From Adaptive Testing to Personalized Adaptive Testing: Applications of Machine Learning Algorithms

Okan Bulut
Measurement, Evaluation, and Data Science
University of Alberta
Outline

1. Beyond adaptive testing

2. Recommender systems (RS)

3. Two RS applications in adaptive testing
Digital Twins and Artificial Intelligence as Pillars of Personalized Learning Models

MODERN EDUCATIONAL SYSTEMS have not really evolved enough to meet the needs of modern students.¹ No wonder, the percentage of dropouts from university studies is quite high (44% in the U.S. and 10% in Europe)². The university student profile has changed over the years. While yesterday’s students were mainly full-time, today’s students face challenges such as work commitments, family obligations, financial constraints, physical impairments, and learning models that do not adequately engage students or help them understand core concepts.³ One might think that this issue concerns only those

¹ The times have come to revolutionize current educational systems, which are too rigid and cannot adequately support students who have work commitments, family obligations, physical impairments, and other barriers.
² AI and digital twin technologies are helping to transform cities into smarter societies, revolutionizing industries, and improving health services.
³ Digital twins and AI can be used to build personalized, inclusive, and accessible learning models. These models will not only provide mental, natural, and economic benefits, but they will also be able to detect and prevent diseases by the United Nations General Assembly.
Digital Twins and Artificial Intelligence as Pillars of Personalized Learning Models

Modern educational systems have not really evolved.
“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)*
STUDENT

• Academic background
• Study habits
• Subject preferences
• Cognitive characteristics
• Learning behaviors
• Digital educational material consumption

DIGITAL TWIN

• Digital student records
• Online learning activities
• Digital learning behaviors
• Data from digital assessments
• Learner knowledge space
• Interactions with learning materials
On the Road to Adaptive Learning Systems

"Adaptive" Variables
- Cognitive learning styles
- Preferences and interests
- Learning progression
- Demographic variables

(Triantafillou et al., 2007)

Concepts, knowledge components, or knowledge units (Essa, 2016)

Cognitive learning styles
Preferences and interests
Learning progression
Demographic variables

(Triantafillou et al., 2007)
<table>
<thead>
<tr>
<th>Layer</th>
<th>Challenges</th>
<th>Approaches</th>
</tr>
</thead>
<tbody>
<tr>
<td>Presentation</td>
<td>8. Provide a face-to-face experience when simultaneously teaching on-site and online students</td>
<td>Universal Design</td>
</tr>
<tr>
<td></td>
<td>7. Support different user types</td>
<td>Video Wall Interface</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Mobile App</td>
</tr>
<tr>
<td>Data</td>
<td>6. Design learning recommender systems</td>
<td>Artifical Intelligence</td>
</tr>
<tr>
<td></td>
<td>5. Create the student's digital twin</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4. Produce enriched content for on-demand use</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3. Define an engaging video lecture format</td>
<td></td>
</tr>
<tr>
<td>Infrastructure</td>
<td>2. Monitor students within university spaces</td>
<td>Off-the-Shelf Hardware</td>
</tr>
<tr>
<td></td>
<td>1. Make teachers comfortable when teaching</td>
<td></td>
</tr>
</tbody>
</table>

**Required Expertise**
- Computer Science
- Cognitive Psychology
- Computer Engineering
- Data Analysis
- Computational Linguistics

**Source:** [https://dl.acm.org/doi/10.1145/3478281](https://dl.acm.org/doi/10.1145/3478281)
6. Design learning recommender systems

5. Create the student's digital twin

Source: https://dl.acm.org/doi/10.1145/3478281
Recommender systems

“... personalized information agents that provide recommendations: suggestions for items likely to be of use to a user” (Burke, 2007)
Recommender Systems

- Content-based Methods
- Collaborative Filtering (CF)
  - Neighborhood-Based CF
    - User-Based CF
  - Model-Based CF
    - Item-Based CF
- Hybrid Methods
User-Based Collaborative Filtering

User-Item Interaction Matrix

- Users we want to make a recommendation for
- Similar users in the dataset
- Recommend the most popular items among similar users

Less favorable - More favorable
User-Based Collaborative Filtering

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

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

Step 1

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

Step 2

Predict an unknown rating for the target user based on the best neighbors identified in Step 2.

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

- user \( u_i, i = 1, \ldots, n \)
- item \( p_j, j = 1, \ldots, m \)
- rating \( r_{ij} \)
Item-Based Collaborative Filtering

User-Item Interaction Matrix

- Preferred item

Less favorable  More favorable

Apply the KNN algorithm and find the most similar item(s)
Cold Start Problem
Data Sparsity Problem
Model-Based Collaborative Filtering

An underlying generative model that explains the user-item interactions.
Matrix Factorization via SVD or NNMF:

\[ R \approx XY^T = \hat{R} \]
Reconstructed Interaction Matrix ($\mathbf{R}$)

- $n$ users ($\mathbf{X}$)
- $m$ items ($\mathbf{Y}$)
- $K$ latent dimensions + additional examinee features
- $K$ latent dimensions + additional item features
Hybrid Recommender Systems

User Feature Matrix
- Clustering (e.g., k-means)
- User cluster vector

Rating Matrix
- User vector (i.e., embeddings)
- Item vector (i.e., embeddings)

Item Feature Matrix
- Clustering (e.g., k-means)
- Item cluster vector

Neural Network (or Deep Learning)
- Hidden Layers

Target Rating

\[ \hat{R} = X \times Y \times C \]
Adaptive Testing via Recommender Systems (RS)

Examinee-Item Interaction Matrix

Less information | More information

$n$ examinees

$m$ items (i.e., item pool)

Lower probability | Higher probability

$n$ examinees

$m$ items (i.e., item pool)
Item Selection for On-the-Fly Multi-Stage Adaptive Testing

Stage 1: A pre-assembled module

Stages 2 & 3: On-the-fly assembled modules via user-based and item-based collaborative filtering

No additional item or user feature used

Item selection using the recommenderlab package in R

https://doi.org/10.1177/01466216221124089
Examinee-Item Interaction Matrix

Training Dataset

Examinee-Item Interaction Matrix

300 or 600 items in the pool

2000 examinees

Stage 1
Pre-assembled module
10 items
or
20 items

Stage 2
Top-N recommendation
10 items
or
20 items

Stage 3
Top-N recommendation
10 items
or
20 items

On-the-fly module assembly using:

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

θ = [-3, -2.6, ..., 2.6, 3] → 500 simulated examinees per ability
<table>
<thead>
<tr>
<th>Item Bank Size</th>
<th>Method</th>
<th>30-item design</th>
<th>60-item design</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Bias</td>
<td>RMSE</td>
</tr>
<tr>
<td>300 items</td>
<td>UBCF</td>
<td>-0.019</td>
<td>0.362</td>
</tr>
<tr>
<td></td>
<td>IBCF</td>
<td>-0.007</td>
<td>0.428</td>
</tr>
<tr>
<td></td>
<td>MFI</td>
<td>-0.016</td>
<td>0.369</td>
</tr>
<tr>
<td>600 items</td>
<td>UBCF</td>
<td>0.044</td>
<td>0.464</td>
</tr>
<tr>
<td></td>
<td>IBCF</td>
<td>-0.010</td>
<td>0.341</td>
</tr>
<tr>
<td></td>
<td>MFI</td>
<td>-0.012</td>
<td>0.365</td>
</tr>
</tbody>
</table>
Personalized Scheduling for Adaptive Tests

What is the optimal test schedule for each student based on their learning progress?

Progress monitoring with Renaissance’s Star Reading and Star Math adaptive tests for K-12

Grade 2 \((n = 668,324)\) and Grade 4 \((n = 727,147)\)

2 to 18 test administrations per student

(Bulut, Shin, & Cormier, 2022; Shin & Bulut, 2022; Bulut, Cormier, & Shin, 2020)
User-Based Collaborative Filtering with Dijkstra's Shortest Path First Algorithm

- **Maximize** the positive and absolute score change between test administrations
- **Minimize** the number of test administrations

- Find similar students (with max score change + fewest test administrations)
- Select the most similar students based on Euclidean distance and recommend their schedule
**Standard Practice** = Schedules Determined by Teachers  \[ \text{RS} = \text{Recommender System} \]

<table>
<thead>
<tr>
<th>Evaluation Criteria</th>
<th>Grade 2</th>
<th></th>
<th>Grade 4</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Standard Practice</td>
<td>RS</td>
<td>Standard Practice</td>
<td>RS</td>
</tr>
<tr>
<td>Average number of tests administered</td>
<td>5.42</td>
<td><strong>3.51</strong></td>
<td>5.37</td>
<td><strong>3.84</strong></td>
</tr>
<tr>
<td>Average score change between tests</td>
<td>8.32</td>
<td><strong>12.25</strong></td>
<td>3.49</td>
<td><strong>4.63</strong></td>
</tr>
<tr>
<td>Range of tests required</td>
<td>(1, 18)</td>
<td><strong>(1, 5)</strong></td>
<td>(1, 17)</td>
<td><strong>(1, 6)</strong></td>
</tr>
<tr>
<td>Non-recommendable cases</td>
<td>-</td>
<td><strong>0.05%</strong></td>
<td></td>
<td><strong>0.10%</strong></td>
</tr>
</tbody>
</table>
Future Directions

• Recommender systems can involve real-time process data (e.g., response time) to consider test-taking engagement in adaptive testing.

• Recommender systems can be used with other psychometric models such as Bayesian Knowledge Tracing to measure mastery of content domain.

• Recommender systems utilizing deep learning algorithms can model both responses and sequential action data in adaptive learning environments.
  - Chen et al. (2019)’s Behavior Sequence Transformer Model
  - Wu et al. (2017)’s Recurrent Recommender Networks
Thank You!

For questions/comments:

bulut@ualberta.ca