

Beyond Adaptive Testing: Adaptive Item Recommendation with Process Data

Okan Bulut

Measurement, Evaluation, and Data Science

University of Alberta



bulut@ualberta.ca



www.okanbulut.com



[@drokanbulut](https://twitter.com/drokanbulut)

Outline

1. Need for stronger adaptability
2. Recommender systems
3. An application of adaptive item recommendation

From “Selection” to “Recommendation”

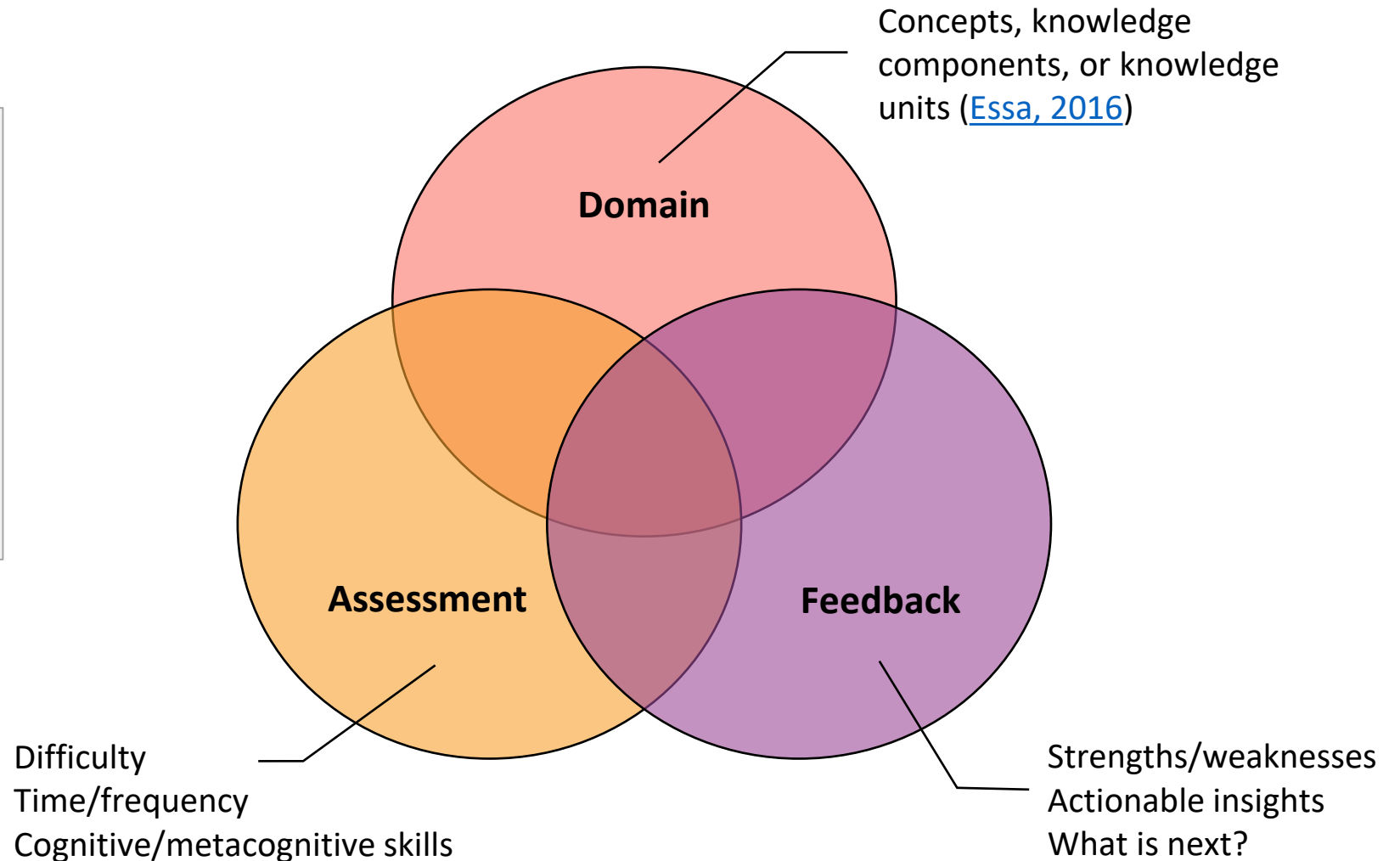
- In conventional computerized adaptive testing,
 - Select and administer the best item from the item bank
 - Update the examinee’s provisional ability estimate
 - Select the best item based on the updated ability
- “Limited adaptability” because CAT relies only on:
 - Item-level statistics (e.g., difficulty)
 - The examinee’s ability to answer the administered items
- Intelligently matching the best items to examinees
 - Individualized item recommendations based on item and examinee attributes

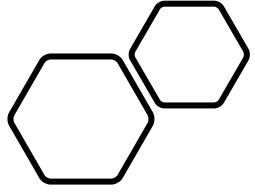


On the Road to Stronger Adaptability

“Adaptive” Variables

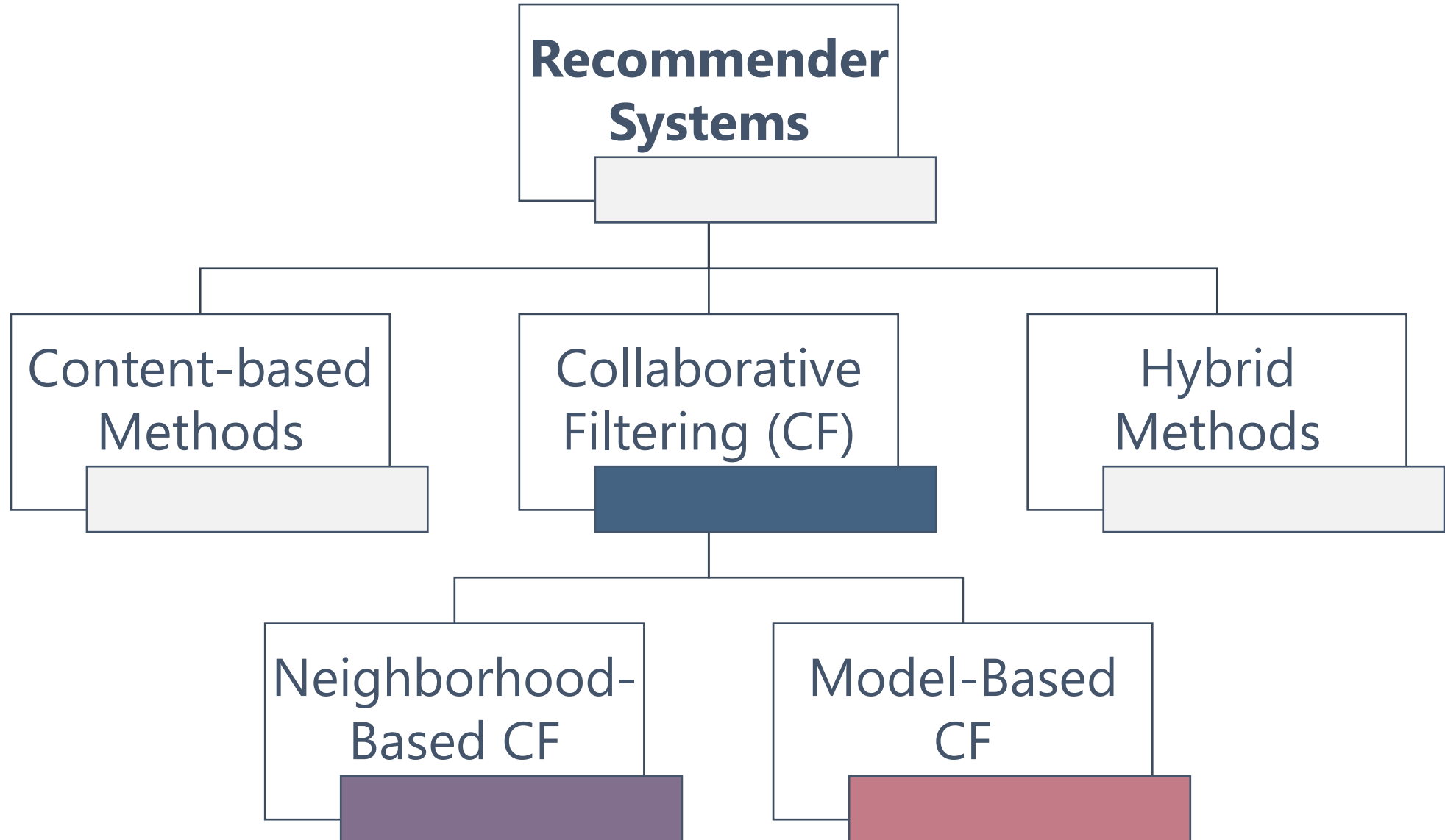
Cognitive learning styles
Preferences and interests
Learning progression
Demographic variables
([Triantafillou et al., 2007](#))

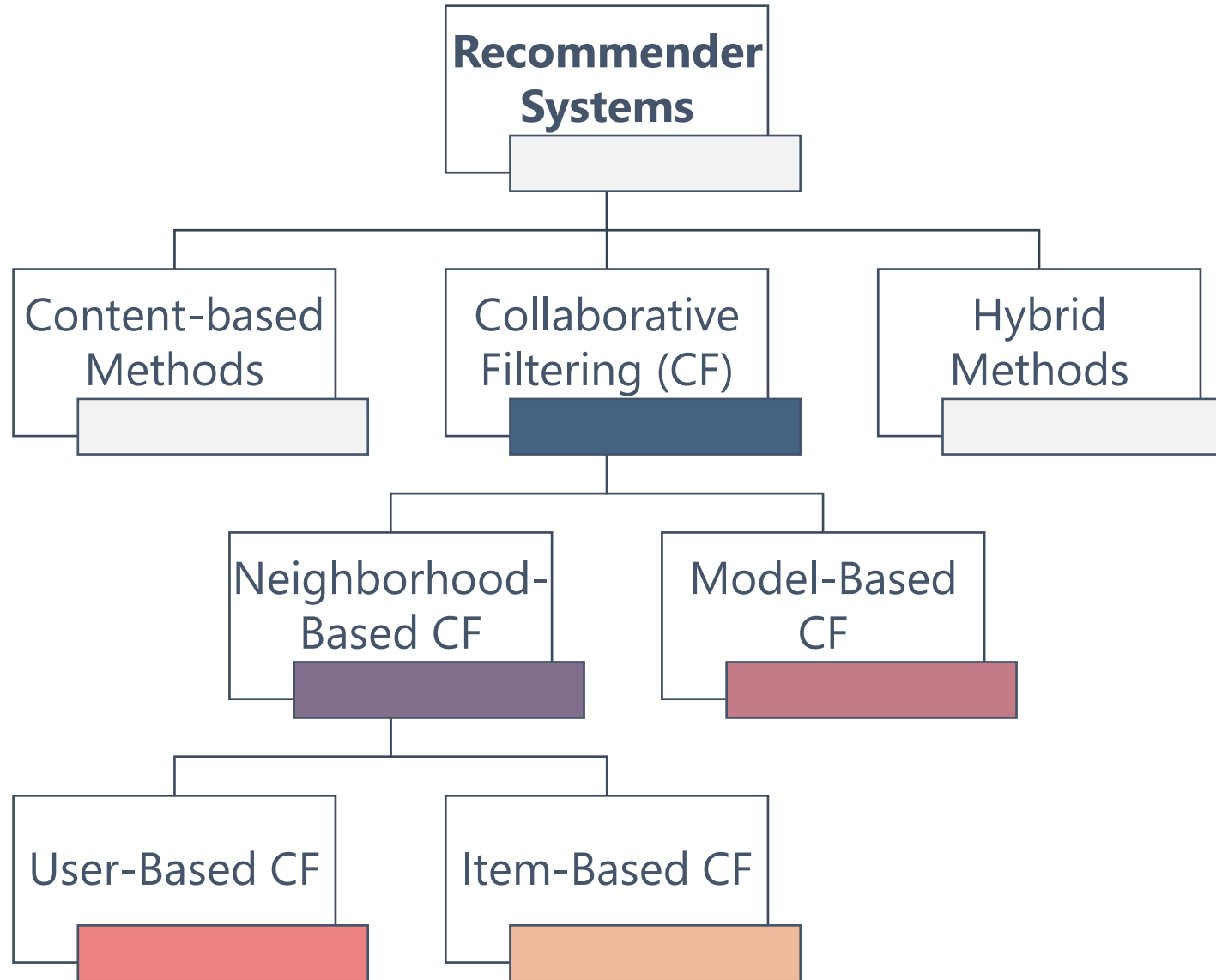




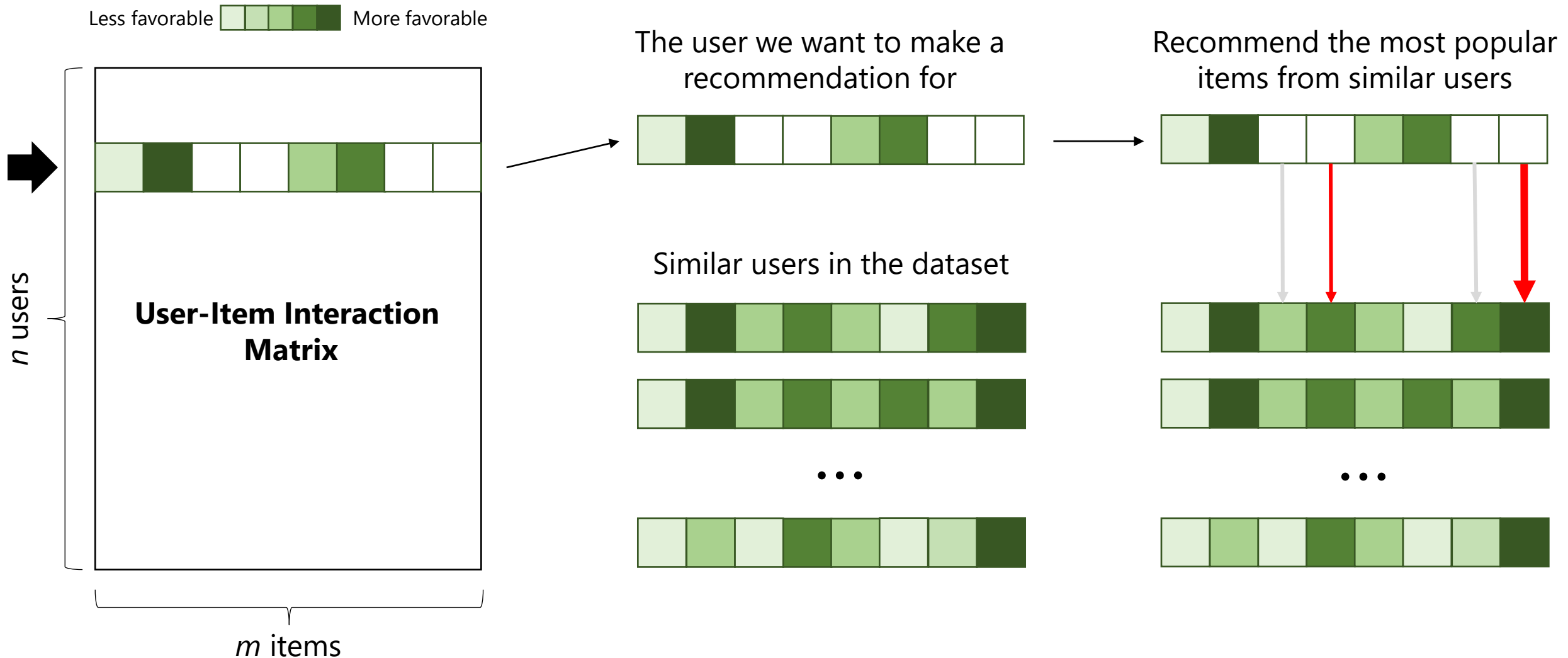
Recommender systems

"... personalized information agents that provide recommendations: suggestions for items likely to be of use to a user" ([Burke, 2007](#))





User-Based Collaborative Filtering



User-Based Collaborative Filtering

Step
1

$$\text{sim}(u_i, u_k) = \cos(u_i, u_k) = \frac{\sum_{j=1}^m r_{ij} r_{kj}}{\sqrt{\sum_{j=1}^m r_{ij}^2 \sum_{j=1}^m r_{kj}^2}}, \text{ or}$$

- **user** $u_i, i = 1, \dots, n$
- **item** $p_j, j = 1, \dots, m$
- **rating** r_{ij}

$$\text{sim}(u_i, u_k) = \text{cor}(u_i, u_k) = \frac{\sum_{j=1} (r_{ij} - \bar{r}_i)(r_{kj} - \bar{r}_k)}{\sqrt{\sum_{j=1} (r_{ij} - \bar{r}_i)^2 \sum_{j=1} (r_{kj} - \bar{r}_k)^2}},$$

Step
2

Perform k -nearest neighbors (KNN) to select the best neighbors of the target user (alternatively, use a similarity threshold)

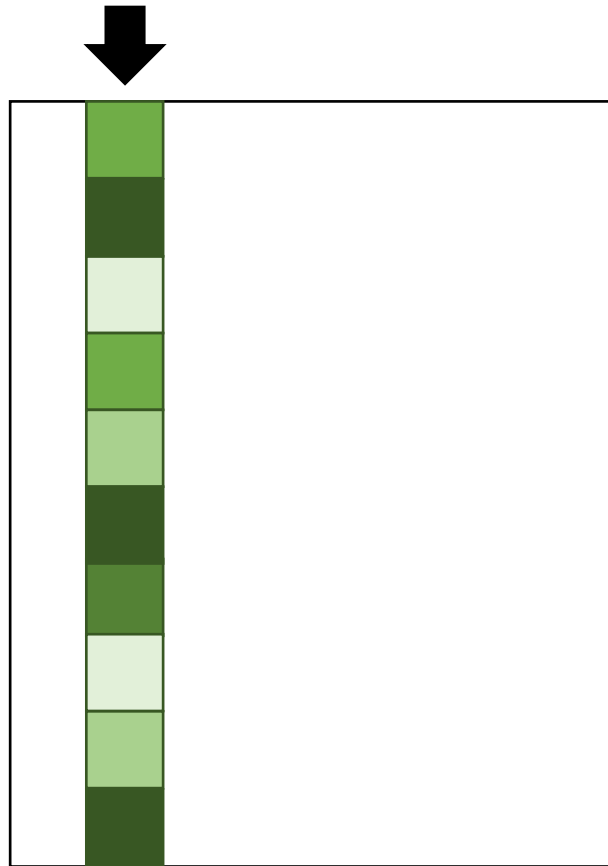
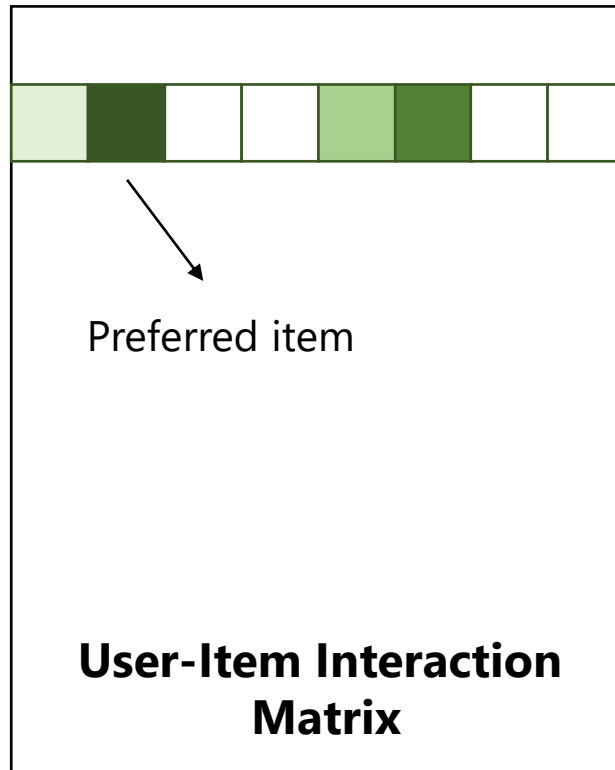
Step
3

Predict an unknown rating for the target user based on the best neighbors identified in Step 2 (i.e., weighted average of ratings from the best neighbors).

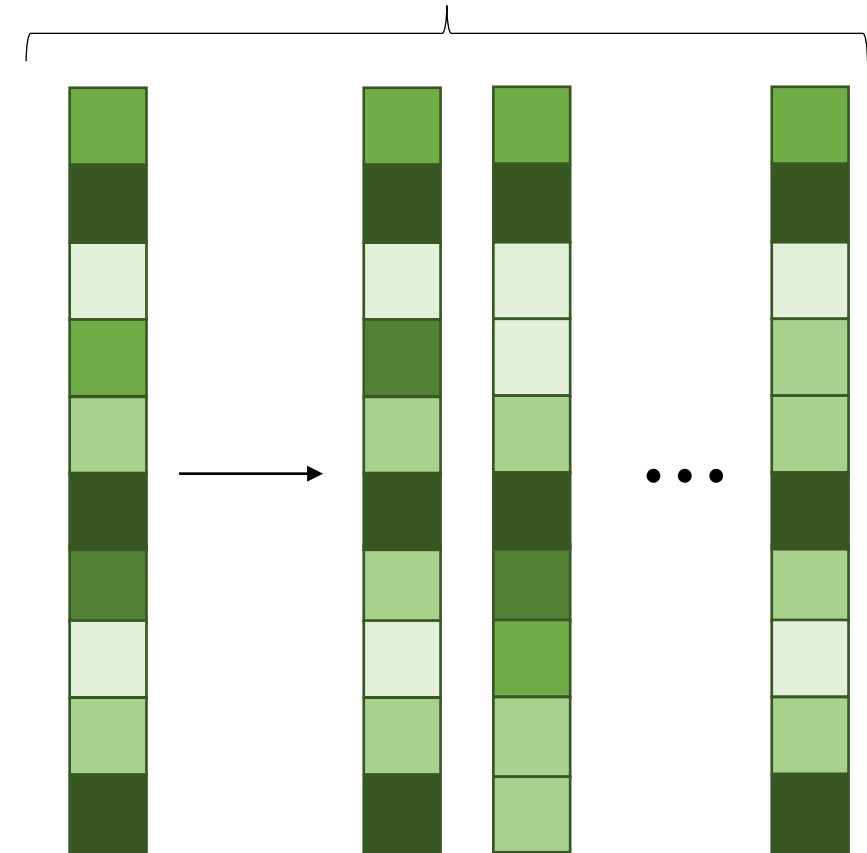
$$\hat{r}_{ij} = \frac{\sum_k \text{sim}(u_i, u_k) r_{kj}}{\# \text{ of ratings}} \quad \text{or} \quad \hat{r}_{ij} = \bar{r}_i + \frac{\sum_k \text{sim}(u_i, u_k) (r_{kj} - \bar{r}_k)}{\# \text{ of ratings}}$$

Item-Based Collaborative Filtering

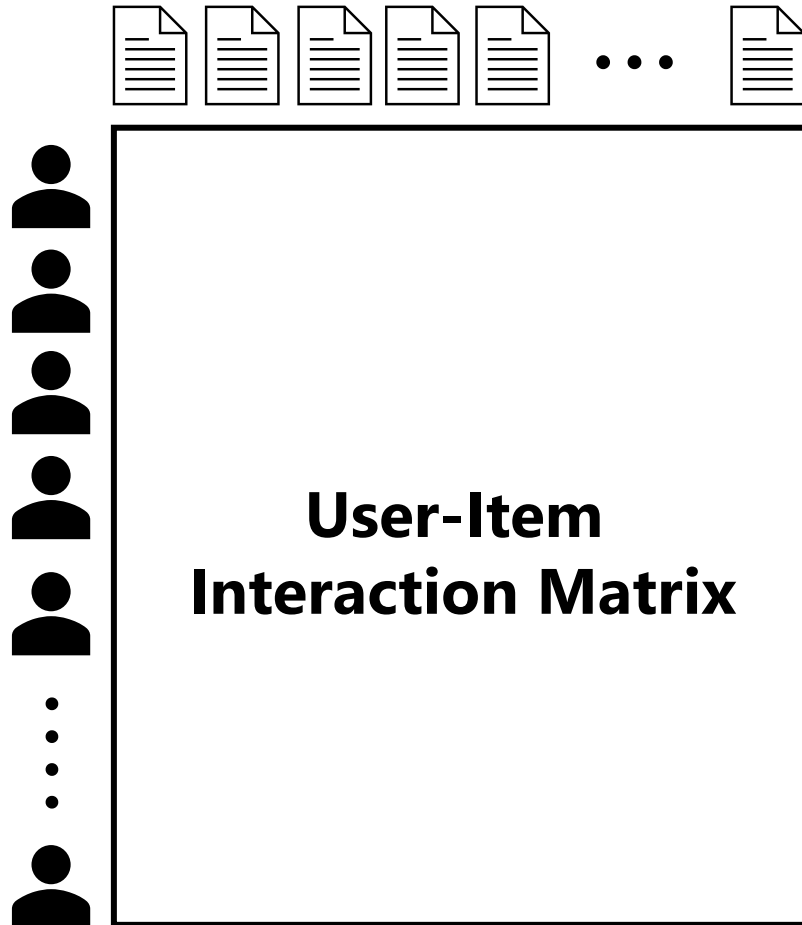
Less favorable  More favorable



Apply the KNN algorithm, find the most similar item(s), and recommend them



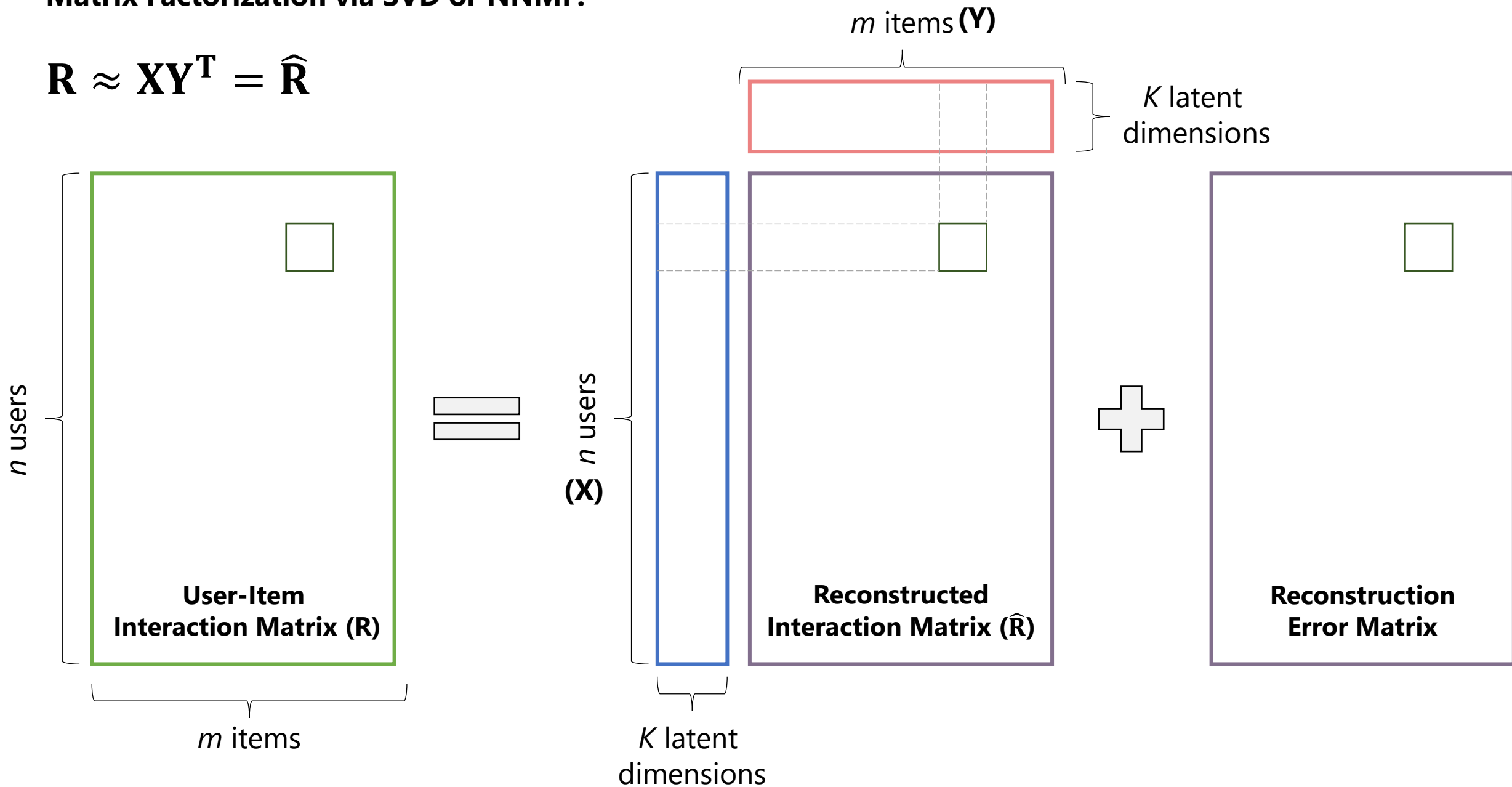
Model-Based Collaborative Filtering

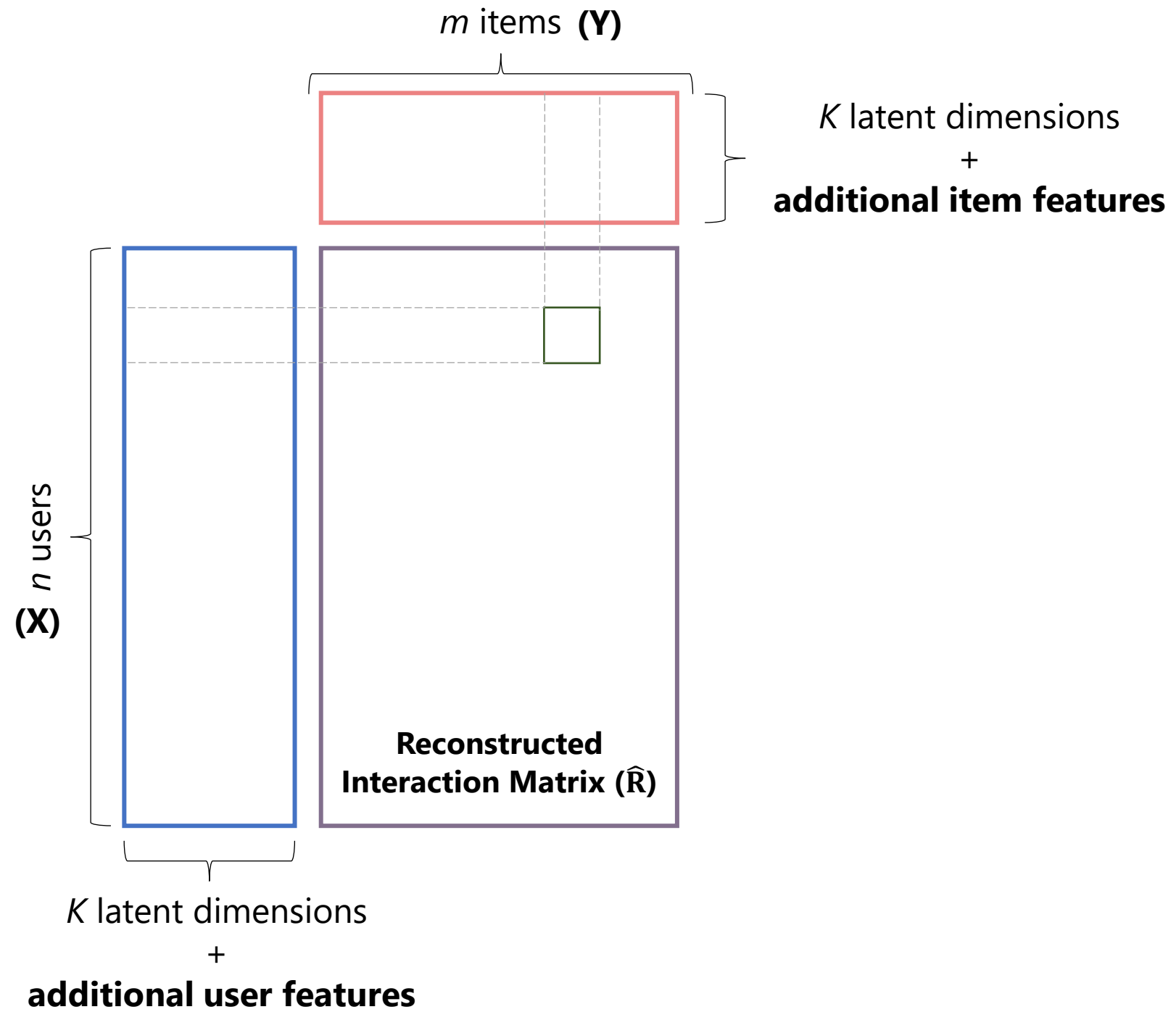


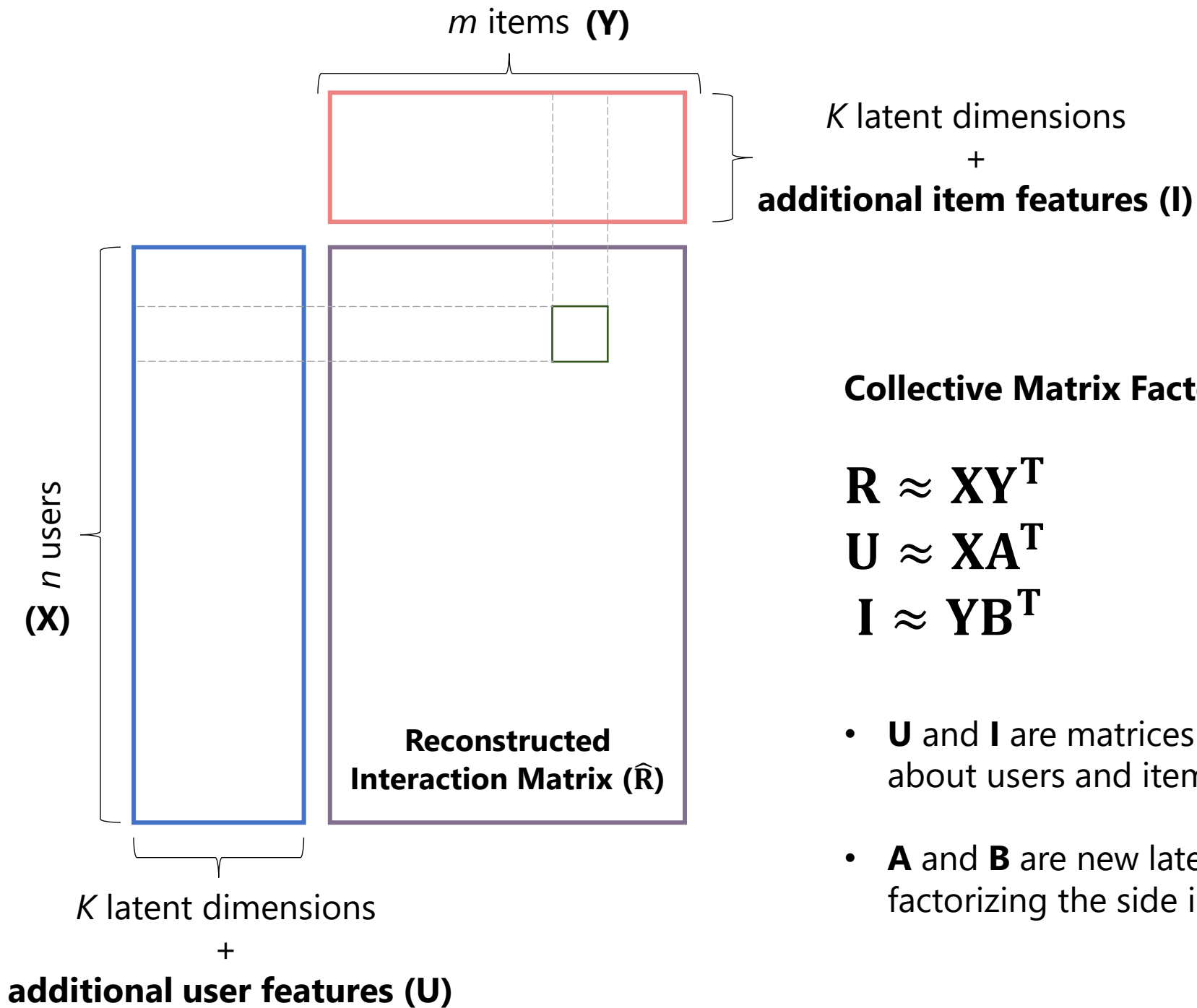
An underlying **generative** model that explains the user-item interactions.

Matrix Factorization via SVD or NNMF:

$$\mathbf{R} \approx \mathbf{X}\mathbf{Y}^T = \hat{\mathbf{R}}$$







Collective Matrix Factorization:

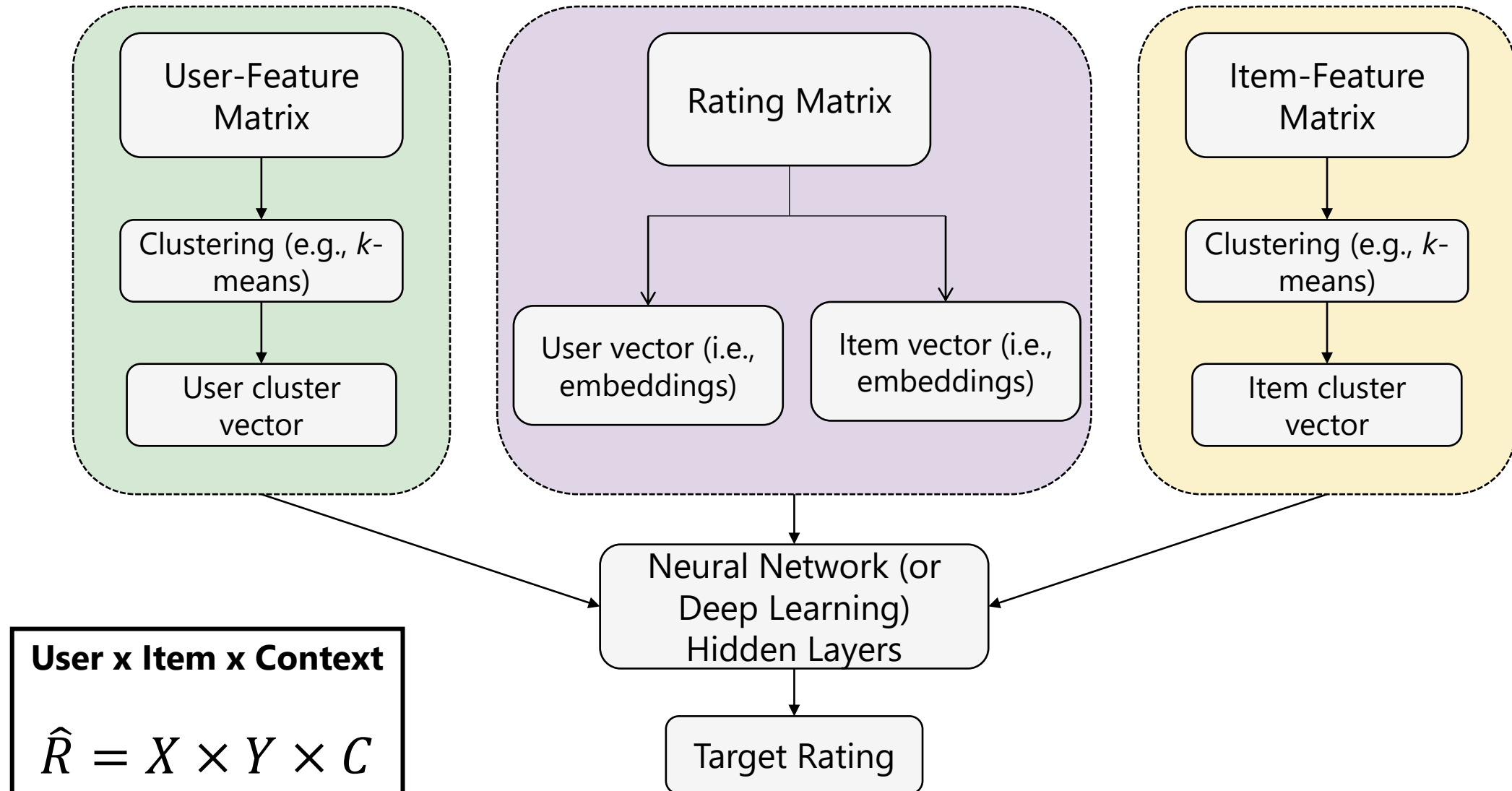
$$\mathbf{R} \approx \mathbf{X}\mathbf{Y}^T$$

$$\mathbf{U} \approx \mathbf{X}\mathbf{A}^T$$

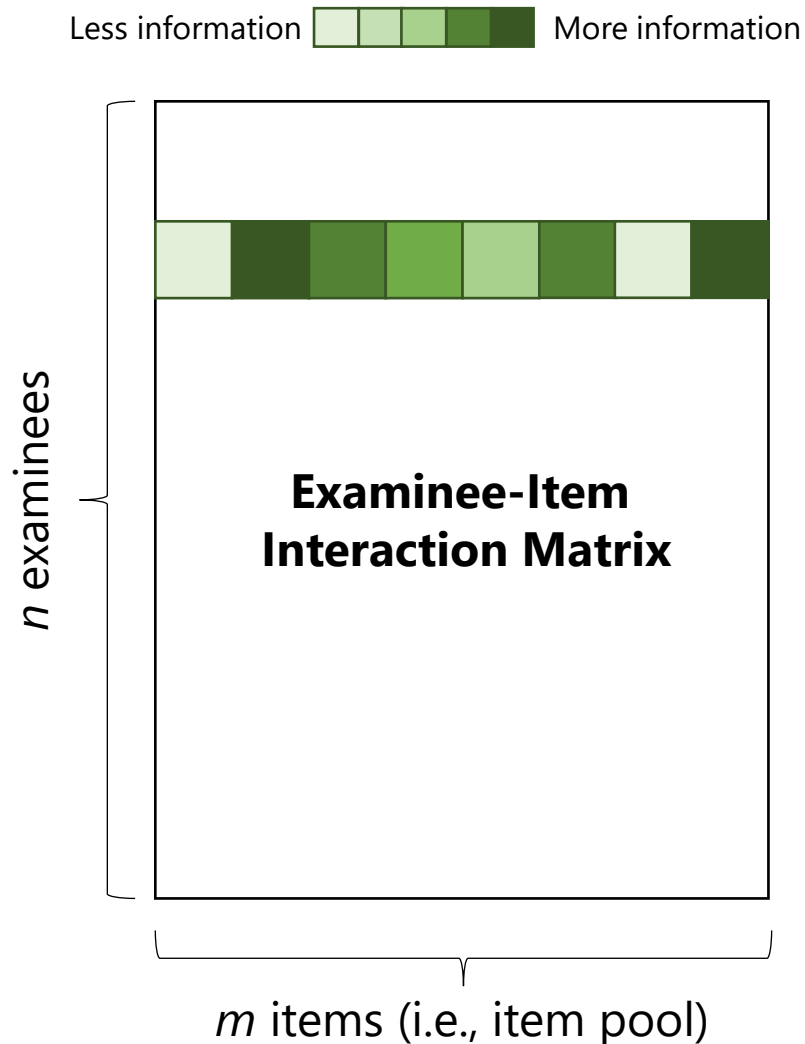
$$\mathbf{I} \approx \mathbf{Y}\mathbf{B}^T$$

- \mathbf{U} and \mathbf{I} are matrices representing side information about users and items, respectively.
- \mathbf{A} and \mathbf{B} are new latent factor matrices used for factorizing the side information matrices

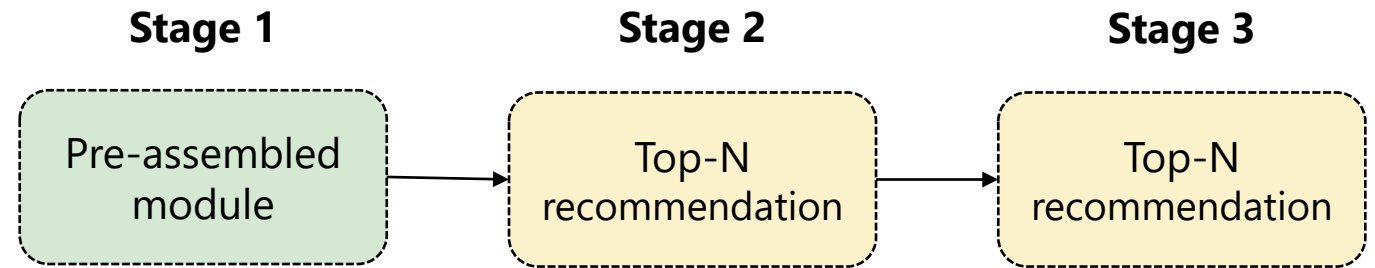
Hybrid Recommender Systems



Adaptive Item (or Task) Recommendation



Item selection with collaborative filtering in on-the-fly multistage adaptive testing ([Xiao, & Bulut, 2022](#))



On-the-fly module assembly using:

- User-based CF (UBCF)
- Item-based CF (IBCF)
- Maximum Fisher information (MFI)

Results:

- Measurement accuracy: UBCF > MFI
- Item bank utilization: IBCF > MFI

Enhancing Accuracy and Adaptability

- **Adaptive variables** are necessary to enhance the performance of adaptive item recommendation.
 - Cold-start problem
- Thanks to the rise of digital assessments, it is now easier to collect process data (e.g., response time, action sequences, and time-stamped events).
 - Distinguishing response processes and behavioral patterns (e.g., [He et al., 2021](#); [Ulitzsch et al., 2021](#))
 - Improving scoring/ability estimation (e.g., [Zhang et al., 2023](#))
- Examinee background characteristics also matter.
 - Background characteristics are correlated with digital problem solving (e.g., [Zhang et al., 2021](#))



This Study

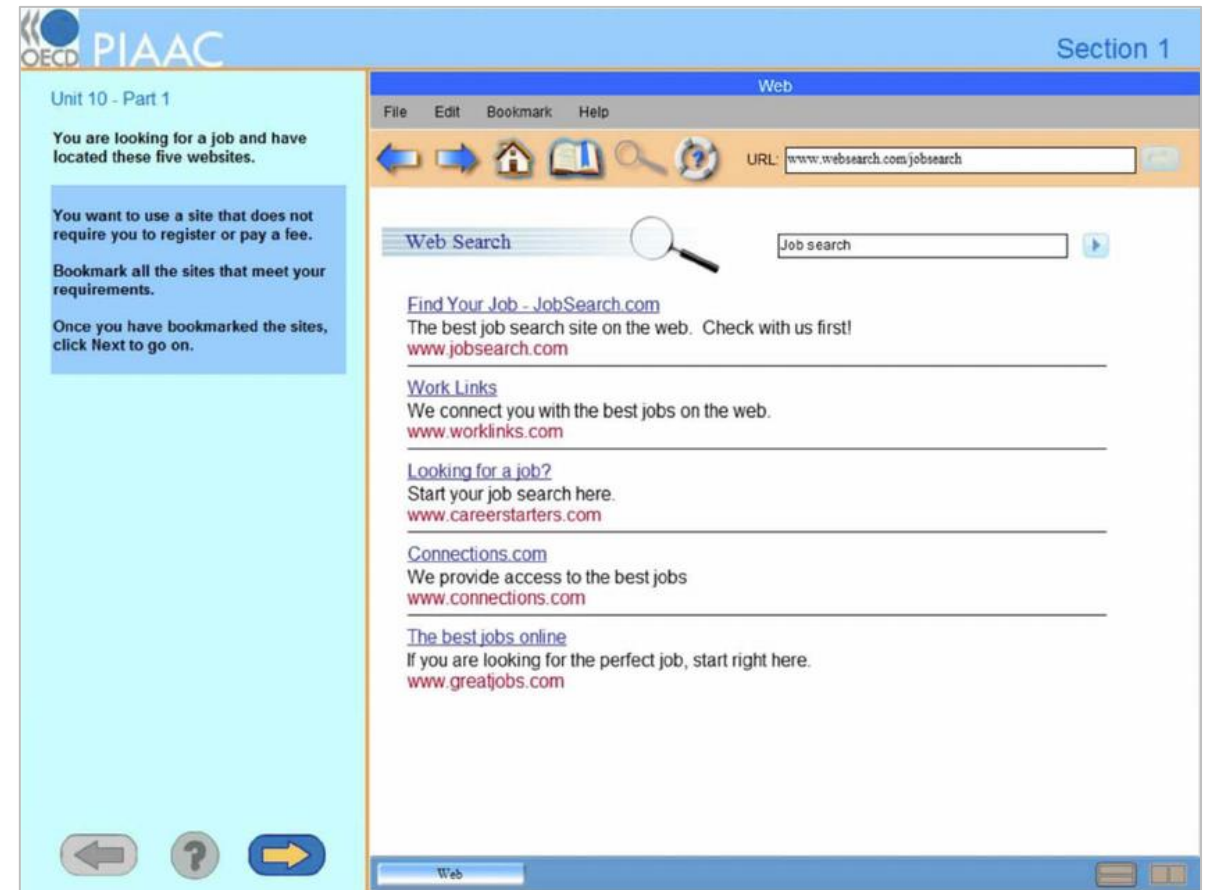
The goal of this study is to improve the quality of item recommendation by incorporating **adaptive variables** into the scoring process:

- Process data (generated on the fly *during* test administration)
- Background characteristics (available *before* test administration)

Simulation Design

Programme for the International Assessment of Adult Competencies (PIAAC)

- Problem-solving in technology-rich environments (PSTRE) tasks in PIAAC 2012
- 14 interactive tasks measuring higher-order skills (e.g., evaluating information and using it to solve a problem)



Simulation Design



- Design conditions:
 - 2000 examinees (training data = 1000, test data = 1000)
 - Theta was generated from a normal distribution, $N(0,1)$.
 - 60 hypothetical problem-solving tasks with up to 26 possible actions (each denoted with an English letter)
 - Actions were generated from a Markov model where the transition probabilities depended on theta.
 - Dichotomous response data
 - Responses were generated for each task based on top 10 n-grams that were most correlated with theta.
 - The 2PL IRT model was used for item calibration.
 - Examinee background variables
 - One binary and two continuous variables were generated (correlated with theta; $r = .25 - .35$).

An Example of Generated Sequence Data

- Examinee 1 (14 steps):
 - A-R-O-O-L-W-R-J-Q-R-Q-U-Z
- Examinee 3 (8 steps):
 - A-O-E-R-J-V-L-Z
- Examinee 5 (22 steps):
 - A-J-R-D-F-T-B-K-N-P-B-Y-G-N-H-P-J-H-X-X-M-Z
- On average, 25-26 actions per item (min = 2, max = 190)

Feature Extraction from Process Data (1)

- Multidimensional scaling (MDS; [Tang et al., 2020](#)) was used to extract the latent features from the process data of each task.
- MDS maps action sequences into the K -dimensional Euclidean space based on **dissimilarity** between the action sequences.
 - Ordering-based Sequence Similarity (OSS; [Gomez-Alonso & Valls, 2008](#))
 - Assume two response processes: $a_i = (a_{i1}, \dots, a_{in_i})$ and $a_j = (a_{j1}, \dots, a_{jn_j})$ with different lengths (n_i and n_j) $\rightarrow d_{ij} = d(a_i, a_j)$

$$d(a_i, a_j) = \frac{f(a_i, a_j) + g(a_i, a_j)}{n_i + n_j}$$

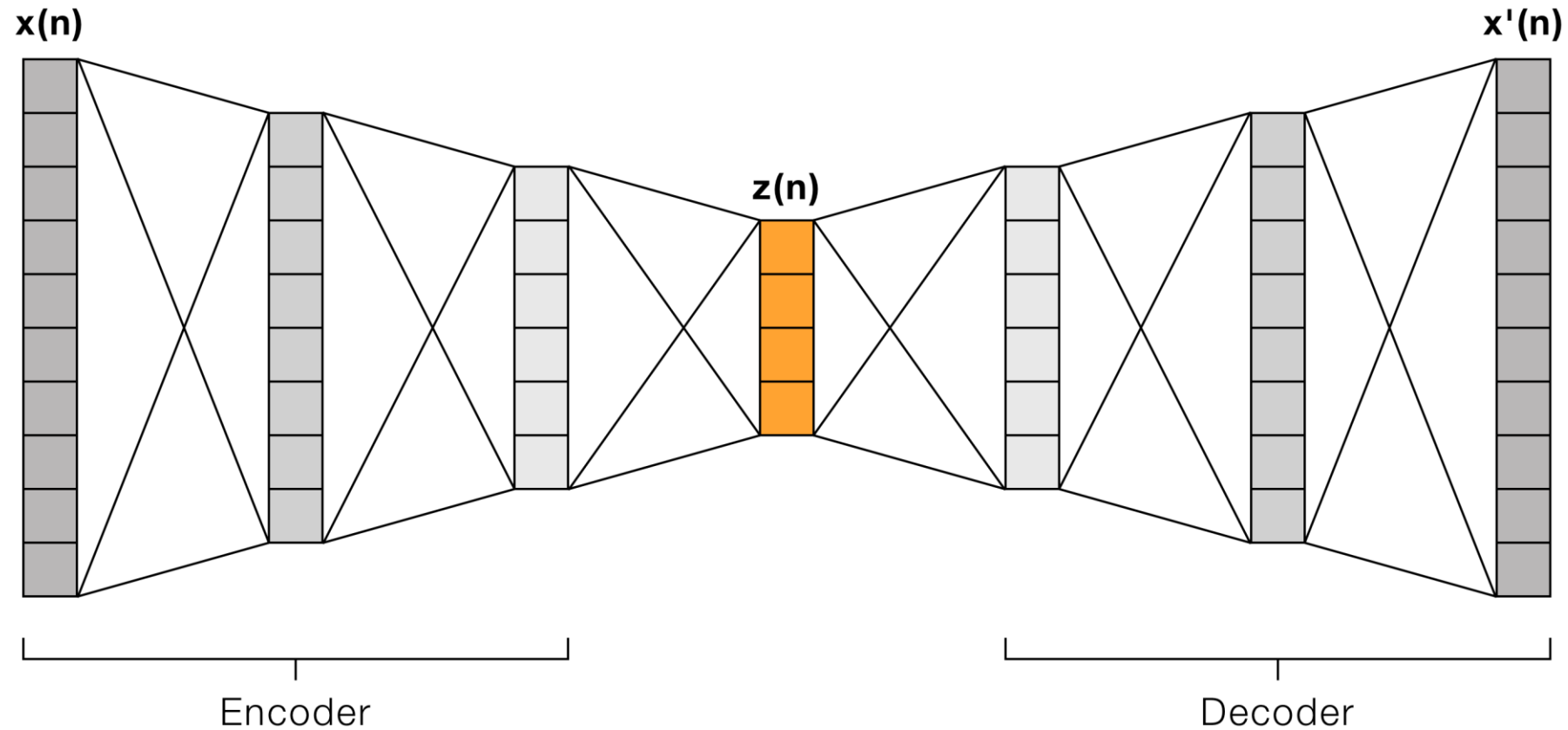


Common actions



Uncommon actions

Alternative: Sequence-to-Sequence Autoencoders



Tang, X., Wang, Z., Liu, J., & Ying, Z. (2020). [An exploratory analysis of the latent structure of process data via action sequence autoencoders](#). *British Journal of Mathematical and Statistical Psychology*, 74(1), 1-33.

Feature Extraction from Process Data (2)

1. For each item, find the distance matrix $\mathbf{D} = (d_{ij})_{n \times n}$ based on the pairwise dissimilarities.
2. Obtain K raw features by applying MDS to each \mathbf{D} matrix (by minimizing the squared distance between d_{ij} and Euclidean distance between i and j in the K -dimensional Euclidean space).
 - $K = 20$ raw features (i.e., problem-solving processes) were extracted for each hypothetical task.
3. Transform K raw features to K principal features via principal component analysis (for increased interpretability).
 - $K = 20$ principal features were extracted for each hypothetical task.

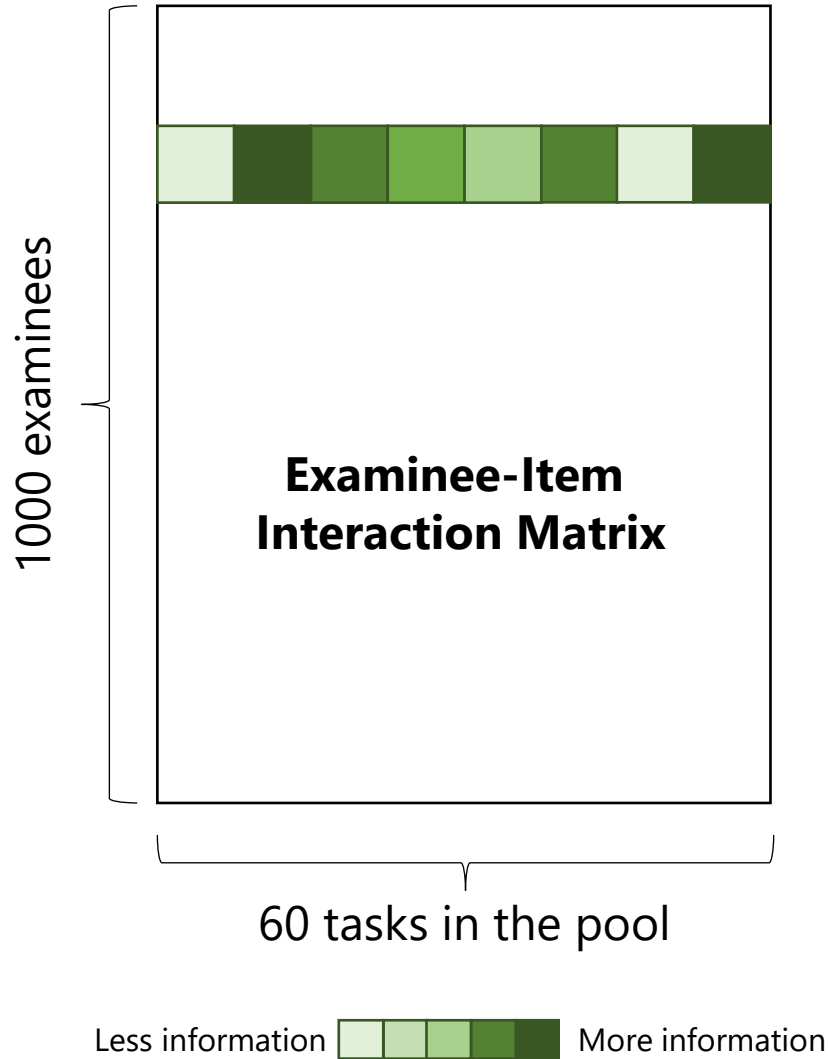
Process-Based Scoring ([Zhang et al., 2023](#))

Goal: To *refine* the latent trait estimates with the additional information from the problem-solving process

1. For each item ($j = 1, \dots, J$), estimate the ability based on the final response (Y_j) to item j only ($\hat{\theta}_{Y_j}$).
2. Also, estimate $\hat{\theta}_Y$ based on final responses to all items administered so far.
3. For each item, regress $\hat{\theta}_{Y_j}$ on all process features and examinee background variables (\mathbf{X}_{-j}), except for those on item j (via L2 regularization; ridge regression). The outcome is denoted as $T_{\mathbf{X}_{-j}}$.
4. Regress $\hat{\theta}_Y$ on $T_{\mathbf{X}_{-j}}$ and Y_j to obtain $\hat{\theta}_{\mathbf{X}_{-j}}$ (via ordinary least squares).
5. Compute the final process-incorporated ability by finding the average across all (J) items:

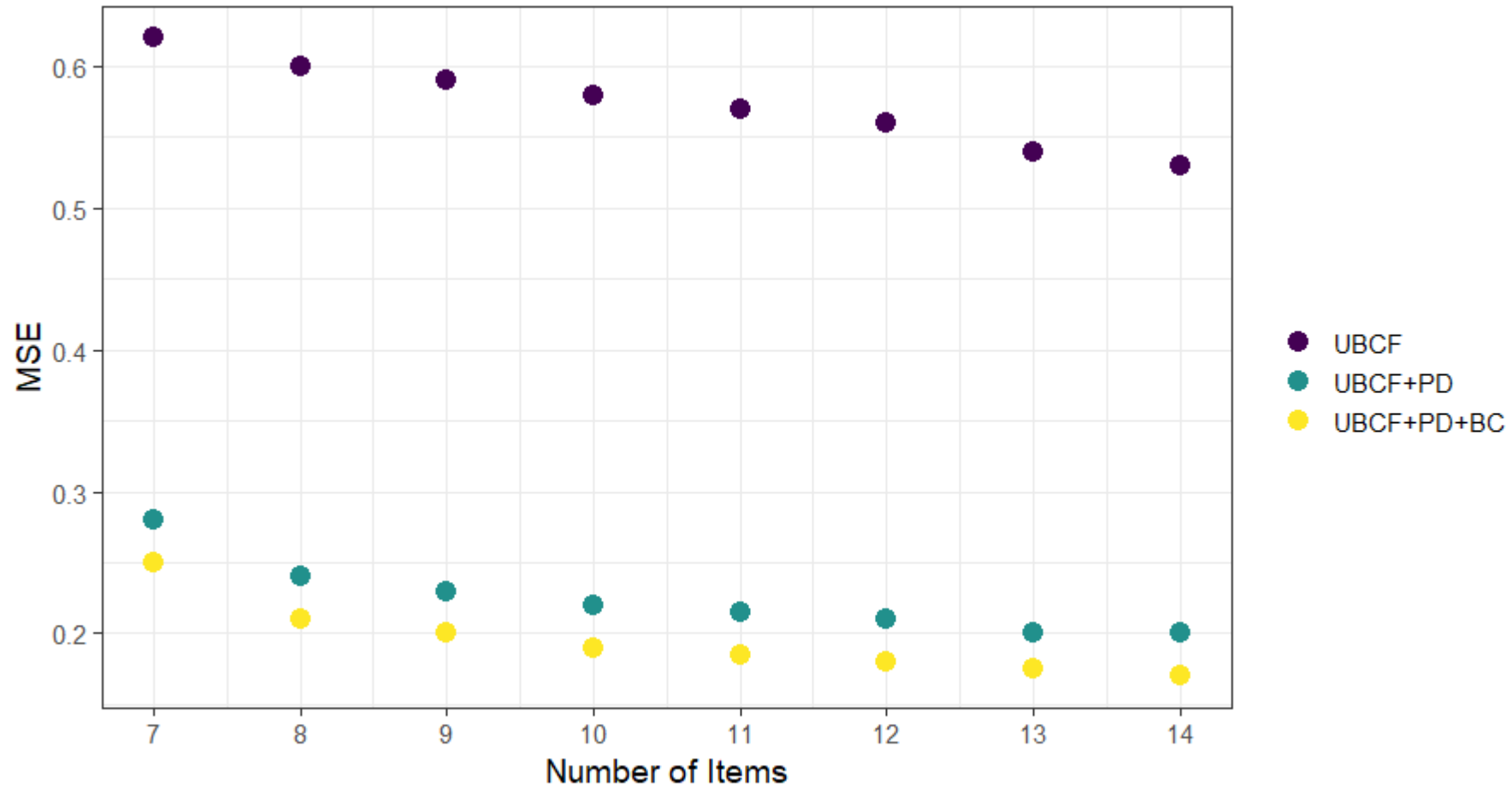
$$\hat{\theta}_X = \frac{1}{J} \sum_{j=1}^J \hat{\theta}_{\mathbf{X}_{-j}}$$

Training Dataset



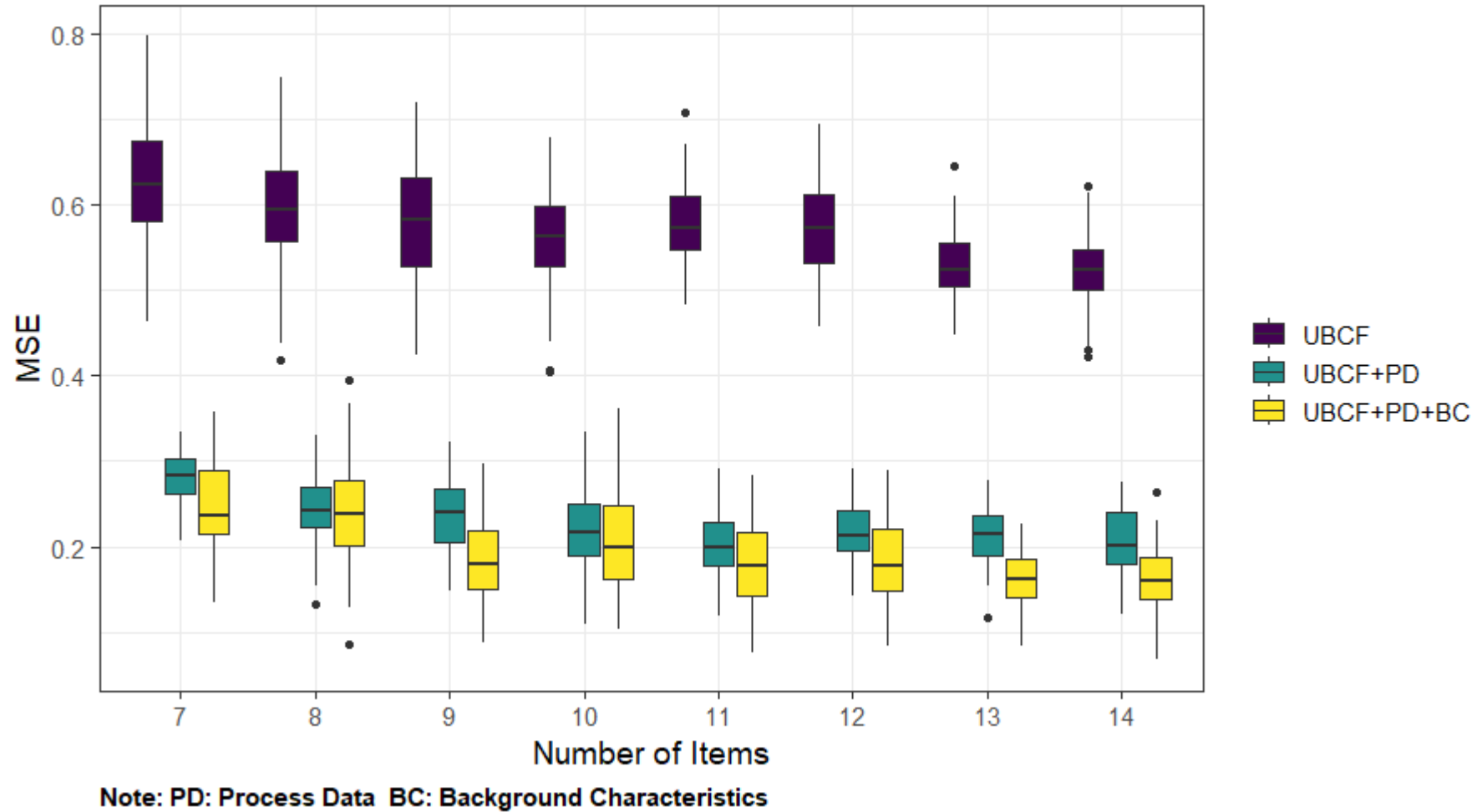
- The training dataset consisted of the Fisher information values calculated across 60 tasks and 1000 examinees.
- Adaptive item recommendation with:
 - User-based CF (UBCF) based on response-based scoring
 - UBCF based on process-based scoring
- **Procedure:**
 1. Administer a randomly selected task to the examinee
 2. Estimate ability using the item response (with or without process data and background variables)
 3. Identify top 5 items from the training data and recommend one of them (randomly) to the examinee
 4. Repeat Steps #2 and #3 until the maximum test length (from 7 to 14 tasks) is reached (50 iterations).

Results – Mean Squared Error (MSE)

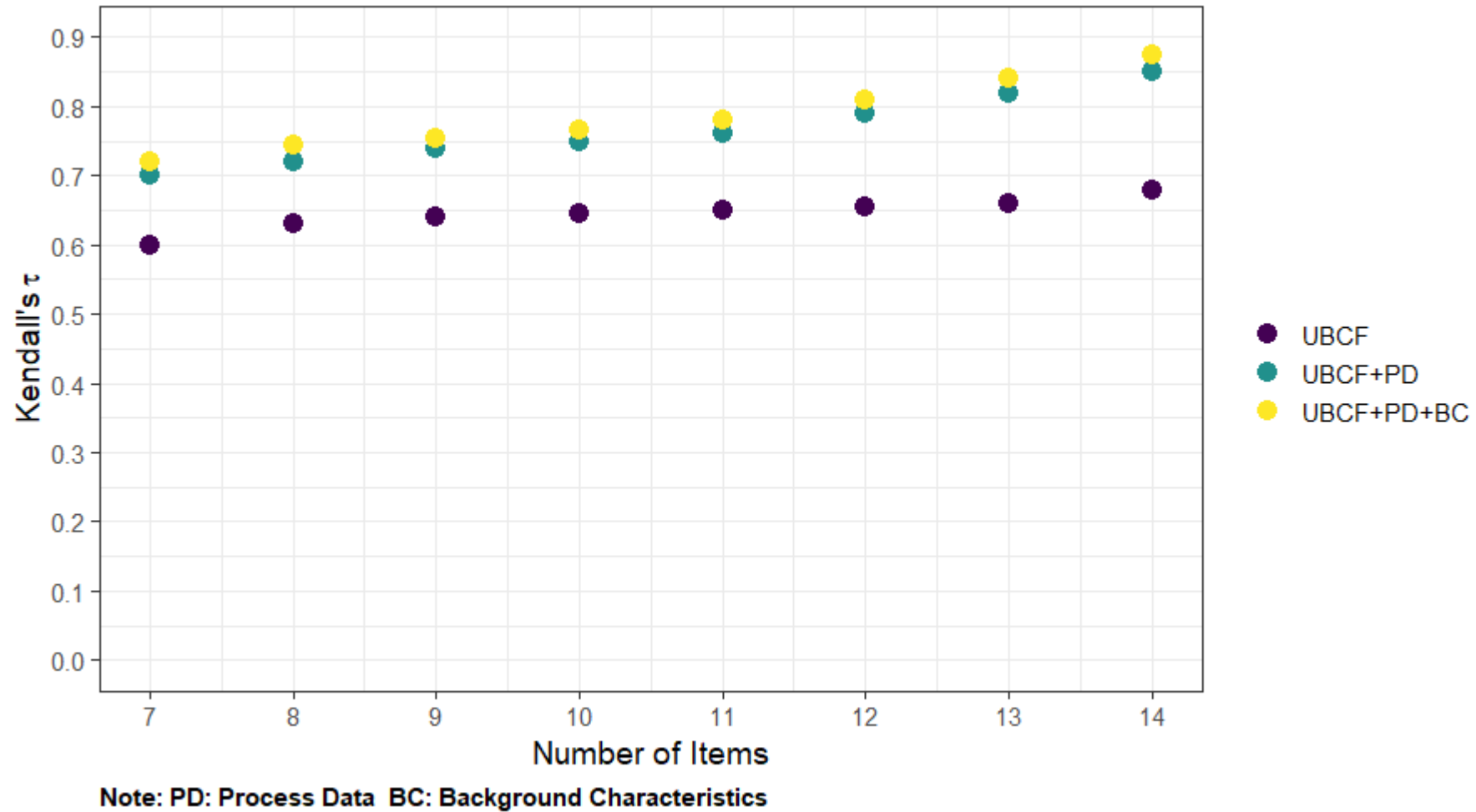


Note: PD: Process Data BC: Background Characteristics

Results – MSE Distribution



Results – Kendall's τ



Results – Item Bank Utilization

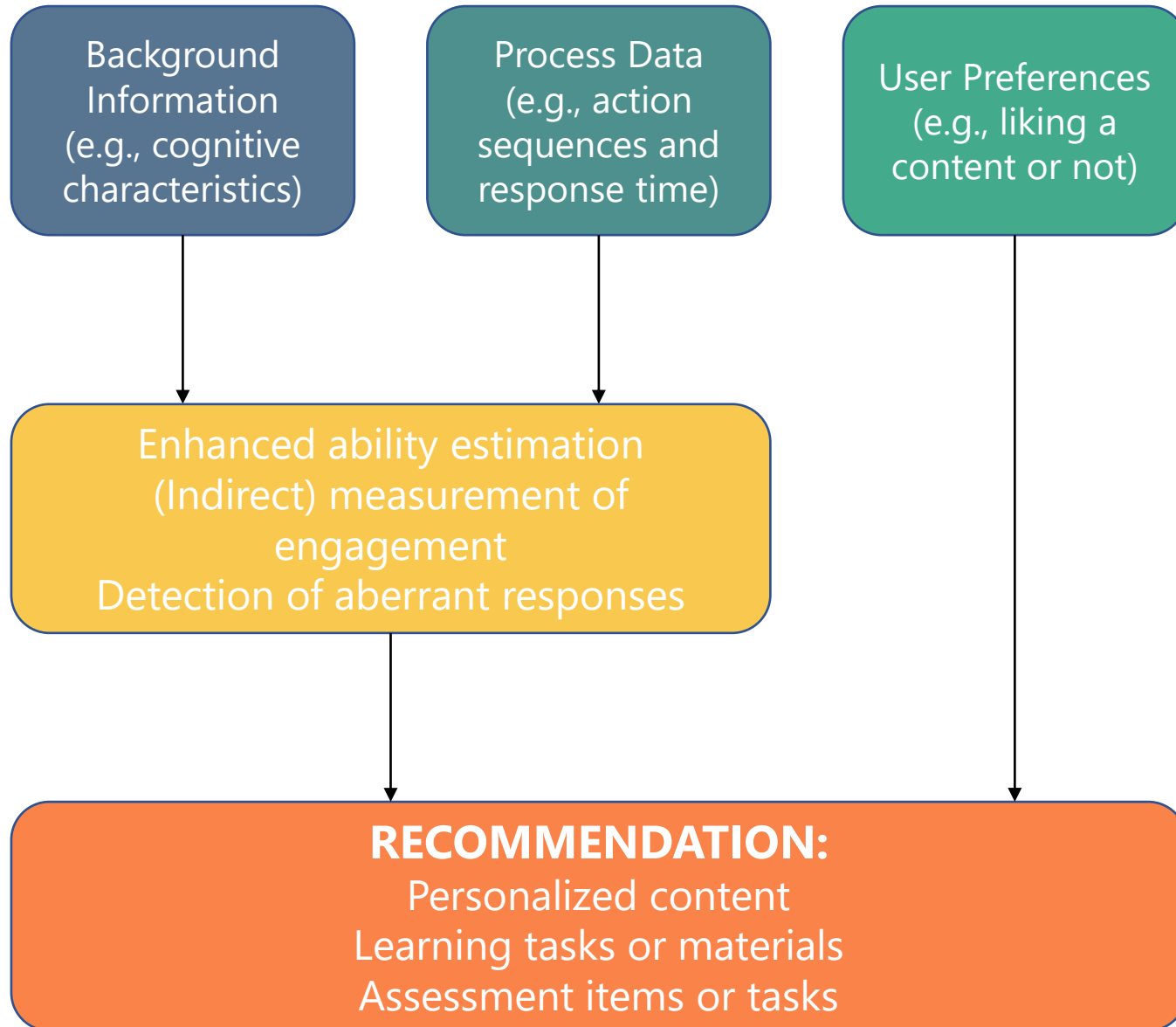
- All three methods yielded acceptable item bank utilization rates
 - More than 80% of the tasks in the pool were used.
- The item bank utilization rate was slightly higher for UBCF without process data or background characteristics.
 - This finding was mainly due to recommending similar items repeatedly to examinees who have similar ability as well as problem-solving sequences and background characteristics.
- With item exposure control, the item bank utilization could improve at the expense of measurement accuracy.

Concluding Remarks

- With the rise of digital assessments at all levels of education, it is now possible to obtain auxiliary information on examinees' response processes.
- Interactive tasks yield action sequences that include valuable information beyond dichotomous or polytomous item scores.
- Recommender systems can incorporate adaptive variables (e.g., process data and examinee background characteristics) to enhance the item recommendation process.
- Recommender systems with process data can also be applied to other settings, such as adaptive learning systems and personalized scheduling for adaptive tests ([Bulut, Shin, & Cormier, 2022](#); [Shin & Bulut, 2022](#); [Bulut, Cormier, & Shin, 2020](#)).

Adaptive Item Recommendation





DIGITAL TWIN

"A **digital twin** is a digital replica of a physical entity, and it is created by combining pieces of data from various sources."

[Furini et al. \(2022\)](#)

Future Research

- This study focused on “enhancing” ability estimation based on background variables and action sequences extracted from problem-solving tasks.
- Incorporating “item-related” process variables into the recommender system
 - Hand-crafted or data-driven item features
 - E.g., Problem-solving complexity/intensity of each task/item
- Process data can be directly utilized in making item recommendations:
 - Recurrent and convolutional neural models are particularly suitable for the sequential nature of process or log data ([Tang & Wang, 2018](#)).
 - If multimodal data (e.g., images, text, sound, etc.) about users/items are available, deep neural networks (DNNs) can be a better option for taking advantage of joint (end-to-end) representation learning ([Zheng, Noroozi, & Yu, 2017](#))

Thank You!

For questions/comments:

bulut@ualberta.ca

To download the slides: <https://bit.ly/ProcessDataApril28>

