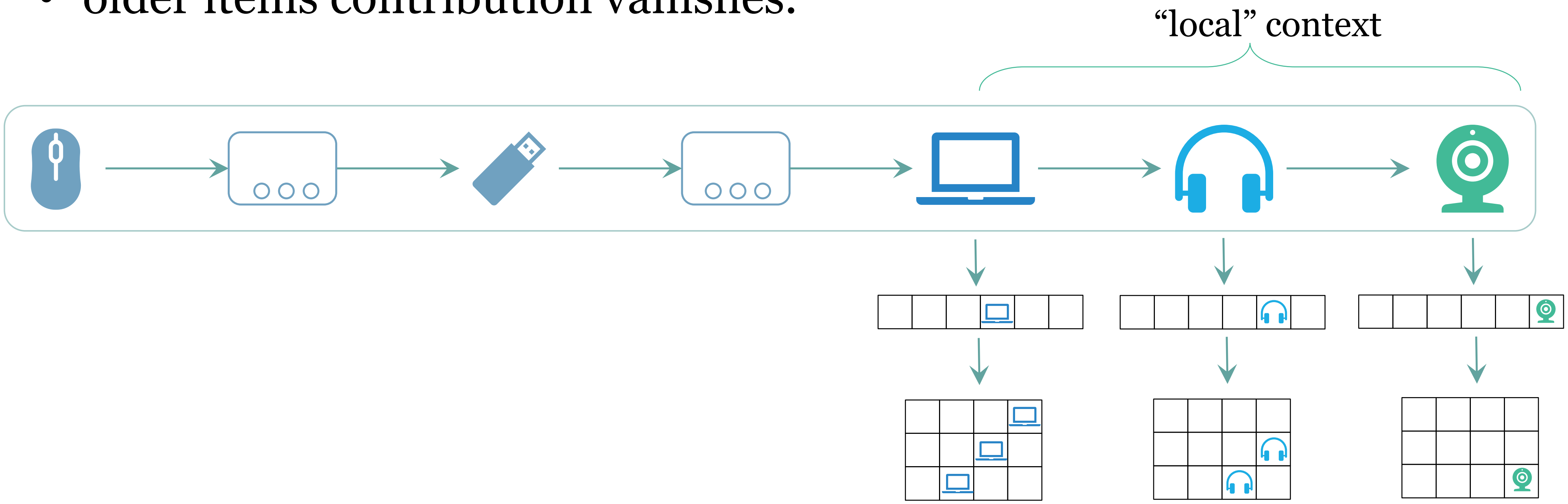
Next item recommendation

$$\text{toprec}(P, n) \coloneqq \operatorname*{argmax}^{n} VV^\top PS\, L_A W \widetilde{w}_K$$

Hands-on

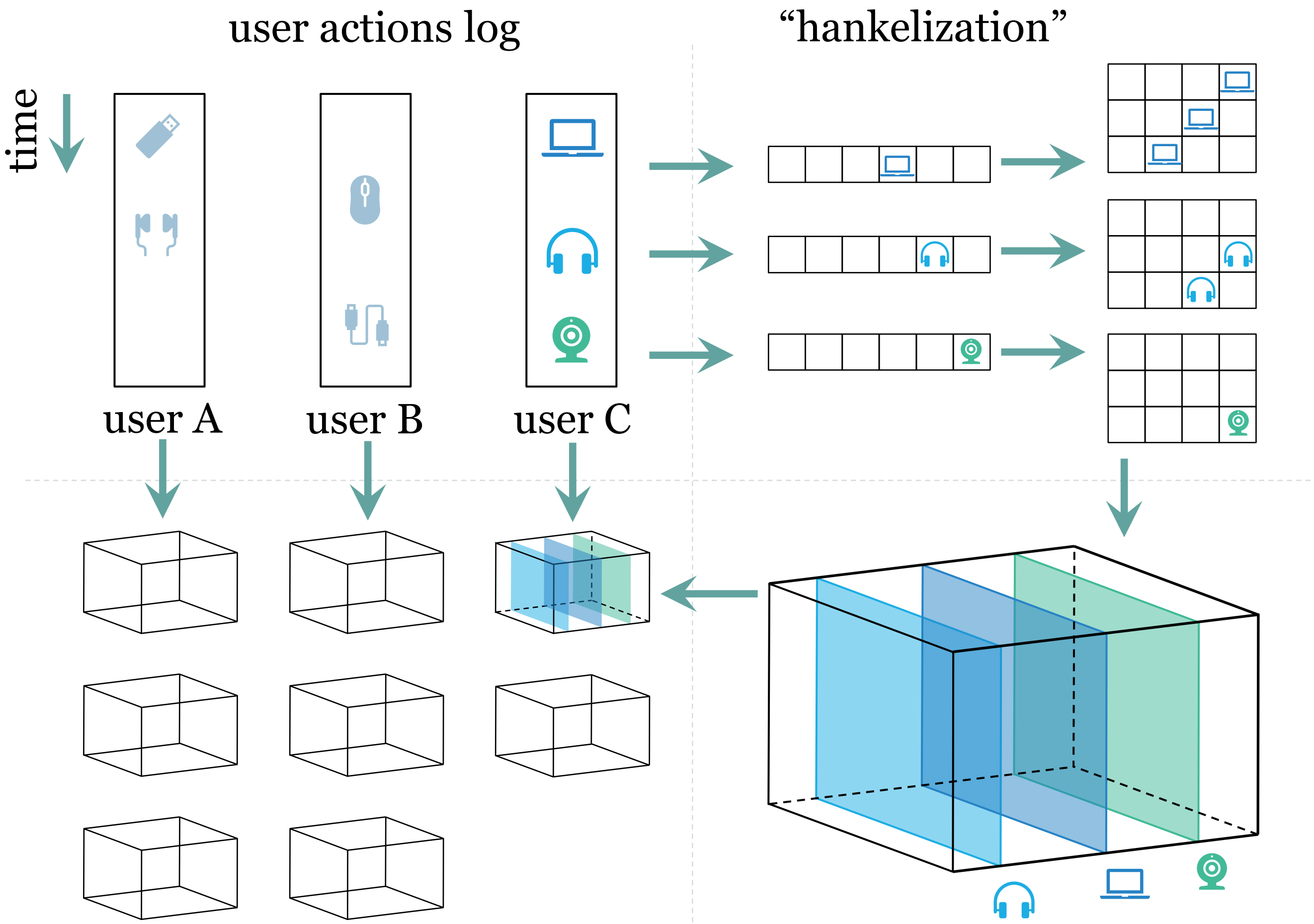
Localizing Attention via Hankelization

- Long-range patterns vs. local context:
  - influence of most recent items is higher,
  - older items contribution vanishes.



column of a Hankel matrix = positional vector within a context window

Hankelized sequence representation



Relevance prediction model:

$$\mathcal{R} = \mathcal{G} \times_1 U \times_2 V \times_3 W_L \times_4 W_S$$

U – user embeddings

V – item embeddings

W_L – positional embedding within local context

W_S – positional embedding at long range

Some results*

Top-10 recommendations quality for Amazon Beauty/Games, Movielens-1M, Steam datasets.

sequential TF without hankelization → GA-SATF

sequential TF with hankelization → LA-SATF

		MP	PureSVD	PureSVD-N	GA-SATF	LA-SATF	SASRec
amz-b	NDCG	0.002 ± 0.000	0.046 ± 0.002	0.047 ± 0.002	0.043 ± 0.002	$\mathbf{0.067} \pm 0.003$	$\underline{0.055} \pm 0.003$
	HR	0.004 ± 0.001	0.082 ± 0.004	0.087 ± 0.004	0.079 ± 0.004	$\mathbf{0.114} \pm 0.005$	$\underline{0.100} \pm 0.004$
	COV	0.007	0.251	0.615	0.182	$\underline{0.608}$	**0.611**
amz-g	NDCG	0.002 ± 0.000	0.042 ± 0.002	$\mathbf{0.058} \pm 0.003$	0.046 ± 0.003	0.052 ± 0.003	$\underline{0.055} \pm 0.003$
	HR	0.003 ± 0.001	0.070 ± 0.004	$\mathbf{0.101} \pm 0.004$	0.074 ± 0.004	0.092 ± 0.004	$\underline{0.094} \pm 0.004$
	COV	0.008	0.467	$\underline{0.631}$	0.241	0.426	**0.700**
ml-1m	NDCG	0.000 ± 0.000	0.029 ± 0.002	0.030 ± 0.002	0.061 ± 0.002	$\mathbf{0.072} \pm 0.003$	$\underline{0.069} \pm 0.002$
	HR	0.000 ± 0.000	0.060 ± 0.003	0.061 ± 0.003	0.112 ± 0.004	$\underline{0.132} \pm 0.004$	$\mathbf{0.134} \pm 0.004$
	COV	0.038	0.187	0.275	0.288	**0.511**	$\underline{0.503}$
steam	NDCG	0.000 ± 0.000	0.020 ± 0.001	0.043 ± 0.002	0.007 ± 0.001	$\underline{0.047} \pm 0.002$	$\mathbf{0.060} \pm 0.002$
	HR	0.000 ± 0.000	0.039 ± 0.002	0.084 ± 0.003	0.013 ± 0.001	$\underline{0.091} \pm 0.003$	$\mathbf{0.115} \pm 0.004$
	COV	0.018	0.070	**0.438**	0.047	$\underline{0.368}$	0.080

SASRec – self-attentive sequential recommendation model ([*Kang & McAuley 2018*])

PureSVD-N – normalized variant of PureSVD (*[Nikolakopoulos et al. 2019]*)

MP – most-popular items recommendation

* currently under review