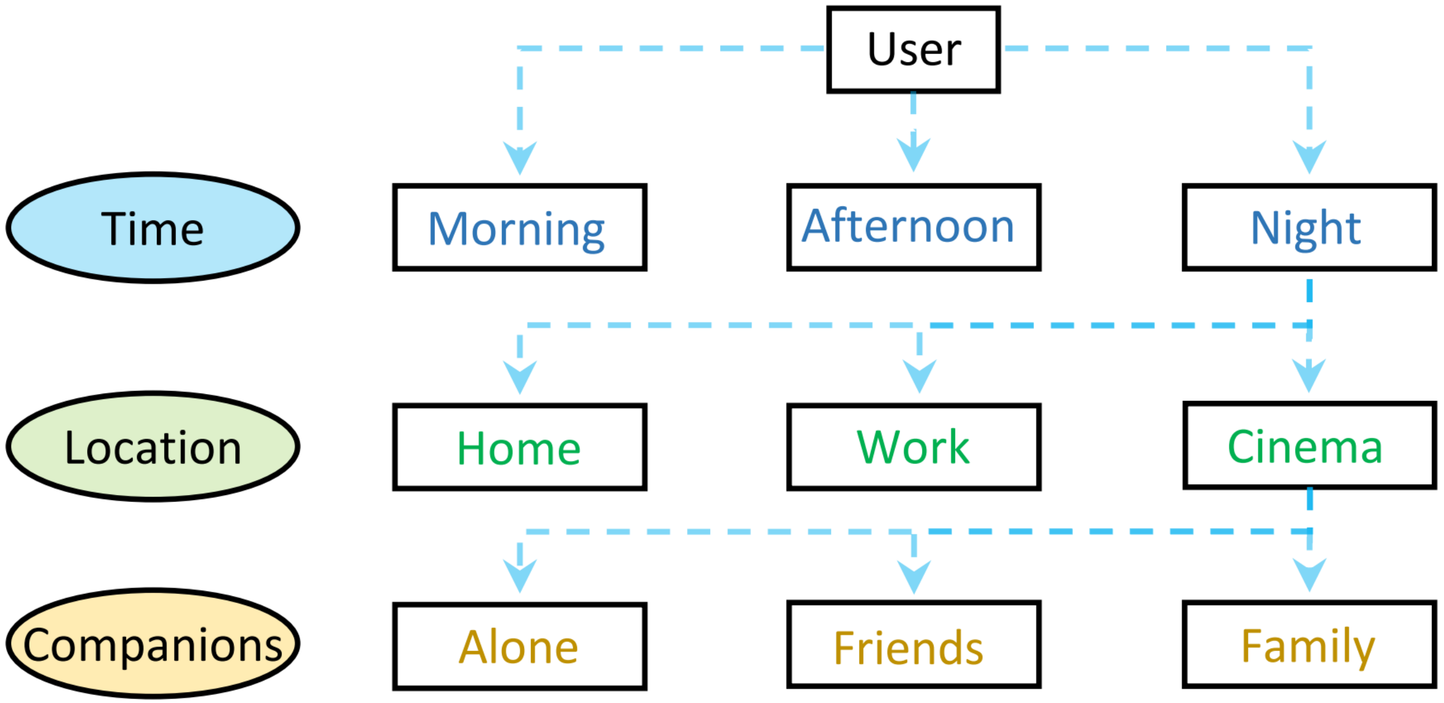
Introduction to Recommender Systems

Lecture 10

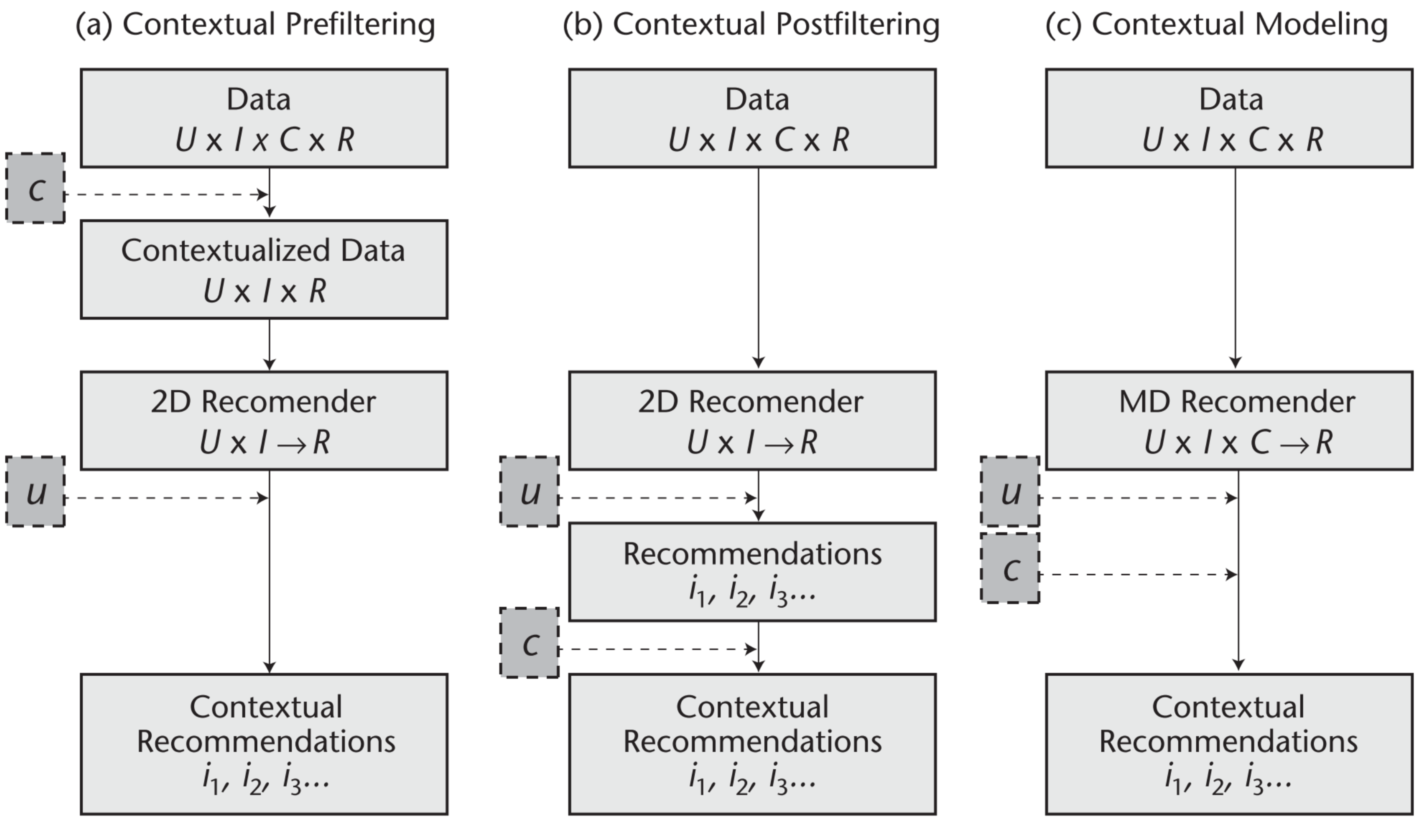
Previous lecture



$$f_U\colon \text{User} \times \text{Item} \times \text{Context}_1 \times \cdots \times \text{Context}_f \rightarrow \text{Relevance}$$

Previous lecture

- When to opt for simpler post- and pre-filtering models vs contextual modeling?



Today's lecture

- Time- and sequence-aware models

Disclaimer

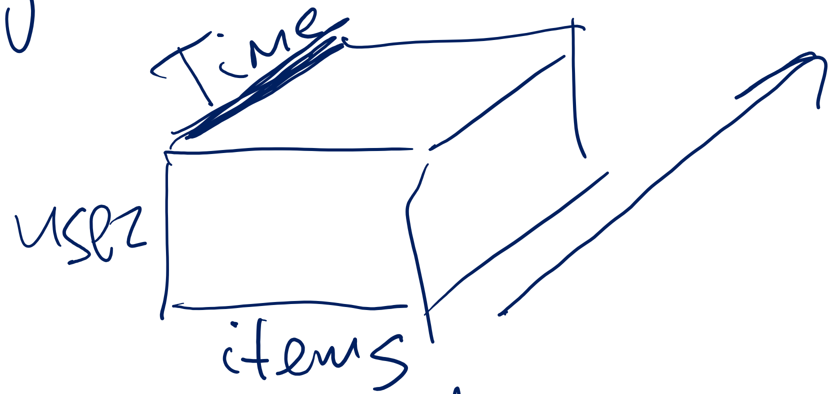
- Our course deviates from the main book in the way ratings prediction task is treated.
- A lot of reasoning for temporal models in the book is tied to the Netflix Prize era models.
- While all the statements in the book are correct in the context of ratings prediction, they do not reveal a full picture.
- We won't go deep into the details of early temporal models. But you may still want to learn about them for an example of feature engineering in the time domain.

Temporal dynamics in data

- shift of preferences:
  - users’ interests may drift over time
  - items’ relevance / perceived quality may change over time
- short/long-term dynamics, bursty/gradual changes
  - users gradually gaining expertise in a domain – harder to satisfy
  - items may become more relevant during a limited promotion period
- periodic effects may also play a role
  - seasonality of certain categories
  - repeated consumption patterns

Examples of temporal modeling

BPTF 2010 Time - categorical



$$\|A_0 - R\|_F^2 + \lambda \underbrace{\sum_{t=1}^{T-1} \|W^{(t+1)} - W^{(t)}\|_F^2}$$

$$R = [\![G; U, V, W]\!]$$

$$\dot{A} = \frac{dA}{dt} \approx A^{(t+1)} - A^{(t)}$$

if small $\rightarrow \underbrace{R^{(t+1)} - R^{(t)}}_{\text{also small}}$

$$R(t) = U(t)\,\Sigma(t)\,V(t)^T$$

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

$$\underbrace{\|\dot{A} - \dot{R}\|_F^2 \to \min}$$

Dynamical low-rank approximation

Xiong, Liang, Xi Chen, Tzu-Kuo Huang, Jeff Schneider, and Jaime G. Carbonell. "Temporal collaborative filtering with bayesian probabilistic tensor factorization." In Proceedings of the 2010 SIAM international conference on data mining, pp. 211-222. Society for Industrial and Applied Mathematics, 2010.

Olaleke, Oluwafemi, Ivan Oseledets, and Evgeny Frolov. "Dynamic modeling of user preferences for stable recommendations." In *Proceedings of the 29th ACM Conference on User Modeling, Adaptation and Personalization*, pp. 262-266. 2021.

Sequence-aware learning

- short vs long-term sequential patterns
  - short means history is unavailable – anonymous sessions
- new scenarios different from CF:
  - reminders
  - a full list of top recs in particular order can be considered
    - e.g., recommendation of a series of learning courses or music playlist

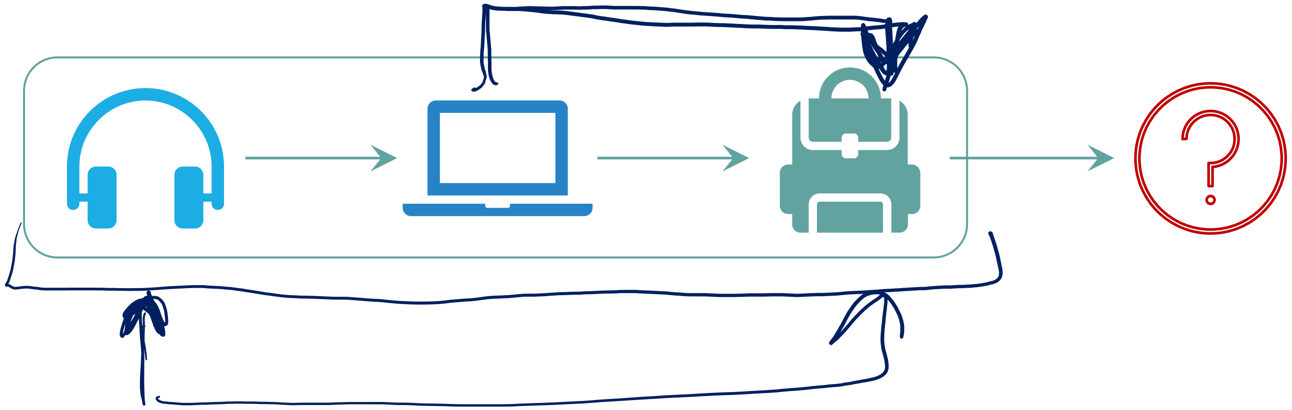
- Types of models:
  - sequential
  - session-aware
  - session-based

Session-aware vs session-based recommendations

Image source: Latifi, Sara, Mauro, Noemi and Jannach, Dietmar. 2021. "Session-aware recommendation: A surprising quest for the state-of-the-art." Sciences 573 :291-315. https://doi.org/10.1016/j.ins.2021.05.048.

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?

Sequential Models

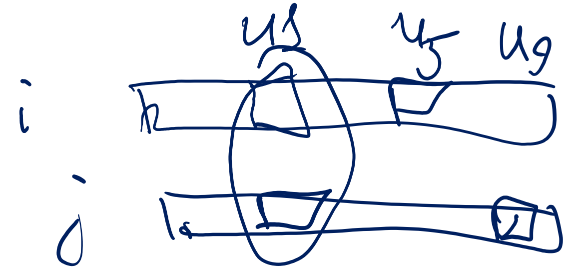
- Sequential AR
- Sequential KNN
- FOSSIL
- FPMC
- Positional TF

Association rules reminder

Assuming no reoccurring consumption in user history:

$$\text{score}_{\text{AR}}(u,i) = \sum_{j \in I_u \setminus \{i\}} \text{pairCount}(i,j), \qquad \text{pairCount}(i,j) = U_i \cap U_j$$

$R = Ca$



More general form:

$$\text{score}_{\text{AR}}(u,i) = \frac{1}{|I_u - 1|^{\beta} \cdot |U_i|^{\alpha}} \sum_{j \in I_u \setminus \{i\}} U_i \cap U_j$$

Unordered!

$$U_i \cap U_j = \sum_{v \in U_i} \mathbb{I}(j \in I_v) = \sum_{v \in U_j} \mathbb{I}(i \in I_v) = U_j \cap U_i$$

Association rules for next item prediction

- Ideas:
  - impose order for item co-occurrence
  - use Markov Chain principle

$$\boxed{U_j \cap U_i}$$

$$\boxed{j \rightarrow_u i}$$

$$\mathrm{pairCount}(i,j) = \sum_{u \in U} \mathbb{I}(j \rightarrow_u i)$$

$$\mathrm{score}_{\mathrm{AR-NI}}(u,i) = \frac{1}{Z} \cdot \sum_{j \in I_u \setminus \{i\}} \sum_{v \in U} \mathbb{I}(j \rightarrow_v i)$$

Association rules for next item prediction

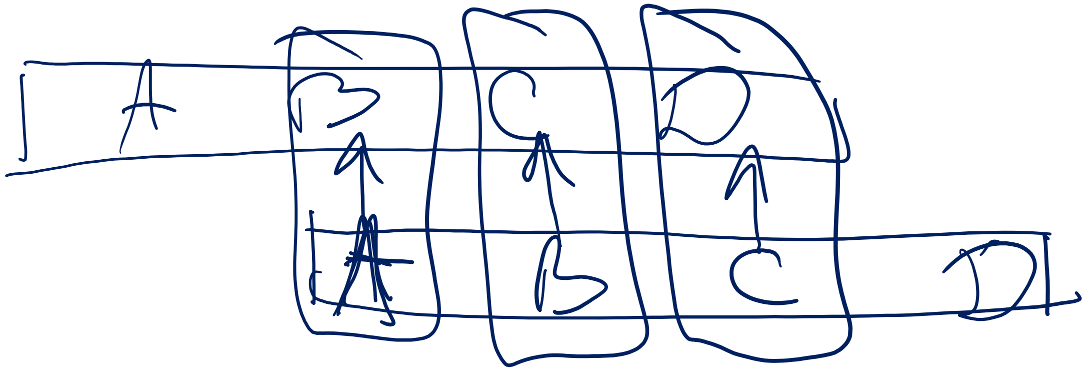
$P(i^{(t+1)} \mid i^{(t)} i^{(t-1)} \ldots) \approx p(i^{(t+1)} \mid i^{(t)})$

- Ideas:
  - impose order for item co-occurrence
  - use Markov Chain principle – only two subsequent events count
- New notation:
  - $j \to_u i$ – item i goes immediately after item j in a history of user u

$$\text{pairCount}(j \to i) = \sum_{v \in U} \mathbb{I}(j \to_v i)$$

$$\text{score}_{\text{AR-NI}}(u, i) = \frac{1}{Z} \cdot \sum_{j \in I_u \setminus \{i\}} \sum_{v \in U} \mathbb{I}(j \to_v i)$$

Example of dataset construction



Strict next item prediction

Idea: count co-occurrences with the last item of current user u

Notation:

- $i_{|I_u|}$ – the last item in the sequence of items of user u

$$\text{score}_{\text{AR-NI}}(u, i) = \frac{1}{z} \cdot \sum_{j \in I_u \setminus \{i\}} \mathbb{I}(j = i_{|I_u|}) \cdot \text{pairCount}(j \to i)$$

Simplifies to:

$$\text{score}_{\text{AR-NI}}(u, i) = \frac{1}{z} \cdot \text{pairCount}(i_{|I_u|} \to i)$$

Association rules for next item prediction

What if data is too sparse (i.e., not enough ordered pairs)?

Nearest Neighbors Models

Recall,

$$\text{score}_{\text{uKNN}}(u,i) = \frac{1}{z} \sum_{v \in \mathcal{N}_i(u)} \text{sim}(u,v) \cdot a_{vi}$$

For binary data we can also rewrite it as:

$$\text{score}_{\text{uKNN}}(u,i) = \frac{1}{z} \sum_{v \in \mathcal{N}(u)} \text{sim}(u,v) \cdot \mathbb{I}(i \in I_v)$$

How to utilize sequential information?

position

$p_i^{(u)}$ $p_i^{(v)}$

2 1

$\frac{1}{p_i^{(u)}}$ $\frac{1}{p_j^{(v)}}$

1

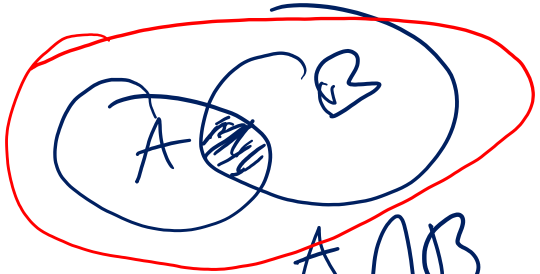
$\frac{1}{200}$

Sequential KNN

$$\text{score}_{\text{SKNN}}(u,i) = \frac{1}{z} \sum_{v \in \mathcal{N}(u)} \text{sim}(u,v) \cdot \mathbb{I}(i \in I_v)$$

VSKNN

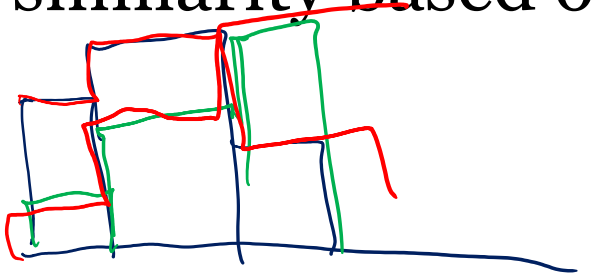
Idea: encode positional information into similarity.



$$\frac{A \cap B}{A \cup B - A \cap B}$$

Options for computing similarity:

- encode reciprocal rank instead of binary data in user vectors
- apply weighted similarity based on positional information



$$WJI = \sum \frac{\min(x,y)}{\max(x,y)} \qquad \frac{1}{p}$$

Latent factors models

- SR models encode interactions based on Markov Chain principles
- likely to inherit the same issues as standard AR models
- possible remedy – low-rank assumption

Factorizing Markov Chain representation

target

$$\underbrace{r_{uij}} \sim \underbrace{q_j^\top q_i}$$

previous item

NSVD

$$\underbrace{?} + \underbrace{\left(\sum_{j \in N_u} q_j\right)^\top q_i}$$

Candidate models:

- Bilinear models (SVDFeature)
- Factorization Machines
- Factorized Sequential Prediction with Item Similarity Models (Fossil)
- Factorized Personalized Markov Chains (FPMC)

Fossil

$$\overbrace{r_{u i j}}^{} \approx \overbrace{\underbrace{p_u^T q_i}_{\text{long-term}}} + \overbrace{(\eta + \eta_u) \underbrace{\bar{q}_j^T q_i}}^{\text{short-term}}$$

target ↑ latest known

$$p_u = \sum_{j \in N(u)} q_j$$

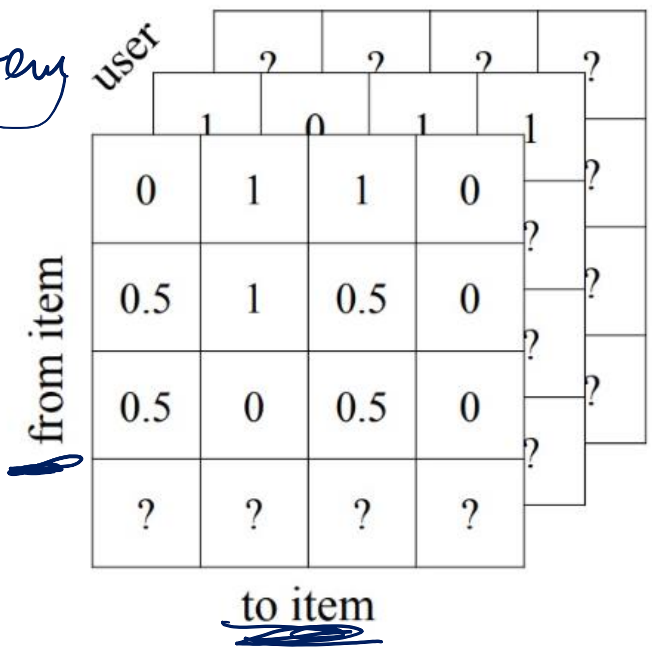
$$\rightarrow \overbrace{p(i^{(t+1)} \mid i^{(t)}, u)}$$

He, Ruining, and Julian McAuley. "Fusing similarity models with markov chains for sparse sequential recommendation." In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pp. 191-200. IEEE, 2016.

Factorized Personalized Markov Chains

Users x Item x Item

$$r_{uij} = \sum_{k=1}^{d} p_{uk} q_{ik} \bar{q}_{jk},$$



- Interactions are restricted to pairwise

$$r_{uij} = \left\langle \boldsymbol{p}_u^{(q)}, \boldsymbol{q}_i^{(p)} \right\rangle + \left\langle \boldsymbol{p}_u^{(\bar{q})}, \bar{\boldsymbol{q}}_j^{(p)} \right\rangle + \cancel{\left\langle \boldsymbol{q}_i^{(\bar{q})}, \bar{\boldsymbol{q}}_j^{(q)} \right\rangle}$$

- ranking doesn't depend on the last term – can be omitted:

$$r_{uij} = \left\langle \boldsymbol{p}_u^{(q)}, \boldsymbol{q}_i^{(p)} \right\rangle + \left\langle \boldsymbol{p}_u^{(\bar{q})}, \bar{\boldsymbol{q}}_j^{(p)} \right\rangle$$

Rendle, Steffen, Christoph Freudenthaler, and Lars Schmidt-Thieme. "Factorizing personalized markov chains for next-basket recommendation." In *Proceedings of the 19th international conference on World wide web*, pp. 811-820. 2010.

- optimized with BPR loss: $-\log \sigma\left(r_{uik} - r_{ujk}\right), \quad i \in I_u^+,\ j \in I_u^-$

Latent factor models with positional information

- position in a sequence is a contextual information
- we can utilize context-aware methods

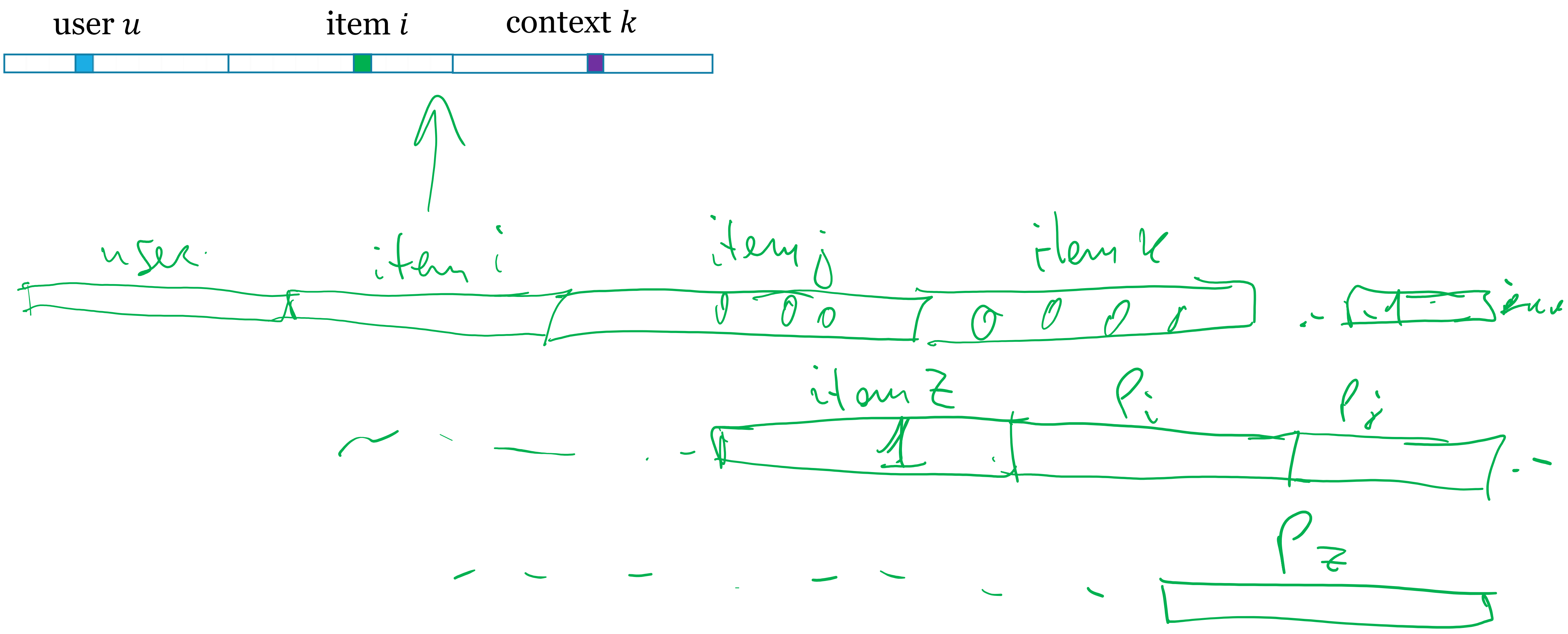
$$f_U\colon \text{User} \times \text{Item} \times \text{Context} \to \text{Relevance}$$

$$\operatorname{toprec}(u, c, n) \coloneqq \arg\max_i^n r_{uic}$$

Candidate models:

- FM
- TF

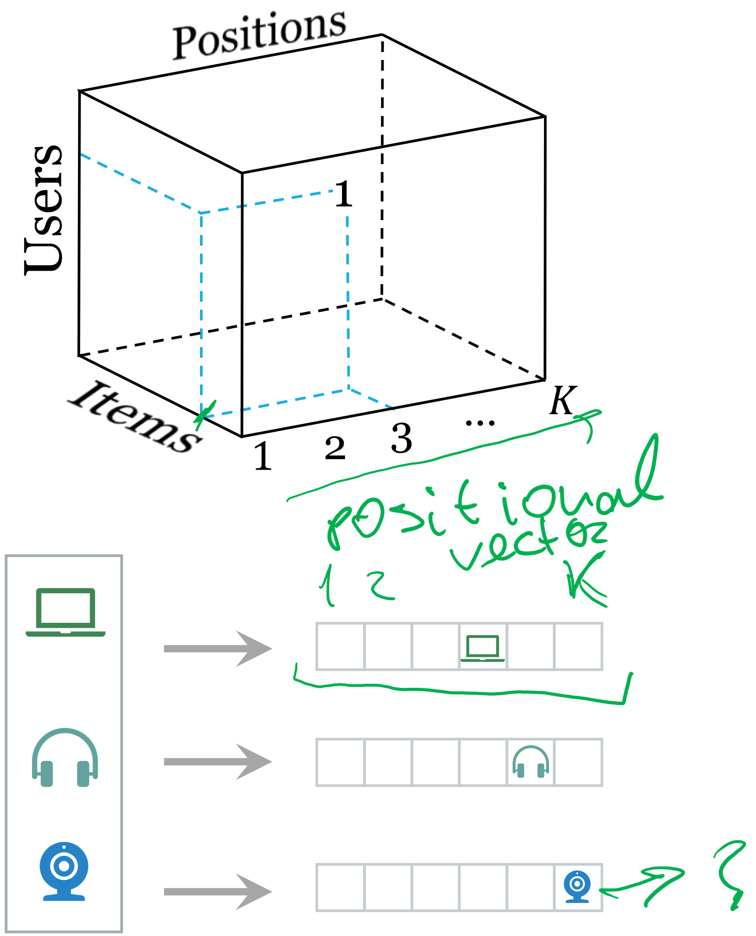
FM representation and complexity



Tensor Factorization with Positional Information

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

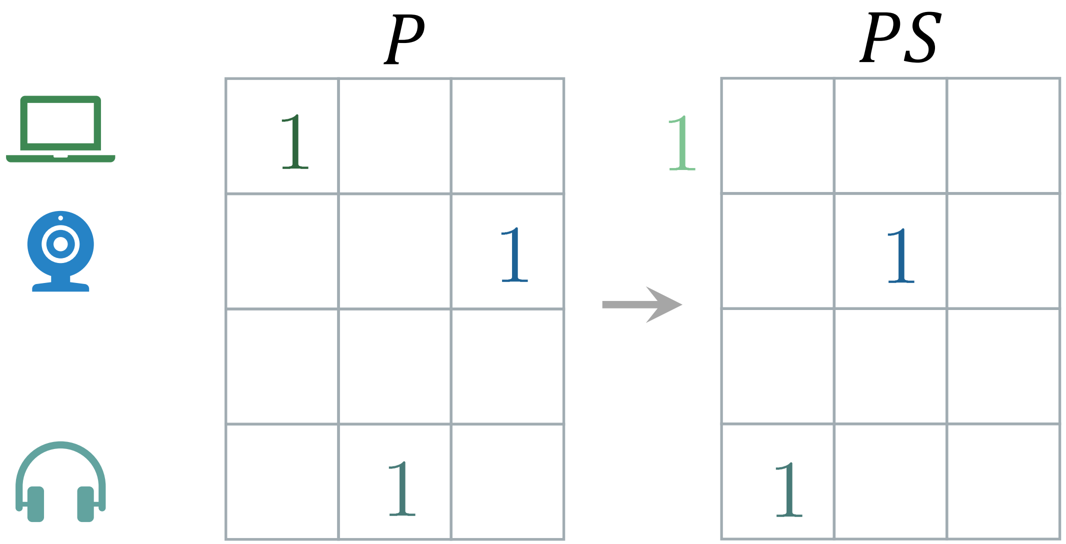
$$\boxed{f_U\colon \text{User} \times \text{Item} \times \text{Position} \rightarrow \text{Relevance Score}}$$

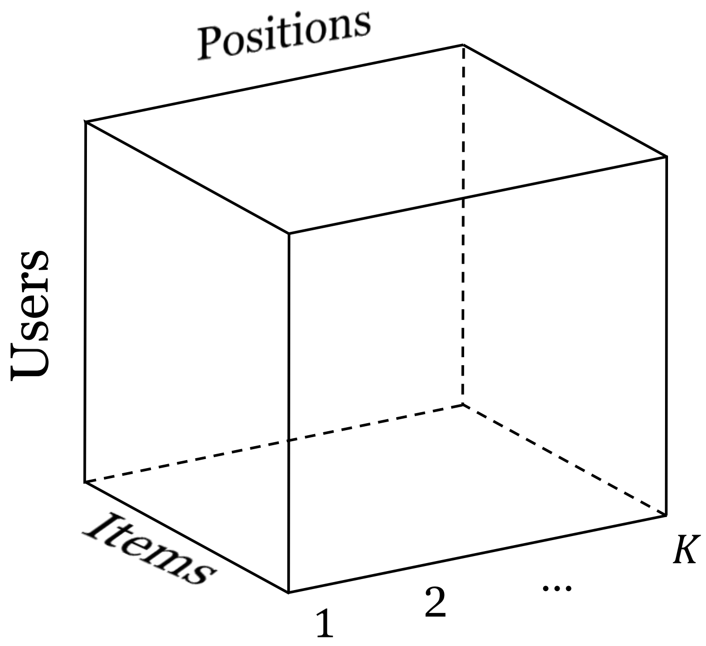


- encode positions as a third categorical entity
- need to handle user sequences of variable length
  - pad with 0
- local vs. global sequential context
  - weighting based on position
- how to generate predictions?

Predicting future interactions with TF

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

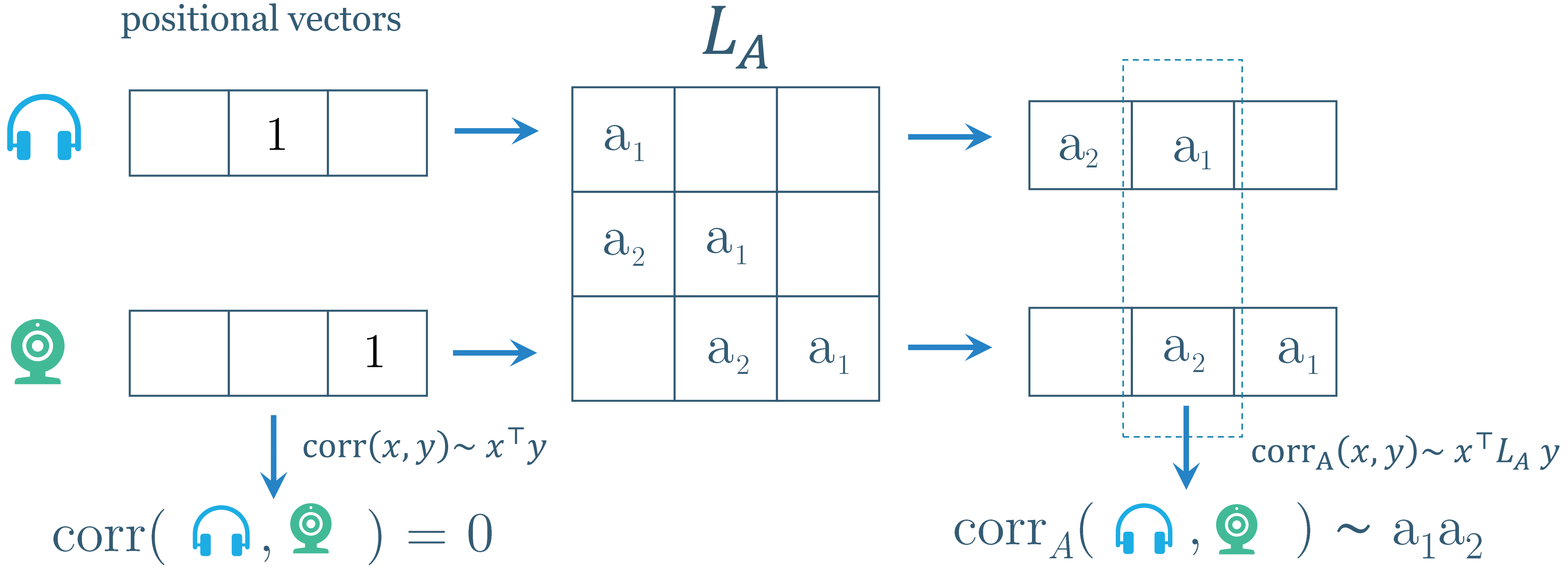




- $S = \left[\delta_{k,k'+1}\right]_{k,k'=1}^{K}$ – "shift operator"

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

Imposing directed positional correlations



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

Hybrid CoFFee

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:

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

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

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

Let’s look at the code