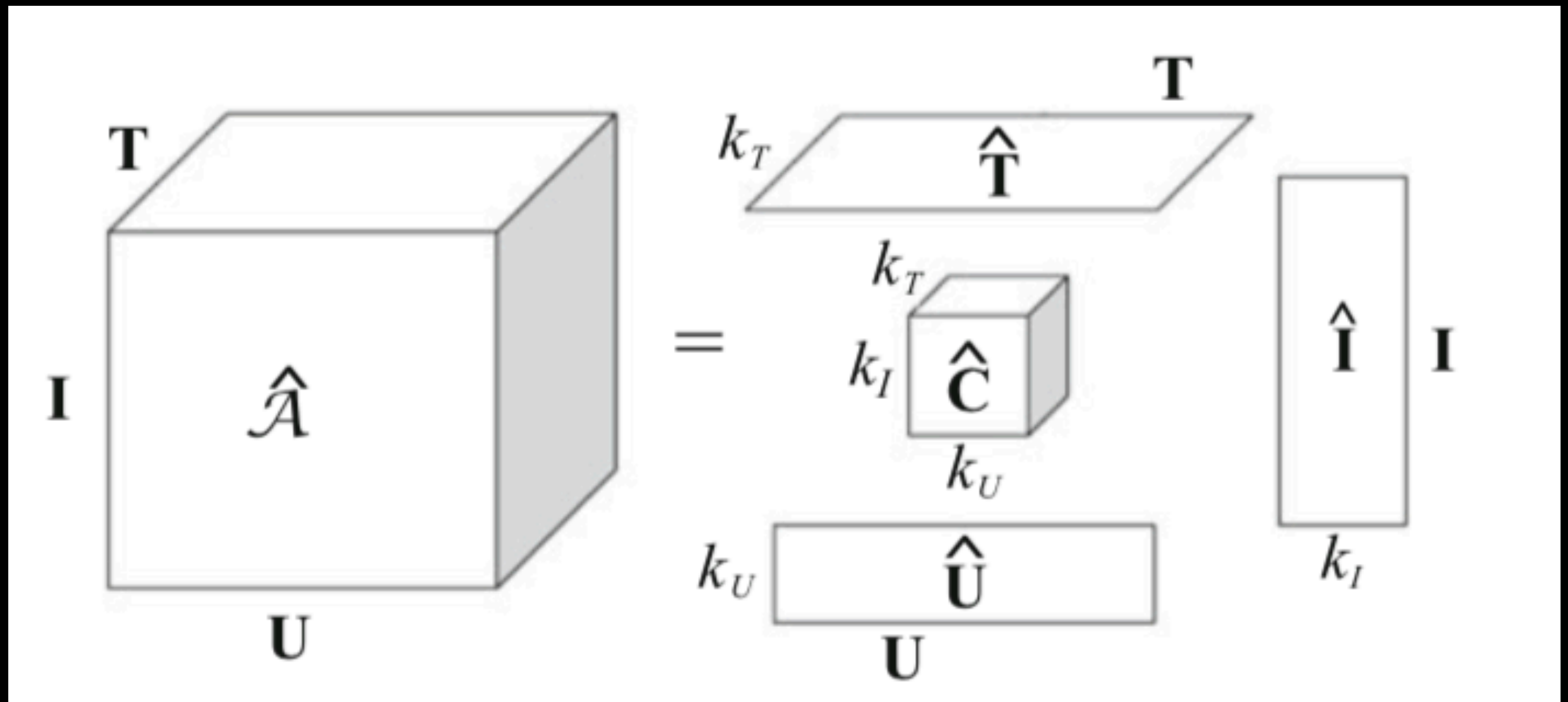
Recommendation Systems

Matrix Factorization. Part II

Eugeny Malyutin / Sergey Dudorov

Tensor Factorization (HOSVD)



Algorithm 5.1 AlsHOSVD

Require: The initial tensor $\mathcal{A}$ with user, tag, and item dimensions.

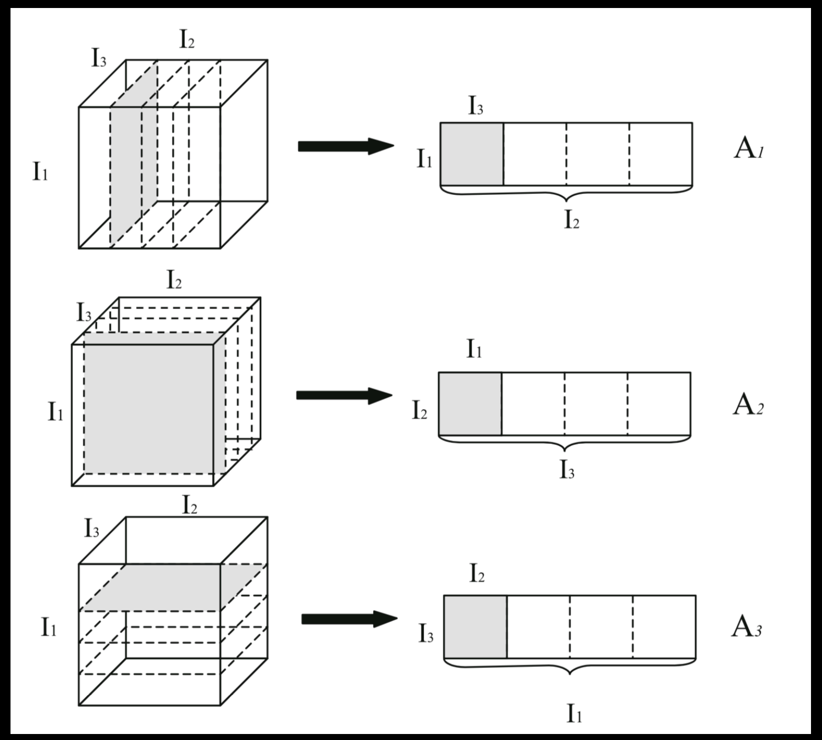
Ensure: The approximate tensor $\hat{\mathcal{A}}$ with k_U, k_I and k_T left leading eigenvectors of each dimension, respectively.

1: Initialize core tensor $\mathcal{C}$ and left singular vectors $U^{(1)}$, $U^{(2)}$, $U^{(3)}$ of A_1, A_2, and A_3, respectively.
2: **repeat**
3: $\quad \mathcal{C} = \mathcal{A} \times_1 {U_{k_U}}^{(1)^T} \times_2 {U_{k_I}}^{(2)^T} \times_3 {U_{k_T}}^{(3)^T}$
4: $\quad \hat{\mathcal{A}} = \mathcal{C} \times_1 {U_{k_U}}^{(1)} \times_2 {U_{k_I}}^{(2)} \times_3 {U_{k_T}}^{(3)}$
5: $\quad {U_{k_U}}^{(1)} \leftarrow k_U$ leading left singular vectors of A_1
6: $\quad {U_{k_I}}^{(2)} \leftarrow k_I$ leading left singular vectors of A_2
7: $\quad {U_{k_T}}^{(3)} \leftarrow k_T$ leading left singular vectors of A_3
8: **until** $\|\mathcal{A} - \hat{\mathcal{A}}\|^2$ ceases to improve **OR** maximum iterations reached
9: **return** $\mathcal{C}$, ${U_{k_U}}^{(1)}$, ${U_{k_I}}^{(2)}$, and ${U_{k_T}}^{(3)}$

$$\hat{\mathscr{A}} := \hat{C} \times_u \hat{U} \times_i \hat{I} \times_t \hat{T}$$

$$\underset{\hat{\theta}}{\operatorname{argmin}} \sum_{(u,i,t)\in Y} \left(\hat{a}_{u,i,t} - a_{u,i,t}\right)^2$$

$$\hat{a}(u,i,t) := \sum_{\tilde{u}=1}^{k_U} \sum_{\tilde{i}=1}^{k_I} \sum_{\tilde{t}=1}^{k_T} \hat{c}_{\tilde{u},\tilde{i},\tilde{t}} \cdot \hat{u}_{u,\tilde{u}} \cdot \hat{i}_{i,\tilde{i}} \cdot \hat{t}_{t,\tilde{t}}$$



$$A_1 = U^{(1)} \cdot S_1 \cdot V_1^T A_2 = U^{(2)} \cdot S_2 \cdot V_2^T A_3 = U^{(3)} \cdot S_3 \cdot V_3^T$$

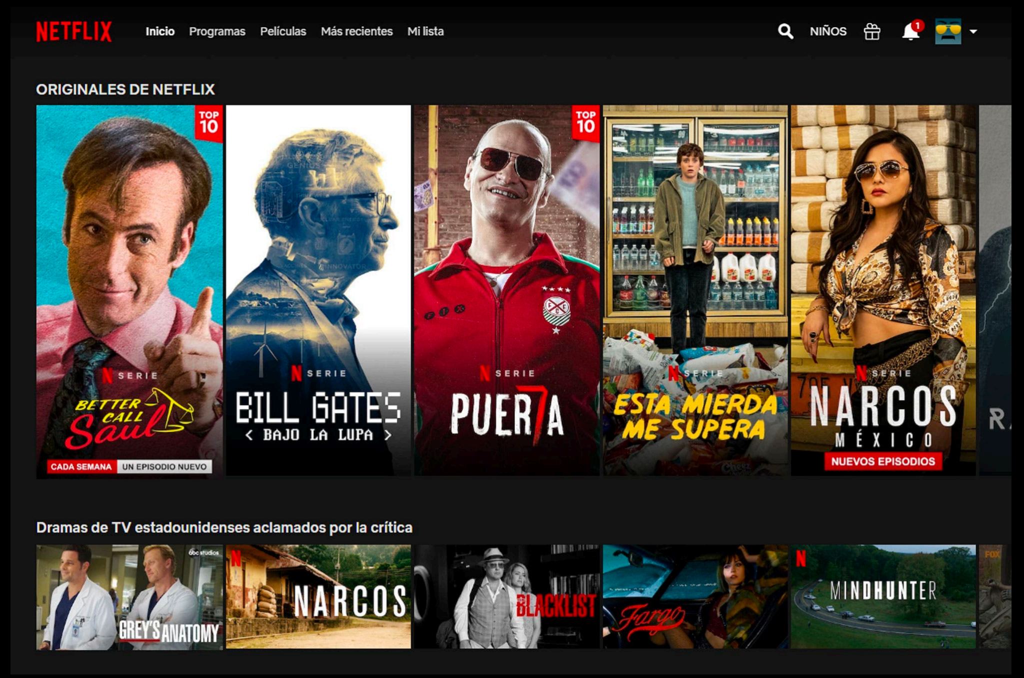
- Matrix and Tensor Factorization Techniques for Recommender Systems ISBN 978-3-319-41357-0

5 - 4 = 3 -2 ?

- RMSE -> approximate number

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} (Predicted_i - Actual_i)^2}{N}}$$

- Recommended systems is about ranking, not about numbers

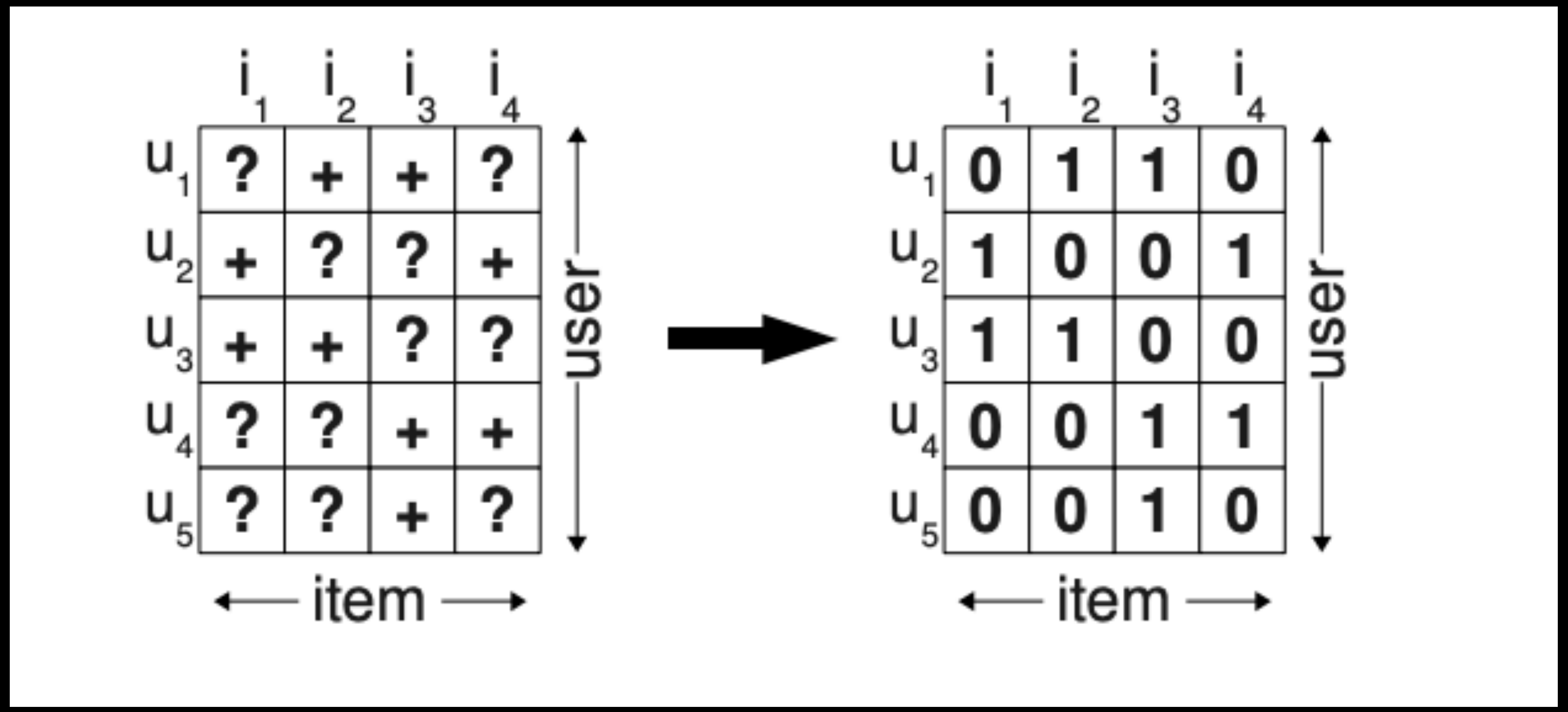


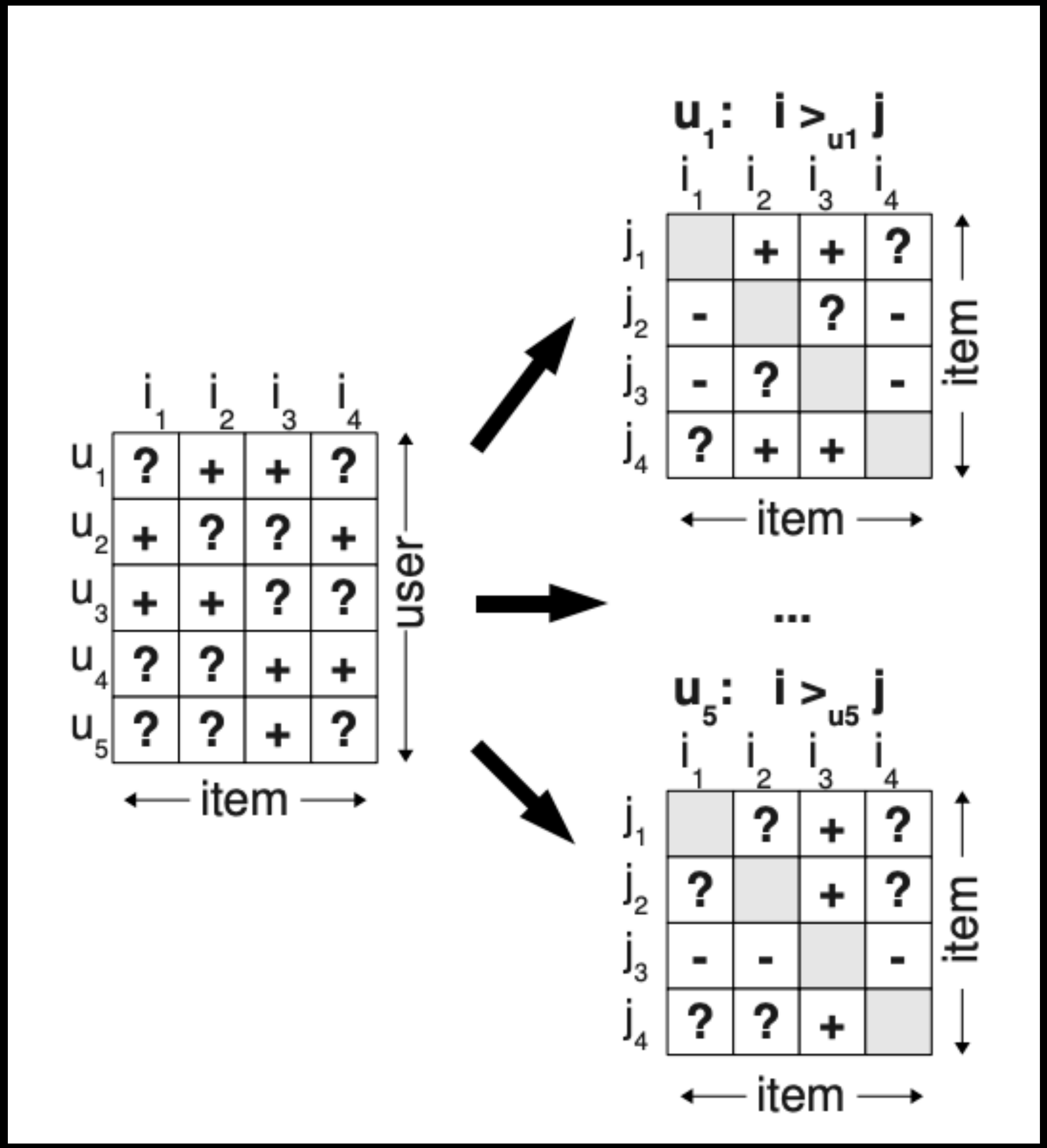
$5 > 4 > 3 > 2 > 1 > 0$

$Positives > Negatives$

BPR

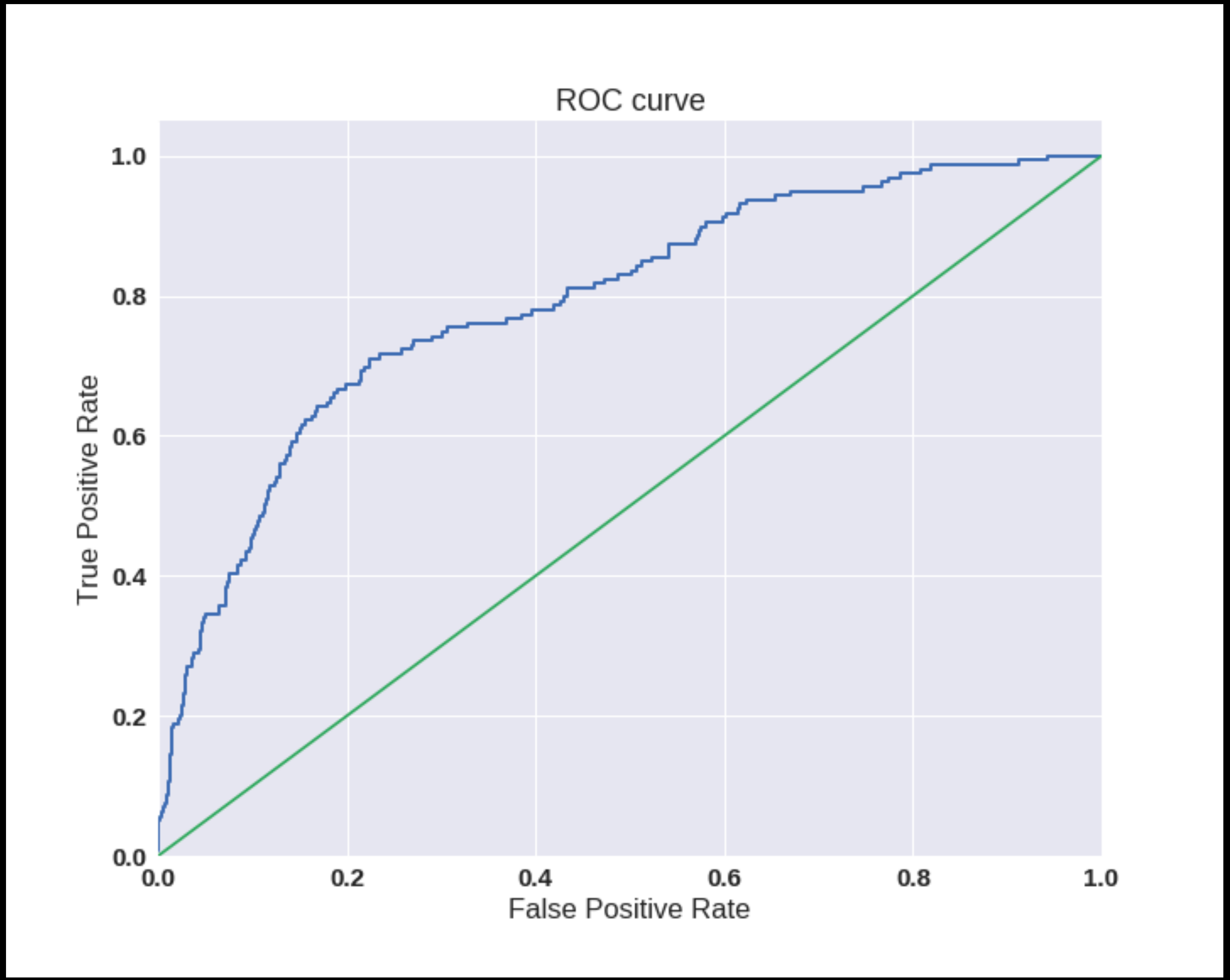
- BPR: Bayesian Personalized Ranking from Implicit Feedback





https://arxiv.org/pdf/1205.2618.pdf

BPR -> AUC



$$\mathrm{AUC}(u) = \frac{1}{\left|I_u^+\right| \left|I \backslash I_u^+\right|} \sum_{i \in I_u^+} \sum_{j \in \left|I \backslash I_u^+\right|} \delta\left(\hat{x}_{uij} > 0\right)$$

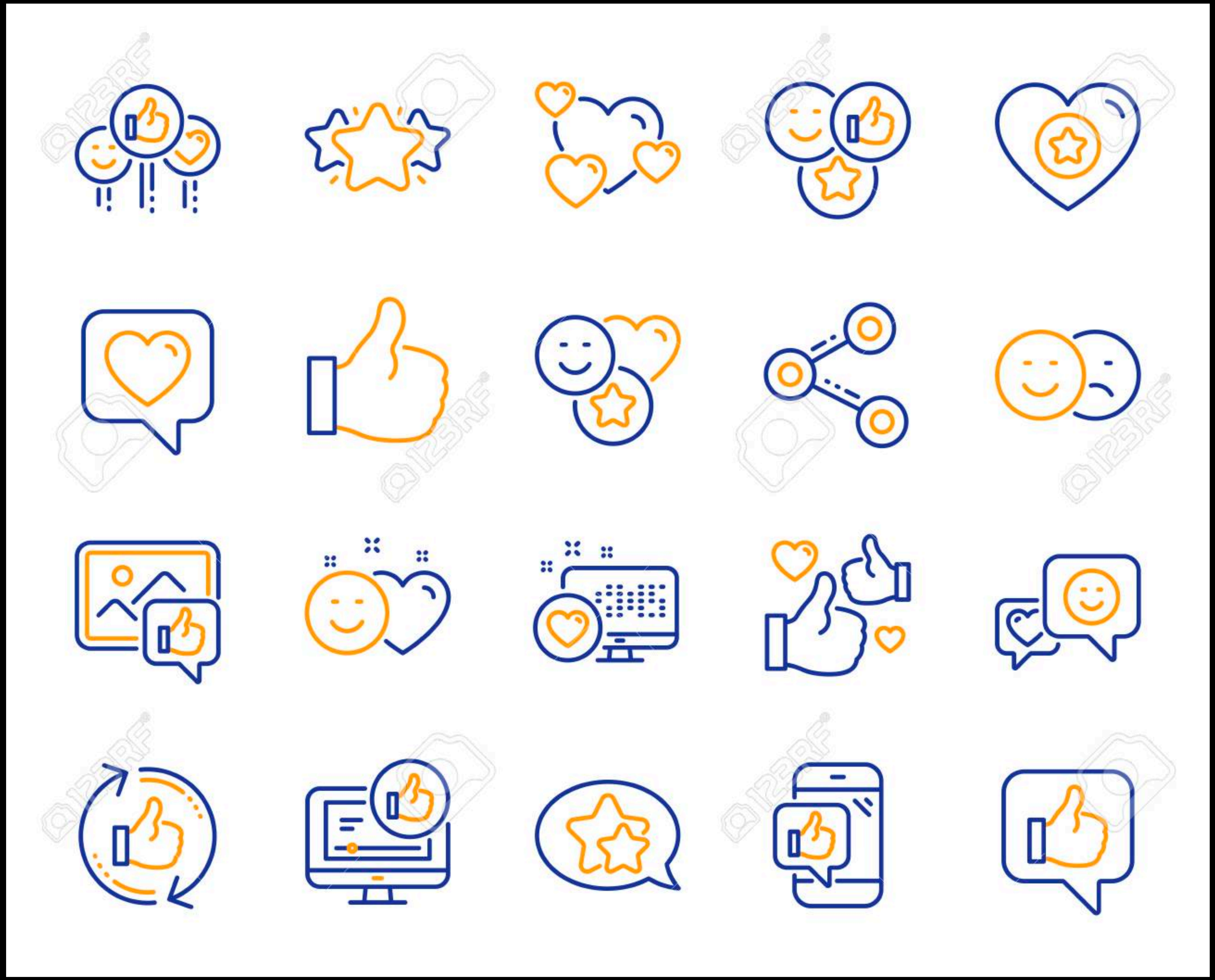
$$\mathrm{AUC} = \frac{1}{|U|} \sum_{u \in U} AUC(u)$$

$$\mathrm{AUC}(u) = \sum_{(u,i,j) \in D_S} z_u \delta\left(\hat{x}_{uij} > 0\right)$$

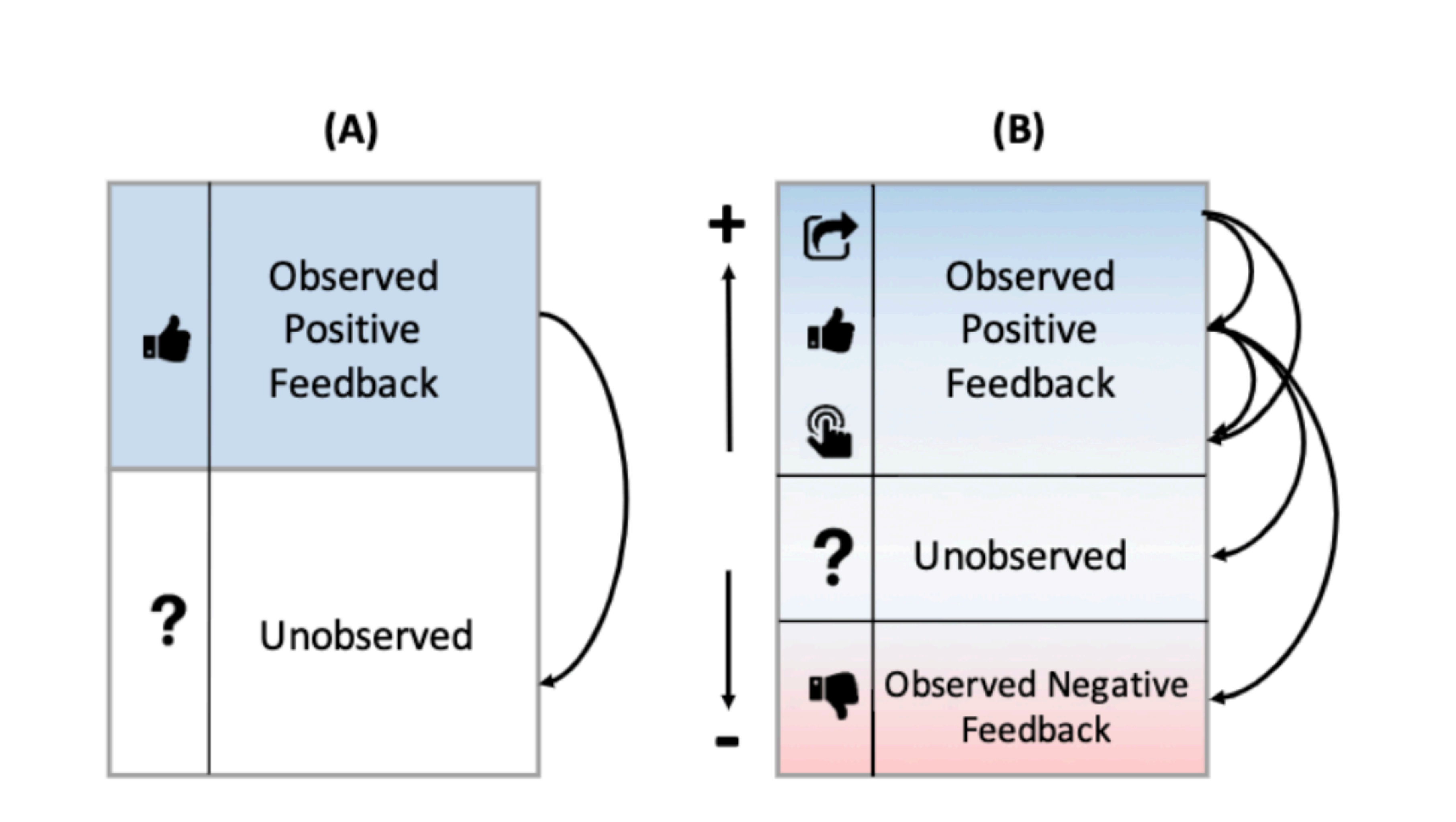
$$z_u = \frac{1}{|U| \left|I_u^+\right| \left|I \backslash I_u^+\right|}$$

https://arxiv.org/pdf/1205.2618.pdf

Learn on likes or shares?



MultiChannel BPR

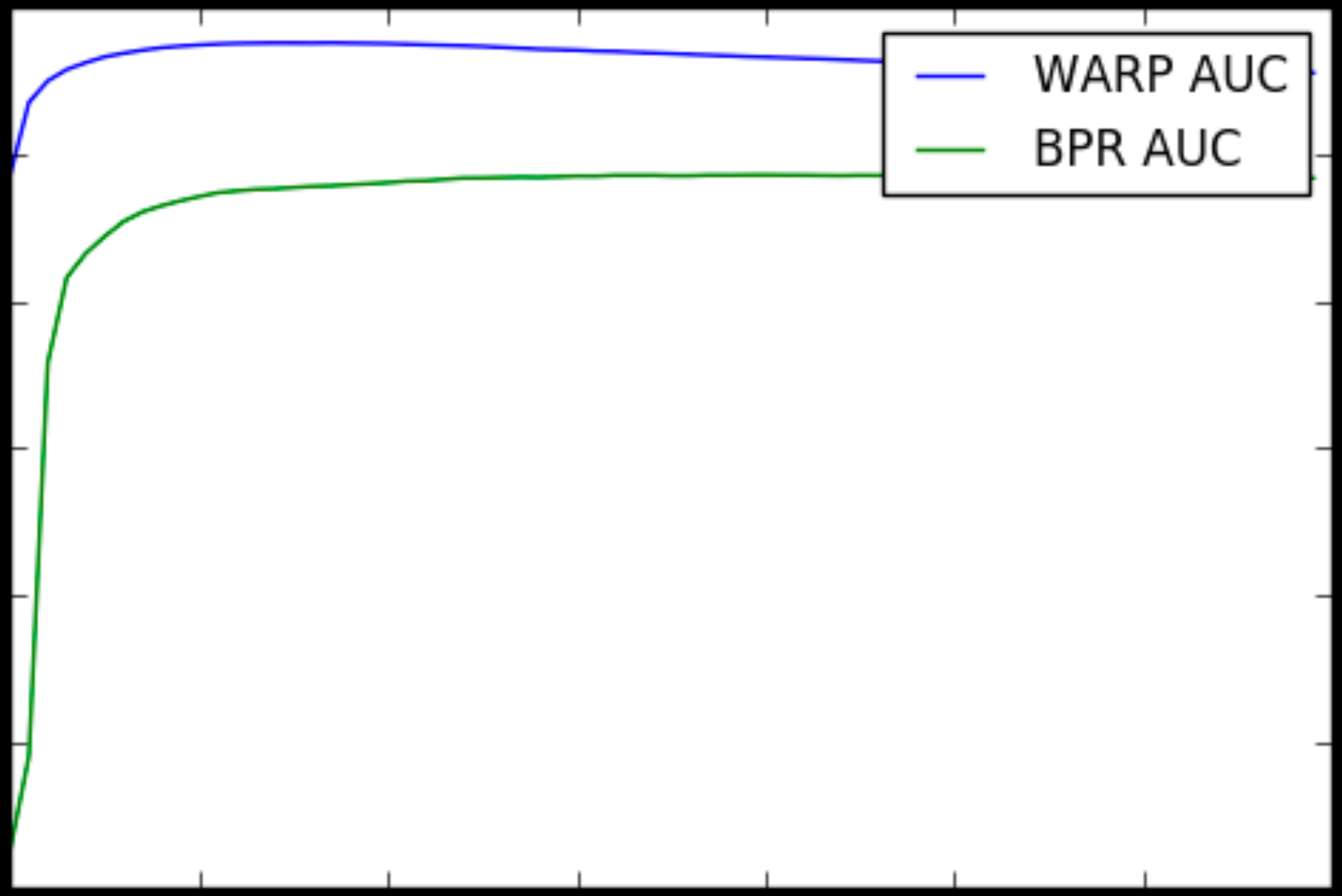


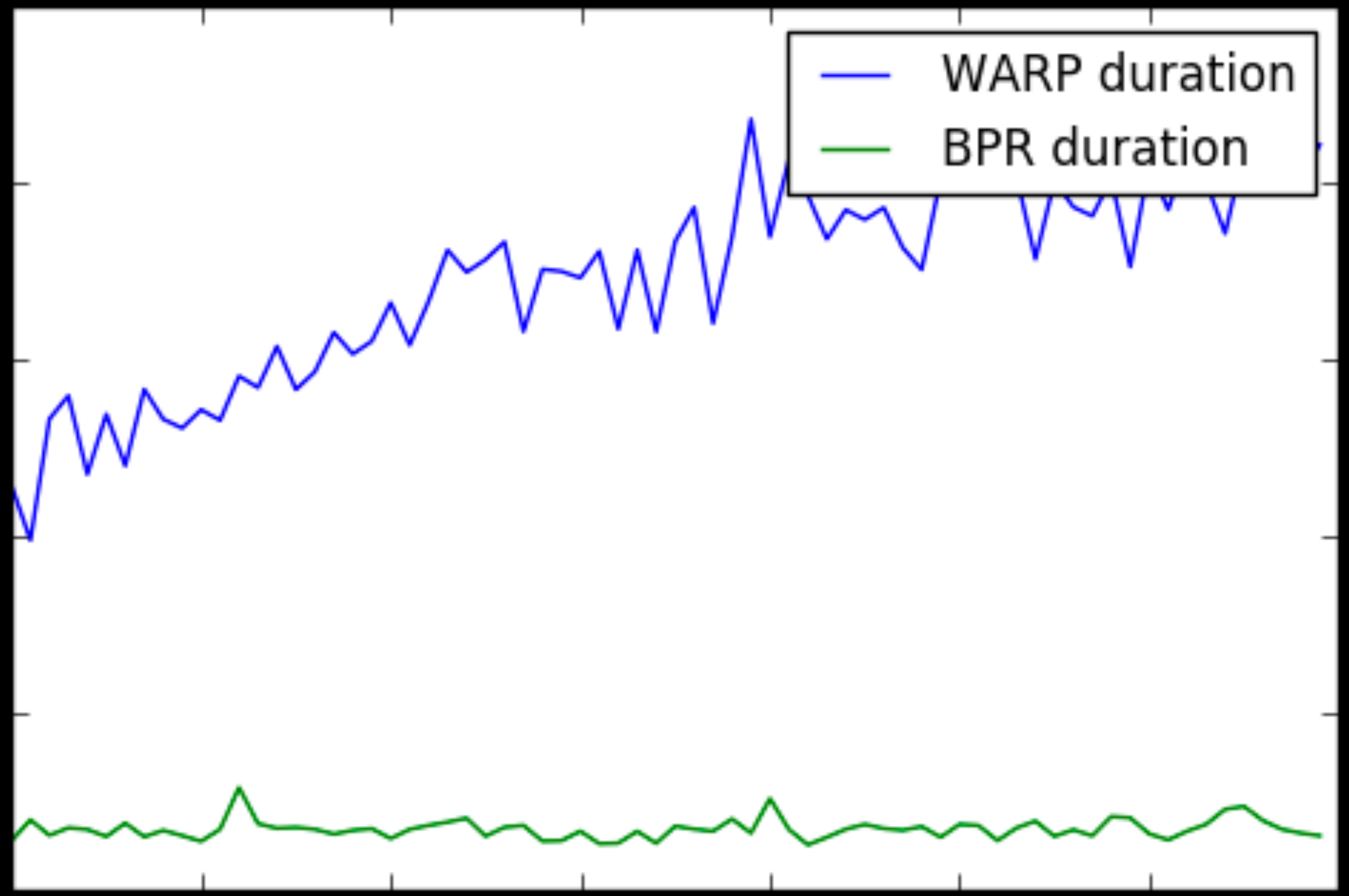
procedure LEARN MF-BPR($S, \beta, W, \mathbb{L}$)
  initialize Θ
  repeat
    draw (u, i, L) from $p(u, i, L)$
    draw N from $p(N|u, L)$
    draw j from $p(j|u, L, N)$
    update Θ with BPR update rule [8]
  until convergence
  return Θ
end procedure

$$[Like, ?, Dislike, ?, Dislike, Like, Like]$$

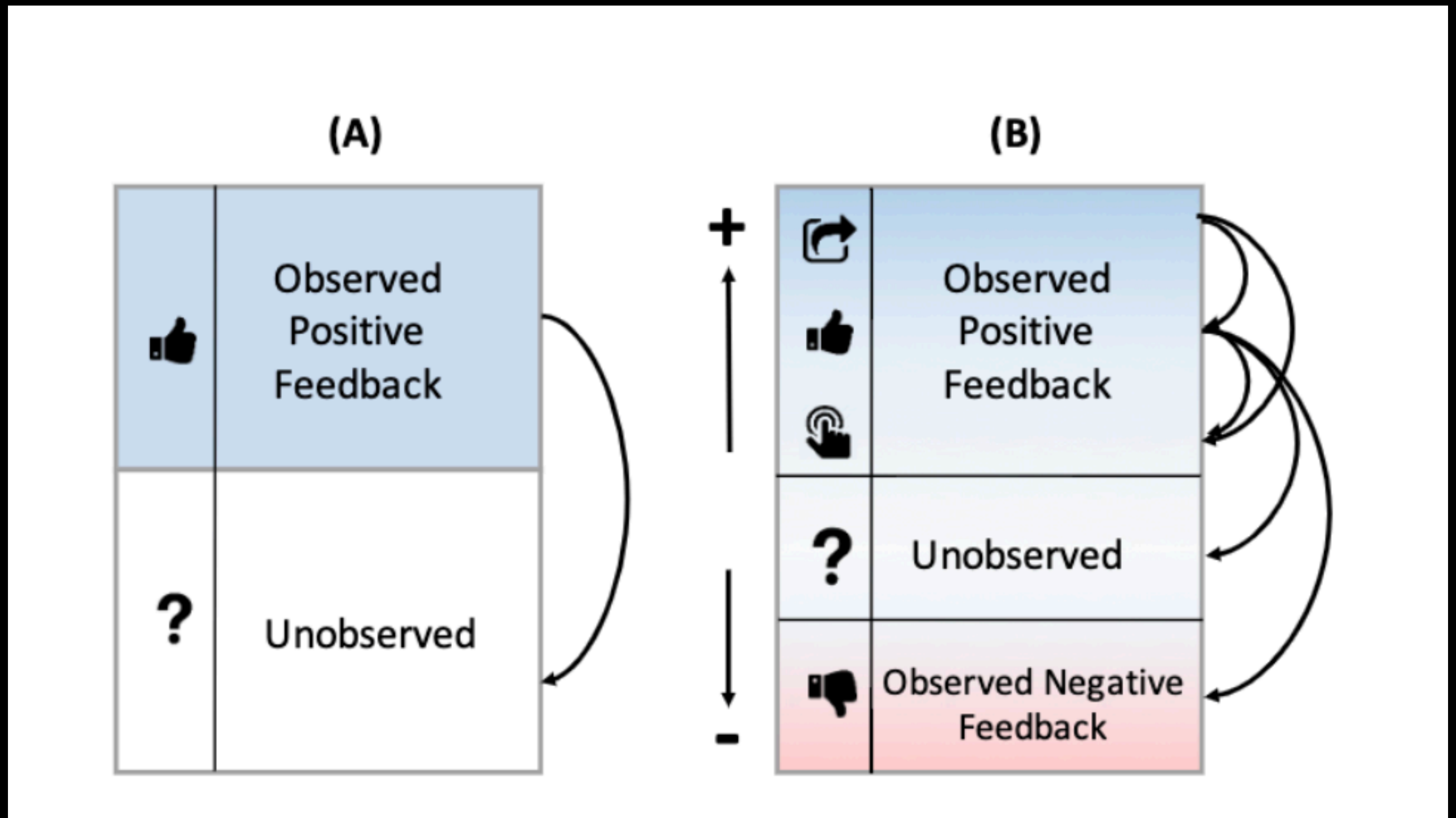
$$[Like, Like, Like, ?, ?, Dislike, Dislike]$$

WARP





MultiChannel WARP



$$[Like, Dislike, Share, Comment, Dislike, Like, Like]$$

$$[Like, Like, Like, Share, Comment, Dislike, Dislike]$$

How to prepare data

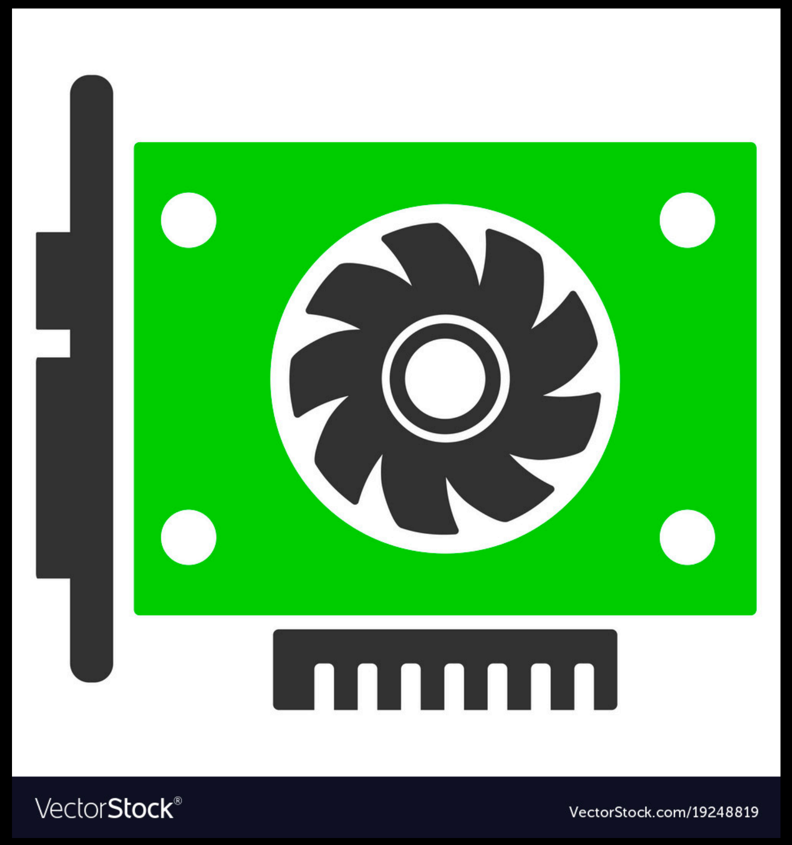
- Few user interactions -> bad recommendations for user | noise in data
- Few item interactions -> bad similars for item | noise in data
- Session analysis
- Detect strong negatives

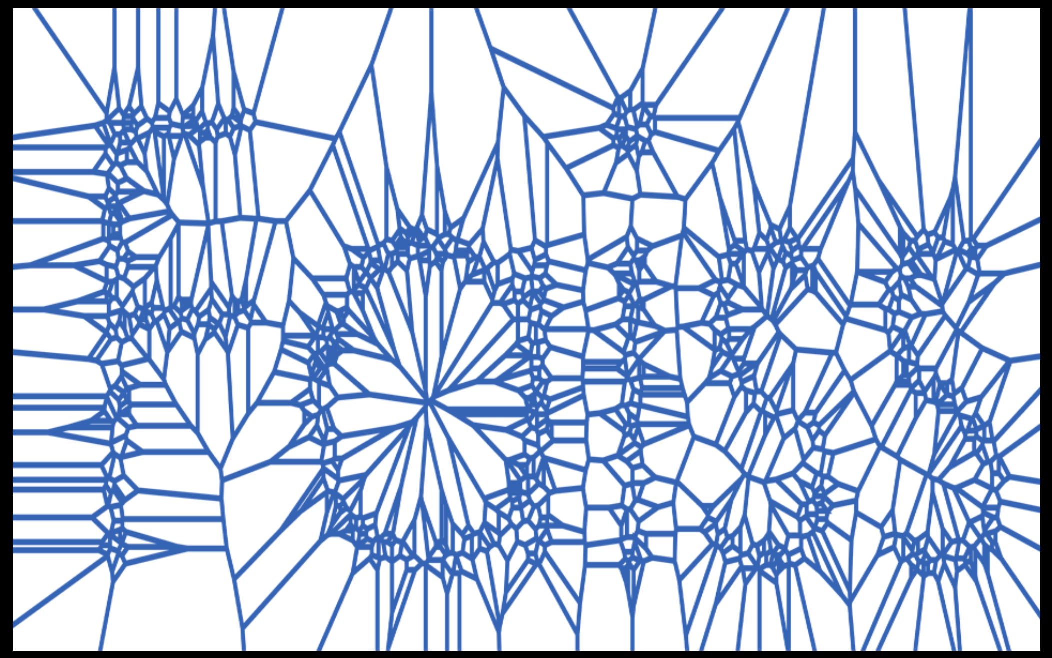
From .ipynb to production

- For each user construct recommendations in ipynb and send it to production and return it on each user's request

- Upload user's and item's embeddings to storage and scores each pair of user-item on user's request

- For each item construct similar items and send it to production and return it on each item's request





MF in business

- Mail.Ru | OK | VK
- Yandex
- Ivy
- Megogo
- etc

Problems

- Context features
- No Sessions
- No information about sequences
- Cold start

Conclusion

- MF - one of the easiest way to construct recommendations and similar users/items
- There is much more implicit feedback than explicit
- Recommendations is not about number approximation, it is about ranking (Use BPR or WARP)

Sources

- Distributed Design and Implementation of SVD++ Algorithm for E-commerce Personalized Recommender System. Communications in Computer and Information Science. **572**. Springer Singapore. pp. 30–44
- https://github.com/benfred/implicit
- https://github.com/lyst/lightfm
- Matrix and Tensor Factorization Techniques for Recommender Systems   ISBN 978-3-319-41357-0
- http://www.machinelearning.ru/wiki/images/9/95/Voron-ML-CF.pdf