Online Supplement to

Spoiled for Choice? Personalized Recommendation for Healthcare Decisions: A Multi-Armed Bandit Approach

Tongxin Zhou [1] · Yingfei Wang [2] · Lu (Lucy) Yan [3] · Yong Tan [2]

[1] W. P. Carey School of Business, Arizona State University, Tempe, Arizona 85287
[2] Michael G. Foster School of Business, University of Washington, Seattle, Washington 98195
[3] Kelley School of Business, Indiana University, Bloomington, Indiana 47405
tongxin.zhou@asu.edu · yingfei@uw.edu · yanlucy@indiana.edu · ytan@uw.edu

A1. Introduction of Recommendation Models

A1.1 Batch-Learning-Based Recommendation Systems and Online Machine Learning Models

The key rationale of batch-learning-based algorithms is "first learn, then earn." Specifically, the algorithms first take users' past behavior data to learn their preference patterns and then make predictions/recommendations based on the learned patterns. For example, collaborative filtering makes recommendations based on similarities in users' item-selection histories (Adomavicius and Tuzhilin 2005, Sedhain et al. 2014). To better exploit users' past behavior data, researchers have made various efforts to come up with different recommendation designs. For instance, one mainstream of studies has developed context-aware recommendation systems (CARS) to capture individual heterogeneity and the contextual dependency of user behaviors, as user behaviors are affected by personal circumstances and environments (Chen 2005, Adomavicius and Tuzhilin 2011, Unger et al. 2016). In addition, advanced modeling frameworks, such as matrix factorization (A. Mnih and Salakhutdinov 2008, Baltrunas et al. 2011) and hidden Markov models (Sahoo et al. 2012, Eskandanian and Mobasher 2018), are used to learn about latent information regarding user-item interactions.

Most recent literature has focused largely on deep-learning models to extract implicit, unstructured information (Zhang et al. 2019). It is suggested that implicit user feedback often provides more comprehensive insights into users' behavior patterns (Donkers et al. 2017). For example, a large body of research has sought to model unstructured item content (e.g., news, music, video) so that users' tastes can be learned at the latent feature level (An et al. 2019, Wu et al. 2019). In addition, as users' preferences on items may evolve and drift over time, researchers have proposed sequence-based neural networks to capture the sequential changes in users' behavior histories (Donkers et al. 2017, J. Tang and Wang 2018, Yu et al. 2019, Yuan et al. 2019). Existing sequence-based models include recurrent neural network (RNN) and long short-term memory (LSTM), as these network structures are suitable for learning the temporal dynamics of interactions (Zhang et al. 2019). Prior studies explored mainly the sequential patterns in individuals' item interactions (i.e., a single behavior path, such as clickstream, browsing sequence, etc.) (Donkers et al. 2017,

J. Tang and Wang 2018, Yu et al. 2019, Yuan et al. 2019). When it comes to healthcare recommendations, however, special considerations are required to properly learn the dynamics in users' healthcare preferences, such as the effect of expected health outcomes on individuals' health decisions and the joint patterns of multiple individuals' health and social behavior sequences. In this study, we address this research gap by proposing a novel deep user-representation learning model to account for these aspects.

In terms of recommendation diversity, recent studies have also proposed a number of strategies to address the over-specialization issue brought by batch-learning methods. For instance, Zhao et al. (2016) used popularity normalization to adjust the importance of items based on their popularity; An et al. (2019) proposed a LSTM-based model to capture both users' long-term preferences and their short-term interests to promote diversity in news recommendations. The major consideration of the extant diversity promotion techniques is to increase or randomize recommendation types; however, no pre-determined dimensions are used to guide recommendation diversification. In the recommendation context of specific fields such as healthcare, it is important to integrate domain-specific theories into the recommendation design to improve user representation and diversity promotion.

Finally, although batch-learning-based models present various techniques to exploit historical data, a key caveat is that, when historical data are not adequate to *fully* characterize users' preference patterns, batch learning can become less effective. The fundamental assumption for batch-learning models is that the probability that a user chooses an item in the historical dataset will remain the same at the time of the recommendation. In online healthcare recommendations, however, as individuals usually take a longer time to engage in healthcare interventions, the behavior data generated can be relatively scarce (especially compared to ecommerce settings where millions of clickstream data are immediately available). As a result, merely exploiting the archived data will likely be inadequate and biased, as individuals' behavior patterns may not be well represented. In addition, batch-learning models do not consider the uncertainty and confidence associated with users' preferences of each item. In other words, an item selected more frequently will be more likely to be recommended to a new user. However, the frequency of the selections is largely dependent on the original data collection policy, that is, the recommendation system itself. Even with frequent retraining, since the collected data always comes from the same recommendation algorithm over and over again, the fundamental assumption of batch learning will only reinforce the same historical recommendation pattern and never have a chance to explore potentially better options.

Together, these research gaps motivate us to come up with an online-learning-based approach, which synthesizes deep user representation to improve context learning and a theory-guided constraint to improve recommendation diversification. Compared to batch-learning-based methods, an online learning framework can adapt to users' changing preferences more quickly, as it not only exploits the historical information but also explores potential better options that are not presented in the archived data. In addition, online learning

can naturally promote recommendation diversity through the "exploration" scheme. Our diversity constraint further enhances this diversification process by structurally mixing intervention recommendations based on the identified key healthcare dimensions. In the following section, we introduce the rationale of an online learning framework and the current studies on healthcare recommendations.

A1.2 Providing Healthcare Support under Uncertainty: An Online-Learning Scheme

In this subsection, we introduce the online learning scheme for solving decision-making problems under uncertainty. Real-world decision-making problems usually face different levels of uncertainty. The uncertainty may be because decision-makers do not gather enough data to guide their decision making, or the decision-making environment is frequently changing (e.g., market instability, technology change, policy environment fluctuation, changing customer behaviors) (Cohen et al. 2007, Mehlhorn et al. 2015, Speekenbrink and Konstantinidis 2015). In an uncertain environment, it is important for the decision-makers to actively gather information during the decision-making process to improve their long-term rewards. On the one hand, decision-makers can reuse highly rewarding alternatives from the past to secure explicit short-term rewards ("exploitation") (Cohen et al. 2007, Mehlhorn et al. 2015); on the other hand, they may deviate from the current "best" option from time to time to learn the rewards associated with the less-explored actions ("exploration") so that they can minimize opportunity costs in the long run (Cohen et al. 2007, Speekenbrink and Konstantinidis 2015).

The tradeoff between "exploitation" and "exploration" is placed at the core of the online learning scheme when only partial information is revealed. In statistics and machine learning, multi-armed bandit (MAB) has been proposed as a classic online machine learning framework to solve the "exploitation-versus-exploration" tradeoff (Gittins 1979, Auer et al. 2002). Specifically, a MAB models a setting in which the underlying reward distribution for each action is unknown, and data can be obtained in a sequential order to update knowledge of the reward distribution. In each decision-making step, a MAB not only considers the current best option but also leverages opportunities to explore sub-optimal actions (Li et al. 2010, L. Tang et al. 2014, M.J. Kim and Lim 2015). This strategy enables MABs to gather adequate information for each action while ensuring as much reward as possible during the entire decision-making process (i.e., *earning while learning*) (Misra et al. 2019).

The key difference between MABs and batch-learning-based algorithms lies in the incorporation of exploration opportunity. Exploration enables learning of unknown territories and gathering information about the overall environment; this allows decision-makers to discover potential better alternatives that are not shown in the past data and thus optimize decision-making performance in the long run. The incorporation of exploration can help online healthcare platforms to quickly discover and adapt to the dynamics and diversity in users' healthcare interests, which may not be well captured by users' historical behavior data.

Online machine learning and multi-armed bandits have been used for recommendations in health-related contexts, aside from other perspectives such as the ontology of healthcare systems designs (J.-H. Kim et al. 2009, Subramaniyaswamy et al. 2019), the front-end user interface (Dharia et al. 2016), and system usability (Wendel et al. 2013). Although there have been several initial efforts to propose design algorithms to automate the generation of recommendation lists (Phanich et al. 2010, Zaman and Li 2014, Zhou et al. 2019, Tomkins et al. 2021), key challenges remain in terms of recommendation personalization and context adaptation. For instance, Phanich et al. (2010) did not account for individual-level heterogeneity to personalize healthcare recommendations for each user. Although Tomkins et al. (2021), Zaman and Li (2014), and Zhou et al. (2019) incorporated individual heterogeneity in their recommendation designs, they did not account for sequential patterns in health and health-behavior histories. For example, Zaman and Li (2014) utilized users' web semantics data but did not incorporate health-related information; Tomkins et al. (2021) and Zhou et al. (2019) considered individuals' attribute features such as demographics and health status. In a healthcare recommendation context, it can be essential to characterize individuals' complete behavior histories to capture the implicit information embedded in the health-related behavior sequences, given that health management is an evolving process (Morid et al. 2021). Our study adds to the healthcare recommendation literature by proposing a personalized recommendation engine algorithm that is well-guided by prominent health-behavior theories to learn users' evolving preferences and diverse tastes.

A1.3 Why Bandit?

Bandit is a particular type of online machine learning algorithms. One may wonder why we choose a bandit framework over other online-learning algorithms. In the following, we provide a detailed discussion on our motivation/intuition.

In a healthcare setting, future contexts may be affected by the actions chosen. It is typical of many contextual bandit problems, such as in online news recommendation. However, this effect is often relatively small over a short or medium time-span (compared to a full RL setting that will be discussed later). Meanwhile, it is usually the case that the noise level in the healthcare context is sufficiently high (Lei et al. 2017) so that a model ignoring this dependency provides a good approximation. The issue is also related to the assumptions of the data generating process. Contextual bandits are considered in three settings: stochastic bandit with i.i.d contexts and rewards, adversarial contexts with stochastic rewards, and a full adversarial setting. In a lot of practical settings, the contextual information can have potential correlations, leaving the i.i.d. assumption unrealistic. For example, a user that is stressed today is more likely to be stressed tomorrow than a user who is not stressed today. One may think of modeling the problem in a full adversarial setting with least assumptions. However, this type of algorithms (e.g. EXP4 (Beygelzimer et al. 2011)) may learn too slowly for a healthcare application, as they are designed to work in the worst case. Yet fast learning is critical due to the high abandonment rate of online healthcare platforms (McLean 2011).

The specific algorithm adopted in the paper, Thompson sampling (TS), was proved to be optimal (in terms of the convergence rate) under an adversarial context setting where the contexts can have arbitrary relationships with each other. Hence, TS appears more appropriate for healthcare recommendations (Lei et al. 2017, Tewari and Murphy 2017).

A more complete model of our problem can falls into the umbrella of reinforcement learning (RL), which is a strict generalization of contextual bandits. RL models a setting in which the agents' actions can influence the future states and exert a significant long-term effect. One difficulty is that many behavior sequences (for example, users' tendencies of self-monitoring and/or social activities) are exogenous and are not controlled nor influenced by the recommendation system. The dynamic evolution of the healthcare-related contexts over time are usually not yet well understood. Using model-based RL may need strong theory support for defining state transitions. Even for the model-free RL specification, it is challenging to define a state variable in this setting such that it satisfies the Markovian property and that it can be viewed as sufficient statistics entailing all the information required to choose the correct decision.

Another challenge is the data scarcity. In a healthcare setting where the effectiveness of each intervention largely depends on users' characteristics and past behavior sequences, the state space is expected to be large and/or continuous. As the neighborhoods of most states are not visited even once during learning episodes, it is difficult to obtain reliable estimates of value functions (Sutton and Barto 2018, Lattimore and Szepesvári 2020). Hence both in theory and in practice, the design of RL algorithms must deal with fundamental statistical problems of sparsity and model misspecification. Recent advancements on both the Atari platform (V. Mnih et al. 2015) and Go (Silver et al. 2016) leverage advances in deep learning for powerful function approximation, which can handle continuous state/action space, and yet require strong domain knowledge and large amounts of data to be successful.

A more advanced model for healthcare recommendations is the Contextual Decision Process (CDP) which makes no assumption on the context space. In CDP, the critical issue is how to generalize across contexts, since the agent usually does not encounter the same context twice. Unfortunately, without further assumptions, learning in CDPs is generally hard, since they subsume MDPs and POMDPs with arbitrarily large state/observation spaces. Designing practical algorithms for general CDP is still an open problem (Jiang et al. 2017).

Considering these challenges, we model the focal problem in contextual bandit, which gives us a more flexible design to model users' historical behavior sequences and a faster convergence rate to offer a reasonable approximation (Lei et al. 2017, Tewari and Murphy 2017). We leave the design of the full RL algorithms to future research.

A2. Deep Learning Embeddings

In this section, we provide additional details of our user-embedding models. The rationale for our deep learning design is that we use the intervention-embedding model as a building block for the later user-embedding model and the downstream intervention recommendation tasks. The goal for our intervention embedding model is to build a vector for each intervention (e.g., weight-loss challenge) such that similar intervention will have close embedding vectors. As long as the embedding learned from our model is able to properly distinguish interventions based on their relative similarity and intrinsic attributes, the user representation learning model will properly extract relevant signals to serve the intervention recommendation task. Let the learned embedding for intervention i be denoted as $c(i)$, which is a 16-dimensional vector. The learned intervention embeddings will be used in the user-embedding model to construct the loss function.

As noted in the paper, to better learn user representations and facilitate recommendation personalization, we focus on both users' static attribute features (i.e., the shallow features such as gender and age) and their sequence features regarding health histories and behavior paths (i.e., the deep features). Whereas users' attribute features provide a basic characterization of personal heterogeneity, the sequence features provide a more comprehensive understanding of users' health contexts (Johnson et al. 2002). Combing these two sets of features is alighed with our objective of offering dynamic healthcare-intervention suggestions based on users' changing health-management contexts. To process the sequence features, we apply the following LSTM module to the unified input sequence x^t:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x^t] + b_f), \tag{A1}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x^t] + b_i), \tag{A2}$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x^t] + b_c), \tag{A3}$$

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t, \tag{A4}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x^t] + b_o), \tag{A5}$$

$$h_t = o_t \cdot \tanh(C_t) \tag{A6}$$

and eventually obtain a sequence of latent vectors $[h_1, h_2, \ldots, h_t]$. To properly aggregate the latent vector sequence, we use a attention mechanism so that the neural network automatically detects the importance of each hidden vector and learns in favor of more important ones. As noted in the manuscript, the attention weight is learned from a fully connected neural network:

$$u_t = \tanh(W^{att} \cdot h_t + b^{att}), \tag{A7}$$

$$\chi_t = \exp(u_t \cdot \upsilon) / \sum_t \exp(u_t \cdot \upsilon), \tag{A8}$$

$$\eta_1 = \sum_t \chi_t h_t \ . \tag{A9}$$

Now we start to construct the loss branches. The main branch combining signals from both the shallow features and the deep features is built for an intervention embedding prediction:

$$\phi_2 = \text{Relu}(W_1^{user}\phi_1 + b_1^{user}), \tag{A10}$$

$$\phi_3 = W_2^{user}\phi_2 + b_2^{user}, \tag{A11}$$

where ϕ_1 is the latent layer that is constructed by concatenating endpoint of the shallow branch ξ_3 and the endpoint of the deep branch η_1, ϕ_3 is the final prediction. By omitting its dependency on the neural network parameters, ϕ_3 becomes a function of user i and time t: $\phi_3 = \phi_3(i,t)$.

For each user i and time (i.e., week) t, if there exists an intervention selection, we get the average intervention embedding $c(i,t)$, and we calculate the cosine distance between the user embedding $f(i,t)$ and $c(i,t)$. We sum up the distance over all users and time to construct our objective loss function. Note that a user may not select any intervention during some time t. We do not include such cases in our objective function, as they do not provide signals of user preference. Formally, we define our objective function as:

$$OBJ_{user} = -\frac{1}{|\Omega|}\sum_{(i,t)\in\Omega}\frac{f(i,t)\cdot c(i,t)}{\|f(i,t)\|_2\|c(i,t)\|_2}, \tag{A12}$$

where $\Omega = \{(i,t)\,|\,\text{there exists an intervention selection from user } i \text{ at time } t\}$, $|\Omega|$ denotes the cardinality of set Ω, and $\|\cdot\|_2$ is the l_2 norm. Note that there is a uniform denominator before the sum, regardless of how many interventions a user has selected during the time window $[1,T]$. Therefore, a more active user will contribute more terms in the objective function. In this way, the objective function will favor active users for more guidance during user representation learning.

For the auxiliary-loss branch, we apply fully connected operators to the endpoint of the deep features. In this way, the learning of the eLSTM module does not interfere with the wide features; hence, the gradients are well propagated to better learn the weights:

$$\eta_2 = \text{Relu}\left(W_1^{user}\phi_1 + b_1^{user}\right), \tag{A13}$$

$$\eta_3 = \text{Relu}(W_2^{health}\eta_2 + b_2^{health}). \tag{A14}$$

The auxiliary prediction η_3 is a prediction of the health condition $\alpha(u,t)$ (e.g. weight loss), and we choose a combined loss of MSE for absolute value prediction and cross-entropy loss for the health-measure sign (e.g., increase or decrease) prediction. Finally, the full loss function for user representation learning is written as:

$$L = -\frac{1}{|\Omega|}\omega_1 \sum_{(i,t)\in\Omega} \frac{f(i,t)\cdot c(i,t)}{\|f(i,t)\|_2 \|c(i,t)\|_2} + \frac{1}{|\Omega|}\omega_2 \sum_{(i,t)\in\Omega} |\eta_3 - \alpha(i,t)|^2 + \omega_3 CrossEntropy(sign(\eta_3), sign(\alpha(i,t))), \quad \text{(A15)}$$

which is a weighted combination of three losses.

A3. Screenshot of Weight-loss Challenges on the Focal Website

Figure A1 is a screenshot of the challenge page on the focal weight-loss platform. As can be seen, six challenges are included in this screenshot; the first three had not been started at the time we took this screenshot, and the last three were already in progress and could not be joined. For each of the challenges, users can find the challenge title at the top, followed by a short description of the challenge guidelines and the duration. Users can consider this information when they make choices about the challenges in which to participate. To join a challenge, users need to click on the challenge title and enter the inner homepage of the challenge, where a "join challenge" button can be found. We provide Figure A2, in which we show the internal homepage of the first challenge (i.e., "Sassy Summer Slimdown") as an example.

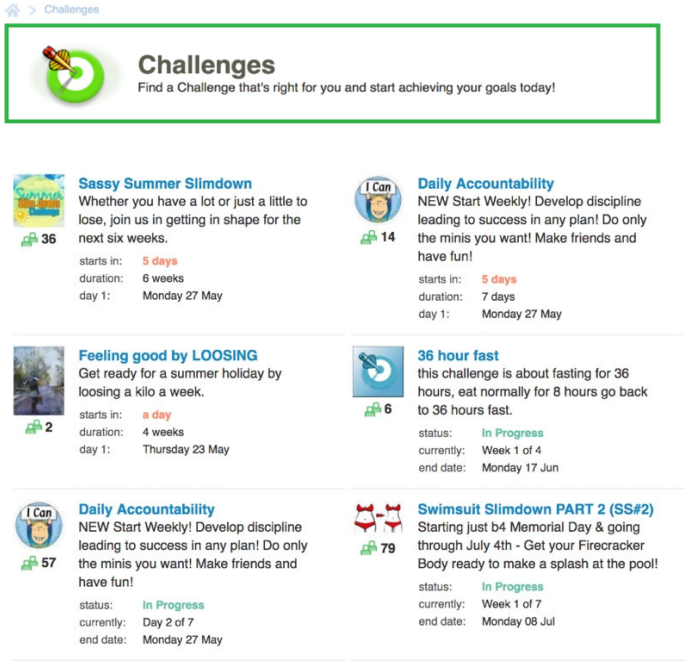


Figure A1 A Screenshot of Challenge Homepage

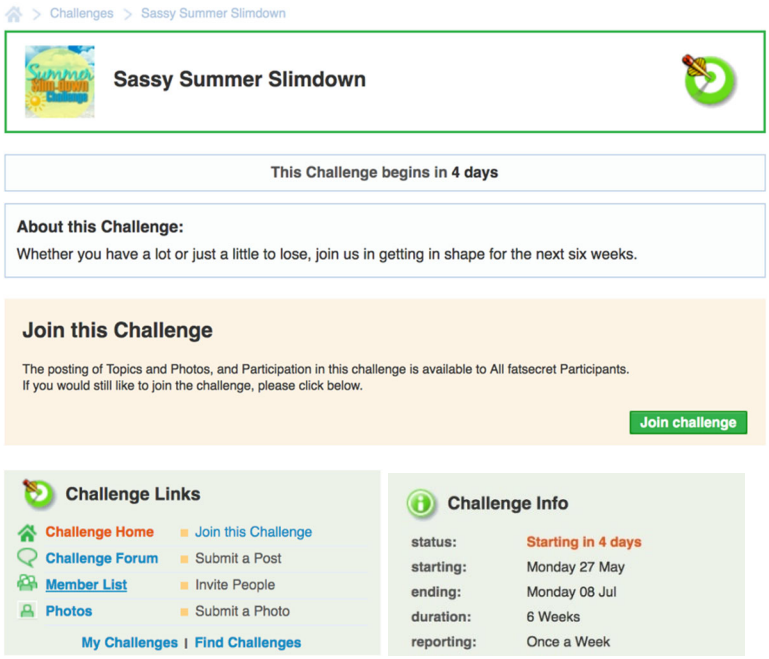


Figure A2 A Screen Shot of the Challenge Inner Homepage

A4. Data Summary Statistics

We summarize key data statistics in Table A1.

Table A1 Summary Statistics

Variable	Mean	Std. Dev.	Min	Max
Challenge Description				
No. of Available Challenges Each Week	51.06	2.30	45	54
Challenge Duration	5.01	2.84	1	20
			Total No. of Challenges: 165	
User Statistics				
Gender (Female = 1, Male = 0)	0.52	0.50	0	1
Age	38.07	11.26	19	57
Membership Duration	394.10	157.97	126	672
Initial Weight	210.62	49.10	127	298
Weight Changes Per Week	0.02	3.05	-24	22
No. of Weigh-ins Per Week	1.01	1.14	0	6
No. of New Friends Per Week	0.97	0.88	0	4
			Total No. of Users: 1049	
User-Challenge Interactions				
No. of Challenges Selected Per Week	1.94	1.69	0	13

Note: The total number of challenges is calculated based on the challenges that have been chosen by at least one user. Available challenges refer to the challenges that can be joined by new participants or existing participants.

A5. Input Features of User-Embedding Model

We summarize the input features used for constructing the user embeddings in the focal evaluation context in Table A2.

Table A2 Summary User-Embedding Input Features

User-embedding input features	**Feature Type**
Gender	Static attribute, categorical
Age	Static attribute, numerical
Initial weight	Static attribute, numerical
Membership duration	Static attribute, numerical
History of weigh-in records	Time-series, numerical
History of challenges chosen	Time-series, numerical (embeddings)
Sequence of the number of weigh-in activities	Time-series, numerical
Sequence of the number of friends	Time-series, numerical
Sequence of published forum posts	Time-series, textual

A6. Challenge Meta-Data Annotation

To illustrate how we annotate the challenge meta features, we provide several challenge examples in Table A3. We determine a challenge to be specific if it has a clearly defined goal (i.e., what to do), and we determine a challenge to be measurable if it can be easily measured with certain numeric criteria (i.e., how much to do). For example, in Table A3, the challenge "30 minutes on a treadmill for 6 weeks" is measurable but not specific, because individuals only know how much time to spend for exercising, but they do not know what exercise to do (walking or running). The challenge "Eat salad for dinner for 2 months" is specific but not measurable, because it clearly specifies the recipe but does not provide information about how much individuals should eat. The challenge "Jog 5 miles a week" is both specific and measurable, as it is clear in the activity form and the way to measure completion. Similarly, the challenge "Lose 10 lbs. in 4 weeks" is also both specific and measurable, as individuals can clearly identify the weight-loss goal and measure challenge completion.

Table A3 SMART Challenge Features

Challenge Description	***Diet***	***Activity***	***Weight-Loss***	***Specific***	***Measurable***	***Intensity***	***Duration***
30 minutes on a treadmill for 6 weeks	0	1	0	0	1	moderate	6
Eat salad for dinner for 2 months	1	0	0	1	0	moderate	8
Jog 5 miles a week	0	1	0	1	1	high	1
Lose 10 lbs. in 4 weeks	0	0	1	1	1	high	4

As shown in Table A3, a challenge can be diet-oriented (challenge 2), physical-activity-oriented (challenge 1, 3), and weight-loss-oriented (challenge 4). For each challenge type, we develop rules to determine their respective intensity levels. The general idea behind these rules is that the more restrictions a challenge places on individuals' calorie intake or expenditure, the higher the intensity level it has. In particular, for a diet-oriented challenge, we determine its intensity level to be low if it requires individuals

to change a minor eating or drinking habit, such as drinking more water or avoiding soda. We determine the intensity level to be moderate if the challenge requires individuals to change their diet structures (e.g., eating certain types of food) or to moderately restrict the amount of calorie intake (e.g., cutting the serving size). Finally, we determine the intensity level to be high if the challenge requires individuals to significantly restrict the amount of calorie intake, such as going through a fast. For a physical-activity-oriented challenge, the intensity level can be determined by the metabolic equivalent of task (MET). Specifically, walking and strolling are defined to be at a low-intensity level; activities such as bicycling, yoga, and walking with speed more than 3 mph are determined to be at a moderate intensity level; and activities such as jogging or running, situps, pushups, or jumping rope are considered to be at a high-intensity level. For a weight-loss-oriented challenge, we define the intensity level based on the average weight-loss rate per week. According to the Centers for Disease Control and Prevention (CDC), individuals can generally lose 1-2 lbs. per week by burning 500-1000 calories more than they consume each day. Therefore, challenges that require individuals to lose weight within this range are considered moderate. Challenges that require individuals to lose less than 1 lbs. per week are considered low-intensity, and challenges that require individuals to lose more than 2 lbs. per week are considered high-intensity.

In addition to the above-described SMART-based features, we consider two other features that may affect individuals' engagement in challenge participation: *Motivational* and *Self-Monitoring*. *Motivational* denotes whether a challenge contains encouraging words or sentences that can help individuals to build up the inner motivation for weight-loss. For example, "Just adding a piece of fruit to the diet daily. Nothing crazy here, healthy habits take a while to kick in, so stick with it!" is a diet-oriented challenge that is described in a motivational way. *Self-Monitoring* describes whether a challenge encourages individuals to regularly self-monitor their weight-loss progress (e.g., body weight, daily diet, running mileage, etc.). For instance, "8 weeks of eating a 1,200 calorie diet; keep track of all food and weigh in every Monday" is a diet-oriented challenge that encourages individuals to self-monitor their diet and body weight.

A7. Implementation of Benchmark Models

The benchmark models used to examine our recommendation performance include state-of-the-art batch-learning-based models and online-learning bandit models. In the following, we provide the implementation details of these models.

A7.1 Benchmark Batch-Learning-Based Recommendation Algorithms

We first consider classic recommendation models, such as context-aware collaborative-filtering and content-based filtering models. In particular, we consider Context-Aware Collaborative Filtering (CACF) (Chen 2005), Social Collaborative Filtering (SCF) (Sedhain et al. 2014), Probabilistic Matrix Factorization (PMF) (A. Mnih and Salakhutdinov 2008), Context-Aware Matrix factorization (CAMF) (Baltrunas et al. 2011), ordinary content-based filtering (Pon et al. 2007, Bieliková et al. 2012), and mixed hybrid models

(Burke 2002). These models provide a basic comparison for our model. As recent recommendation literature has mostly focused on deep-learning methods, we also incorporate more advanced deep recommenders as comparison benchmarks. Specifically, as our context involves users' sequential behaviors, we investigate sequence-based models, such as SLi-Rec (Yu et al. 2019), Caser (J. Tang and Wang 2018), Gru4Rec (Hidasi et al. 2016), A2SVD (Yu et al. 2019), and NextItnet (Yuan et al. 2019). We also consider two content-based deep models, LSTUR (An et al. 2019) and NPA (Wu et al. 2019), among which LSTUR explicitly accounts for users' diverse preferences. In the following, we explain how we implement these models for recommendations.

Table A4 Summary of Comparison Benchmark Algorithms

Benchmark Models	**Description**
FAST	FASTAI Embedding Dot Bias (Howard and Gugger 2020);
SLi-Rec	Sequential-based algorithm that aims to capture both long and short-term user preferences using attention mechanism, a time-aware controller and a content-aware controller (Yu et al. 2019);
Caser	Algorithm based on convolutions that aim to capture both user's general preferences and sequential patterns (J. Tang and Wang 2018);
GRU4Rec	Sequential-based algorithm that aims to capture both long and short-term user preferences using recurrent neural networks (Hidasi et al. 2016);
A2SVD	Sequential-based algorithm that aims to capture both long and short-term user preferences using attention mechanism (Yu et al. 2019);
NextItNet	Algorithm based on dilated convolutions and residual network that aims to capture sequential patterns (Yuan et al. 2019);
LSTUR	Neural recommendation algorithm with long- and short-term user interest modeling to capture users' diverse tastes (An et al. 2019);
NPA	Neural recommendation algorithm with personalized attention network (Wu et al. 2019);
CACF	Context-Aware Collaborative Filtering, which incorporates the contexts of users' item selections as weights into a normal collaborative filtering procedure (Chen 2005);
SCF	Social Collaborative Filtering, a neighborhood-based method for cold-start collaborative filtering in a generalized matrix algebra framework (Sedhain et al. 2014);
PMF	Probabilistic Matrix Factorization, a model-based approach that uses matrix factorization under a probabilistic framework (A. Mnih and Salakhutdinov 2008);
CAMF	Context-Aware Matrix Factorization, which is an extension of the classic matrix factorization approach for incorporating contextual information (Baltrunas et al. 2011);
CB	An ordinary content-based filtering approach (Pon et al. 2007, Bieliková et al. 2012);
hybrid_pure	A hybrid model that combines pure collaborative filtering with CB using mixed hybridization (Burke 2002);
hybrid_cacf	A hybrid model that combines CACF and CB using mixed hybridization;
UCB	A bandit algorithm that incorporates the opportunity for exploration by choosing the arm with the highest upper bound of the reward confidence interval;
ε-greedy	A bandit algorithm that chooses the arm with the seemingly highest average reward with probability $1-\varepsilon$ and explores a random arm with probability ε;

Context-Aware Collaborative Filtering (CACF) incorporates the contexts of users' item selections as weights into a normal collaborative filtering procedure (Chen 2005). Formally, we define the decision of

user i on item k weighted by contexts; that is, $R_{itk} = \eta_1 \sum_{\tau<t} r_\tau(i,k) \cdot sim(\boldsymbol{v}_{i\tau k}, \boldsymbol{v}_{itk})$, where η_1 is a normalizing factor such that the weights sum to unity, and $sim(\boldsymbol{v}_{i\tau k}, \boldsymbol{v}_{itk})$ denotes the similarity between contexts $\boldsymbol{v}_{i\tau k}$ and $\boldsymbol{v}_{itk}$. Define R_t^{new} as the weighted challenge selection matrix at time t; that is, $R_t^{new} = \{R_{itk}\}_{i\in[I],k\in C_t}$. The predicted decision of user i on item k is derived based on the weighted challenge selection data; that is, $\hat{r}_t(i,k) = \overline{r}_i + \eta_2 \sum_{j\neq i} (R_{jtk} - \overline{r}_j) \cdot sim(R_t^{new}[i,], R_t^{new}[j,])$, where $\overline{r}_i$ denotes user i's overall challenge selection tendency, $sim(R_t^{new}[i,], R_t^{new}[j,])$ denotes the similarity between user i and user j's challenge selection histories, and η_2 is a normalizing factor. In each recommendation period, we provide the top-*K* challenges that have the highest prediction score and obtain user feedback to update R_{itk}.

Social Collaborative Filtering (SCF) formulates a neighborhood-based method for cold-start collaborative filtering in a generalized matrix algebra framework (Sedhain et al. 2014). Specifically, let $X_{I\times d}$ be users' context matrix, with each row representing a set of context features (e.g., gender, age, social activities, etc.) of user i. Let $R_{I\times L}$ be users' challenge selection matrix. I is the number of users, L is the total number of challenges, and d is the feature dimension. SCF calibrates the pseudo similarity between users' context features and challenge selection histories; that is, $w_{jk} = sim(X[,j], R[,k])$. The method then utilizes the pseudo similarity matrix W to make predictions on users' challenge selection decisions; that is, $\hat{R}_{ik} \propto sim(X[i,], W[,k])$, where $\hat{R}_{ik}$ is user i's predicted decision propensity on item k, $W = \{w_{jk}\}_{j\in[d],k\in[L]}$. For both CACF and SCF, we use cosine similarity to measure the affinity between two vectors. In each period, we provide the top-K challenges that have the highest R_{ik} values to each user and collect user feedback based on their preference set. We then use user feedback to update the challenge selection matrix R to calculate users' decision propensity in the next round.

Probabilistic Matrix Factorization (PMF) is a model-based collaborative filtering method that assumes users' preferences to be determined by a set of latent factors. It models the probabilistic distribution of users' preferences based on a latent user matrix U and a latent item matrix V. We define the conditional distribution of users' challenge selection tendency as $p(R \mid U, V, \sigma^2) = \prod_{i\in[I]} \prod_{k\in[L]} N(R_{ik} \mid U_i^T V_k, \sigma^2)$, where U_i and V_k denote user-specific and item-specific latent feature vectors, respectively. The priors of user and item latent features are assumed to be normally distributed; that is, $p(U \mid \sigma_U^2) = \prod_i N(U_i \mid 0, \sigma_U^2 I)$, $p(V \mid \sigma_V^2) = \prod_k N(V_k \mid 0, \sigma_V^2 I)$. The objective of PMF is to estimate U and V such that the posterior of R is maximized, or, equivalently, the following sum-of-squared-errors function with quadratic regularizers is

minimized: $F = \sum_{i\in[I]}\sum_{k\in[L]}(R_{ik} - U_i^T V_k)^2 + \lambda_U \sum_{i\in[I]}\|U_i\|_2^2 + \lambda_V \sum_{k\in[L]}\|V_k\|_2^2$, where $\lambda_U = \sigma^2 / \sigma_U^2$, $\lambda_V = \sigma^2 / \sigma_V^2$. User i's decision on item k can then be predicted by $\hat{R}_{ik} \sim N(\hat{U}_i^T \hat{V}_k, \hat{\sigma}^2)$. Each period, we construct users' challenge selection matrix R based on the "historical" data and solve the above convex optimization problem. Then, we calculate $\hat{R}_{ik}$ to make recommendations. Once users' feedback is collected, the "historical" data will be updated for recommendations in the next period.

Context-Aware Matrix factorization (CAMF) is an extension of the classic matrix factorization approach for incorporating contextual information (Baltrunas et al. 2011). Specifically, it minimizes the following objective function:

$$\min_{U,V,B,b} \sum_{i,k}[(R_{ik} - U_i^T V_k - \overline{r}_k - b_i - \sum_j B_{jc_j})^2 + \lambda(b_i^2 + \|U_i\|_2^2 + \|V_k\|_2^2 + \sum_j \sum_{c_j} B_{jc_j}^2)], \quad \text{(A16)}$$

where R_{ik} denotes user i's decision on item k; U_i is the latent user feature vector, and V_k is the latent item feature vector; $\overline{r}_k$ is the average selection rate of item k in the data; b_i is the baseline parameter for user i; B_{jc_j} are parameters capturing the effects of contextual conditions, in which j indexes the contextual factors and c_j indexes the possible values of contextual factor j; λ is a regularization factor. Users' challenge-selection decisions can be predicted by $\hat{R}_{ik} = \hat{U}_i^T \hat{V}_k + \overline{r}_k + \hat{b}_i + \sum_j \hat{B}_{jc_j}$. Similarly, we update R_{ik} with the collection of user feedback and obtain new estimates of $\hat{R}_{ik}$ to make top-K recommendations in each period.

Our content-based filtering method is similar to the Prod2Vec or Item2Vec method in the prior studies (Barkan and Koenigstein 2016, Vasile et al. 2016). We use challenge embeddings to evaluate the similarity between two challenges, and we recommend the top K challenges that are most similar to users' challenge-selection profile. Again, our similarity measure is cosine. Our hybrid models combine content-based filtering and collaborative filtering by a mixed hybridization approach (Burke 2002). In particular, the first hybrid model combines content-based filtering with a pure collaborative filtering model, which entirely leverages users' challenge selection histories. Our second hybrid model combines content-based filtering with CACF, which further incorporates contextual information in collaborative filtering. For the above described models (i.e., CACF, SCF, PMF, etc.), we use the first 4 weeks of data to warm up the predictors; during the recommendation period, as we mentioned, we use additional user profile data to re-train the predictors after each week.

The parameter settings for the deep learning models are reported as follows. For SLi-Rec, the batch size is 100, the learning rate is 0.001, and the number of epochs is 20. For Caser, GRU4Rec, A2SVD, and NextItNet, the batch size is 200, the learning rate is 0.001, and the number of epochs is 20. For LSTUR and

NPA, the batch size is 32, the learning rate is 0.0001, the number of epochs is 10, and the dropout rate is 0.2. The FAST model uses n_factors of 40 and weight decay of 0.1.

A7.2 Benchmark Bandit Algorithms

UCB algorithms incorporate the opportunity for exploration by choosing the alternative with the highest upper confidence bound of the reward (Auer et al. 2002). That is, a UCB chooses an alternative a_t on round t according to $a_t = \arg\max_k (\mu_t(k) + U_t(k))$, where k indexes alternatives and $U_t(k)$ is the upper confidence bound that plays the role of uncertainty bonus. ε-greedy deploys a random exploration strategy. Specifically, it chooses the arm with the seemingly highest average payoff with probability $1-\varepsilon$ and explores a random arm with probability ε. We present these two algorithms as follows.

UCB Algorithm

Input: Prior mean m_j and prior variance σ_j for parameters θ_j, $j = 1,2,\cdots d$.

Let $\boldsymbol{\theta} = (\theta_1,\ldots,\theta_d)$, $\boldsymbol{m} = (m_1,\ldots,m_d)$, $\boldsymbol{A} = diag(\sigma_1,\ldots,\sigma_d)$.

Constant λ to control the extent of exploration.

For $t = 1,2,...,T$ **do**

For $i = 1,2,...,I$ **do**

For arm $k \in C_t$ (C_t is the available chalenge set at week t) **do**

Compute $r_t(i,k) = (1+\exp(-\boldsymbol{v}_{itk}^T \boldsymbol{m}))^{-1} + \lambda\sqrt{\boldsymbol{\zeta}^T \boldsymbol{A}^{-1}\boldsymbol{\zeta}}$,

where $\boldsymbol{\zeta} = -\boldsymbol{v}_{itk}\exp(-\boldsymbol{m}^T\boldsymbol{v}_{itk}) / (1+\exp(-\boldsymbol{m}^T\boldsymbol{v}_{itk}))^2$, $\boldsymbol{v}_{itk}$ is the context vector.

End for

Select the K arms with the highest $r_t(i,k)$ value, denote this set as S_{it}, $|S_{it}| = K$.

Observe a new batch of data $(\boldsymbol{v}_{itk}, r_t(i,k))$, $i \in [I]$, $k \in S_{it}$.

Update the posterior mean by:

$$\boldsymbol{m} = \arg\min_{\boldsymbol{\theta}} \frac{1}{2}\sum_{j=1}^{d}\sigma_j^{-1}(\theta_j - m_j)^2 + \sum_{i=1}^{I}\sum_{k\in S_{it}}\log(1+\exp(-r_t(i,k)\boldsymbol{\theta}^T\boldsymbol{v}_{itk})).$$

Update the posterior variance by:

$$\sigma_j^{-1} = \sigma_j^{-1} + \sum_{i=1}^{I}\sum_{k\in S_{it}} v_{jitk}^2 p_{itk}(1-p_{itk}),$$

where $p_{itk} = (1+\exp(-\boldsymbol{m}^T\boldsymbol{v}_{itk}))^{-1}$, v_{jitk} is the j-th element of $\boldsymbol{v}_{itk}$.

End for

End for

ε -greedy Algorithm

Input: Prior mean m_j and prior variance σ_j for parameters $\theta_j, j = 1,2,\cdots d$. Random exploration rate ε.

For $t = 1,2,...,T$ **do**

For $i = 1,2,...,I$ **do**

For arm $k \in C_t$ (C_t is the available chalenge set at week t) **do**

Compute $r_t(i,k) = (1+\exp(-\boldsymbol{v}_{itk}^T \boldsymbol{m}))^{-1}$, where $\boldsymbol{\theta} = (\theta_1,\ldots,\theta_d)$, $\boldsymbol{v}_{itk}$ is the context vector.

End for

Draw $y \sim Uniform(0,1)$

If $y < \varepsilon$, randomly select K arms; Otherwise, select the K arms with the highest $r_t(i,k)$ value.

Denote the set of the selected arms as S_{it}, $|S_{it}| = K$.

Observe a new batch of data $(\boldsymbol{v}_{itk}, r_t(i,k)),\ i \in [I],\ k \in S_{it}$.

Update the posterior mean by:

$$\boldsymbol{m} = \arg\min_{\theta} \frac{1}{2}\sum_{j=1}^{d} \sigma_j^{-1}(\theta_j - m_j)^2 + \sum_{i=1}^{I}\sum_{k\in S_{it}} \log(1+\exp(-r_t(i,k)\boldsymbol{\theta}^T \boldsymbol{v}_{itk})).$$

Update the posterior variance by:

$$\sigma_j^{-1} = \sigma_j^{-1} + \sum_{i=1}^{I}\sum_{k\in S_{it}} v_{jitk}^2 p_{itk}(1-p_{itk}),$$

where $p_{itk} = (1+\exp(-\boldsymbol{m}^T \boldsymbol{v}_{itk}))^{-1}$, v_{jitk} is the j-th element of $\boldsymbol{v}_{itk}$.

End for

End for

For the UCB algorithm, we vary the parameter λ to tune the extent of exploration. Our parameter tuning results suggest that the optimal value of λ is 0.3. For the ε -greedy algorithm, we vary the exploration rate ε, and the optimal value is 0.5.

A8. Evaluation Approach

A8.1 Recision, Recall, MAP, and nDCG

Precision and recall are defined as:

$$precision_{i,t} = |S_{it} \cap L_i| / |S_{it}|, \tag{A17}$$

$$recall_{i,t} = |S_{it} \cap L_i| / |L_i|, \tag{A18}$$

where S_{it} is the recommendation set for user i at week t, and L_i denotes user i's preference set, which contains the challenges that the user has chosen in the historical data. MAP is calculated as:

$$MAP = \frac{1}{I \times T}\sum_{i,t}\left\{\sum_{j=1}^{|S_{it}|}\left[\frac{|S_{it}[1:j] \cap L_i|}{j} \times rel_j\right] / |L_i|\right\}, \tag{A19}$$

where $S_{it}[1:j]$ denotes the first j items in S_{it}; rel_j is the relevancy for the j-th item in S_{it}, that is, $rel_j = 1$ if the j-th item in S_{it} is selected in L_i. We calculate nDCG as:

$$nDCG = \left[\sum_{j=1}^{|S_{it}|}\frac{rel_j}{\log_2(j+1)}\right] / \left[\sum_{j=1}^{|S_{it}|}\frac{2^{REL_j}-1}{\log_2(j+1)}\right], \tag{A20}$$

where REL_j is the relevancy for the j-th item in the ideal ranking list.

A8.2 Doubly-Robust Estimation

Doubly-robust estimation is an offline evaluation approach that enables one to assess the value of a policy using a historical dataset (Dudík et al. 2011). This method combines two popular policy evaluation methods, direct simulation (DS) and inverse propensity score (IPS). The former forms an estimate of the expected reward conditioned on the context and action, and the latter forms an approximation of the action propensity in data collection to correct for the shift in action proportions between the offline dataset and the focal policy to be evaluated. By combining these two methods, the doubly-robust estimator is able to adjust the potential bias caused by the data collection process.

Formally, let $G = \{(\boldsymbol{x}, a, r_{a,\boldsymbol{x}})\}$ denote the offline evaluation dataset, with each row containing information about context $\boldsymbol{x}$, action a, and reward $r_{a,\boldsymbol{x}}$. Corresponding to our focal context, $\boldsymbol{x}$ refers to users' weight-management contexts, such as weight variations and weight-management behavior paths; a refers to a certain challenge that the platform provides; and $r_{a,\boldsymbol{x}}$ is users' challenge-selection decision, which depends on individuals' weight-management contexts and the challenge attributes. As noted in the manuscript, we use user embeddings and challenge embeddings to capture individuals' weight-management contexts and challenge attributes, respectively. As the platform did not promote any personalized recommendation during our data-collection period, we observe the same set of challenges for each individual. The number of challenges we observe each week may be different, as the platform may provide different challenges across time. Figure A3 provides an illustration of our offline dataset.

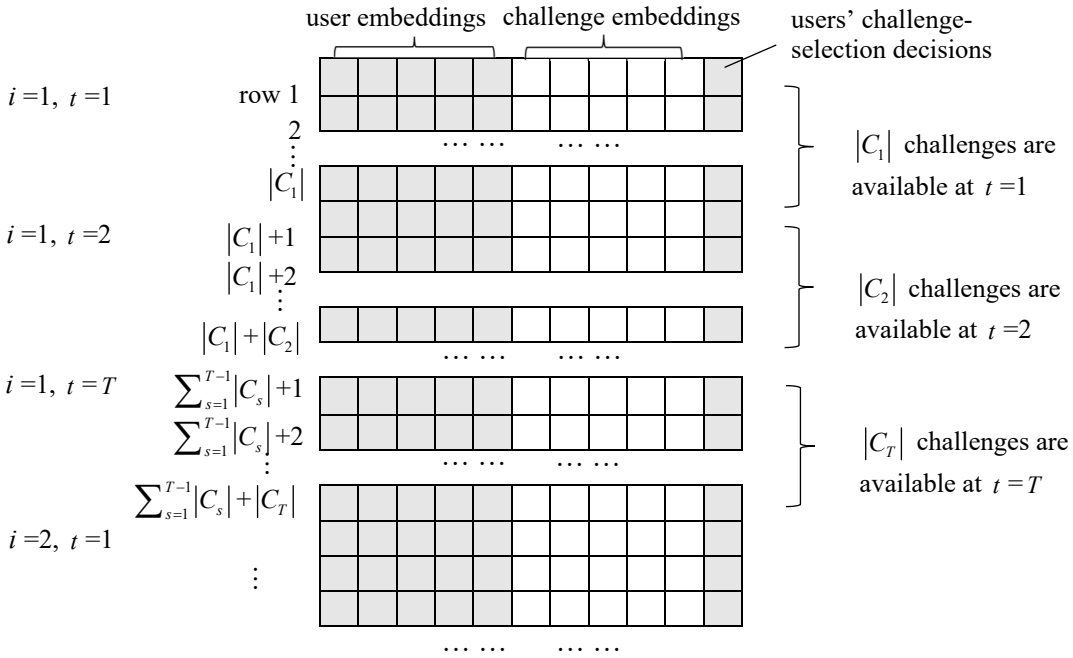


Figure A3 Offline Dataset for Performance Evaluation

In Figure A3, C_t denotes the set of challenges that are available at time t. The total number of rows in the dataset G is $I \cdot \sum_{t=1}^{T} |C_t| \triangleq |G|$, in which I is the number of users and T is the number of time periods. The cumulative recommendation performance during the entire periods, which is also known as the "policy value," can be estimated based on G as follows:

$$\overline{R}_T = \hat{V}_G^{\pi} = \frac{1}{|G|} \sum_{i,t,a \in C_t} \left\{ \frac{1}{K} \sum\nolimits_{k \in S_{it}} \left[\hat{\varphi}_{\boldsymbol{v}_{itk}} + \frac{(r_{a,\boldsymbol{x}_{it}} - \hat{\varphi}_{\boldsymbol{v}_{ita}}) I_{[k=a]}}{\hat{p}(a \mid \boldsymbol{x}_{it})} \right] \right\}, \tag{A21}$$

where $\hat{\varphi}$ is an estimate of the expected reward conditioned on individuals' decision-making context, which can be obtained through estimating a reward function using the offline data, $\hat{p}(a \mid \boldsymbol{x})$ is the probability of observing the platform providing challenge a under individuals' weight-management context $\boldsymbol{x}$, and S_{it} is the set of challenges provided to user i at time t according to the recommendation policy.

A high-level rationale behind this method is that, when data are not available, we directly apply $\hat{\varphi}$ to predict the reward; otherwise, we use the actual data to adjust the predictor $\hat{\varphi}$, taking into account the propensity of observations in the data. Following the convention, we randomly split users into a training set and a test set, and we use the training set to build the reward predictor $\hat{\varphi}$ and use the test set to evaluate recommendation performance. In particular, we adopt a logistic reward predictor: $r_{a,\boldsymbol{v}} = (1 + \exp(-\boldsymbol{v}^T \boldsymbol{\zeta}))^{-1}$, where $\boldsymbol{\zeta}$ is fitted using the training data only. Because the platform does not prioritize weight-loss challenges for users, the probability of observing each challenge conditioned on context, i.e., $\hat{p}(a \mid \boldsymbol{x})$, is uniform across the available challenge set. That is, $\hat{p}(a \mid \boldsymbol{v}, a \in C_t) \equiv |C_t|^{-1}$.

For completeness, we define the reward for providing challenge k to user i at time t as:

$$r_t(i,k) = \frac{1}{|C_t|} \sum_{a \in C_t} \left[\hat{\varphi}_{\boldsymbol{v}_{itk}} + \frac{(r_{a,\boldsymbol{x}_{it}} - \hat{\varphi}_{\boldsymbol{v}_{ita}}) I_{[k=a]}}{\hat{p}(a \mid \boldsymbol{x}_{it})} \right], \tag{A22}$$

and the performance (or reward) for all the recommendations provided to users at time t is:

$$\overline{r}_t = \frac{1}{I \cdot |C_t|} \sum_{i,a \in C_t} \left\{ \frac{1}{K} \sum\nolimits_{k \in S_{it}} \left[\hat{\varphi}_{\boldsymbol{v}_{itk}} + \frac{(r_{a,\boldsymbol{x}_{it}} - \hat{\varphi}_{\boldsymbol{v}_{ita}}) I_{[k=a]}}{\hat{p}(a \mid \boldsymbol{x}_{it})} \right] \right\}. \tag{A23}$$

A8.3 Simulation

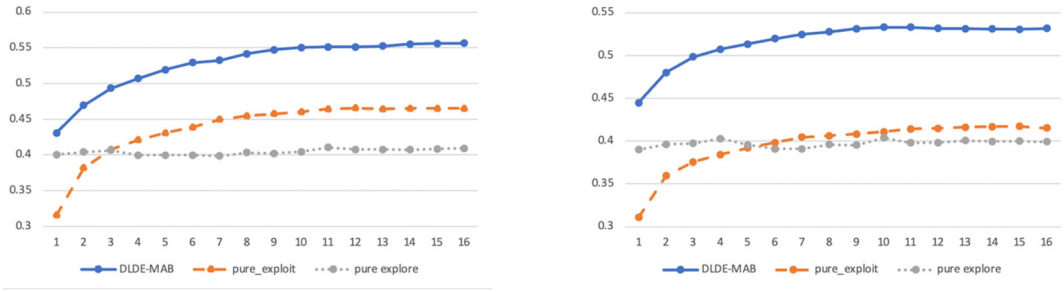
In light of previous theoretical MAB studies (Sani et al. 2012, Hertz et al. 2018), we also consider a simulation approach to evaluate our model. We use logistic regression to train a predictor for users' binary challenge-selection decisions. Specifically, let $r_t(i,k)$ denote user i's decision for challenge k at week t, and let $\boldsymbol{v}$ be a context vector containing user embeddings, challenge embeddings and their interaction terms. Our logistic predictor can be expressed as $r_t(i,k) = (1 + \exp(-\boldsymbol{v}_{itk}{}^T \boldsymbol{\zeta}))^{-1}$. Again, we build our predictor on a

training dataset, and the performance evaluation is conducted on a test set. We simulate users' challenge selection decisions based on the estimated choice-making propensity; that is, $\hat{r}_t(i,k) = (1+\exp(-\boldsymbol{v}_{itk}{}^T\hat{\boldsymbol{\zeta}}))^{-1}$. The average challenge-selection rate among users at week t is estimated by: $\bar{r}_t = \frac{1}{I \cdot K}\sum_i \sum_{k \in S_{it}} \hat{r}_t(i,k)$.

A9. Additional Evaluation Results

A9.1 Comparison with Pure Exploitation and Pure Exploration

To show that online learning is indeed preferable in our setting, we compare our model with pure exploitation and pure exploration strategies. The former selects the best options based on the current knowledge, whereas the latter fully randomizes recommendations. We plot the learning curves of our model and those of pure exploitation and pure exploration in Figure A4. In each plot, the *x*-axis denotes recommendation rounds, and the *y*-axis denotes the average challenge selection rate up to round t, which is computed based on users' preference sets as revealed by the challenge-selection data. As can be seen, our model achieves a higher challenge selection rate across all periods as compared to pure exploitation and pure exploration. The learning curve of pure exploration is almost flat, as the pure exploration strategy randomly provide recommendations without utilizing users' feedback information. Pure exploitation is shown to have a good learning rate at the beginning; however, it can get stuck in a local optimal without further improvement. Together, these results demonstrate the necessity of balancing between exploitation and exploration in the focal recommendation setting. Whereas exploitation enables usage of past information, exploration seeks better opportunities to improve overall learning performance.



(1) Top-5 Recommendations (2) Top-10 Recommendations

Figure A4 The Learning Curves of Our Model and Pure Exploitation/Exploration

A9.2 Robustness Checks

A9.2.1 Evaluation Results for Top-5 Recommendations in Experiment 1&2

Table A5 Comparison with State-of-the-Art Benchmarks (Top-5 Recommendations)

Model	Precision@5	Recall@5	nDCG@5	MAP@5	DR@5	Simu@5
DLDE-MAB	**0.5519**	**0.0337**	**0.7747**	**0.0257**	**0.4837**	**0.4439**
FAST	0.3823 ***	0.0235 ***	0.5368 ***	0.0112 ***	0.3374 ***	0.3592 ***
SLi_Rec	0.3955 ***	0.0230 ***	0.1667 ***	0.0163 ***	0.4042 ***	0.4039 ***
Caser	0.3991 ***	0.0232 ***	0.1387 ***	0.0166 ***	0.4045 ***	0.3980 ***
GRU4Rec	0.3924 ***	0.0228 ***	0.1557 ***	0.0165 ***	0.3984 ***	0.3946 ***
A2SVD	0.3929 ***	0.0228 ***	0.1313 ***	0.0166 ***	0.3871 ***	0.3866 ***
NextItNet	0.3857 ***	0.0224 ***	0.1553 ***	0.0167 ***	0.3893 ***	0.3877 ***
LSTUR	0.5309 ***	0.0327 ***	0.2091 ***	0.0181 ***	0.4369 ***	0.4109 ***
NPA	0.4735 ***	0.0291 ***	0.1200 ***	0.0206 ***	0.4334 ***	0.4413
CACF	0.3260 ***	0.0215 ***	0.5038 ***	0.0145 ***	0.4425 ***	0.4354 *
SCF	0.4147 ***	0.0257 ***	0.6634 ***	0.0184 ***	0.4194 ***	0.3997 ***
PMF	0.2096 ***	0.0131 ***	0.4448 ***	0.0075 ***	0.3565 ***	0.3929 ***
CAMF	0.3388 ***	0.0216 ***	0.5882 ***	0.0157 ***	0.3569 ***	0.3930 ***
CB	0.2525 ***	0.0160 ***	0.4726 ***	0.0087 ***	0.3920 ***	0.4327 **
hybrid_pure	0.3076 ***	0.0199 ***	0.6929 ***	0.0129 ***	0.3599 ***	0.4277 ***
hybrid_cacf	0.2864 ***	0.0187 ***	0.4714 ***	0.0116 ***	0.4307 ***	0.4378
UCB	0.3416 ***	0.0206 ***	0.6699 ***	0.0182 ***	0.4224 ***	0.4220 **
ε -greedy	0.4706 ***	0.0289 ***	0.7696 ***	0.0211 ***	0.4648 *	0.4207 **

Note: Asterisk in superscript denotes that a benchmark model performs significantly worse than the proposed model. Significance levels are: * $p < 0.1$, ** $p < 0.05$, *** $p < 0.01$.

Table A6 Ablation Analysis Results (Top-5 Recommendations)

Model	Precision@5	Recall@5	nDCG@5	MAP@5	DR@5	Simu@5
DLDE-MAB	**0.5519**	**0.0337**	**0.7747**	**0.0257**	**0.4837**	**0.4439**
No cons	0.4969 ***	0.0304 ***	0.7710 ***	0.0228 ***	0.4690 *	0.4344
No user embs	0.5273 **	0.0324 **	0.7604 ***	0.0239 ***	0.4758	0.4056 ***
No chlng embs	0.4217 ***	0.0258 ***	0.7304 ***	0.0197 ***	0.3871 ***	0.3646 ***
No embs	0.4720 ***	0.0290 ***	0.7529 ***	0.0233 ***	0.4496 ***	0.2956 ***
No embs no cons	0.3896 ***	0.0238 ***	0.7417 ***	0.0198 ***	0.3791 ***	0.3760 ***
DLDE-Collab	0.5058 ***	0.0310 ***	0.7561 ***	0.0228 ***	0.4774	0.3970 ***
DLDE-Tabular	0.4416 ***	0.0271 ***	0.7326 ***	0.0199 ***	0.4179 ***	0.3921 ***
DLDE-BERT	0.4708 ***	0.0291 ***	0.7225 ***	0.0214 ***	0.4262 ***	0.3969 ***
DLDE-FastText	0.4811 ***	0.0297 ***	0.7152 ***	0.0220 ***	0.4407 ***	0.3887 ***

Note: Asterisk in superscript denotes that a benchmark model performs significantly worse than the proposed model. Significance levels are: * $p < 0.1$, ** $p < 0.05$, *** $p < 0.01$.

A9.2.2 Robustness Checks on Challenge Selection Dataset

In the main analysis, we consider the challenges that has at least one user selection, as the never-selected challenges do not provide much information on users' preference variations. As a robustness check, we include the challenges that are never selected by users to re-run the experiment in Section 5.1. The results are provided in Table A7.

The results show that after adding the never-selected challenges, the performance of the collaborative-filtering-based models remain the same, whereas the performance of other benchmark models and our

proposed model deteroriate. The content-based model is impacted the most, followed by the hybrid models and the online learning models. This is expected, as the collaborative-filtering-based models are purely based on users' selection histories, whereas the other models do not rely entirely on the histories and may choose the items that are never selected. Despite these changes, our conclusions regarding the advantages of the proposed method still hold.

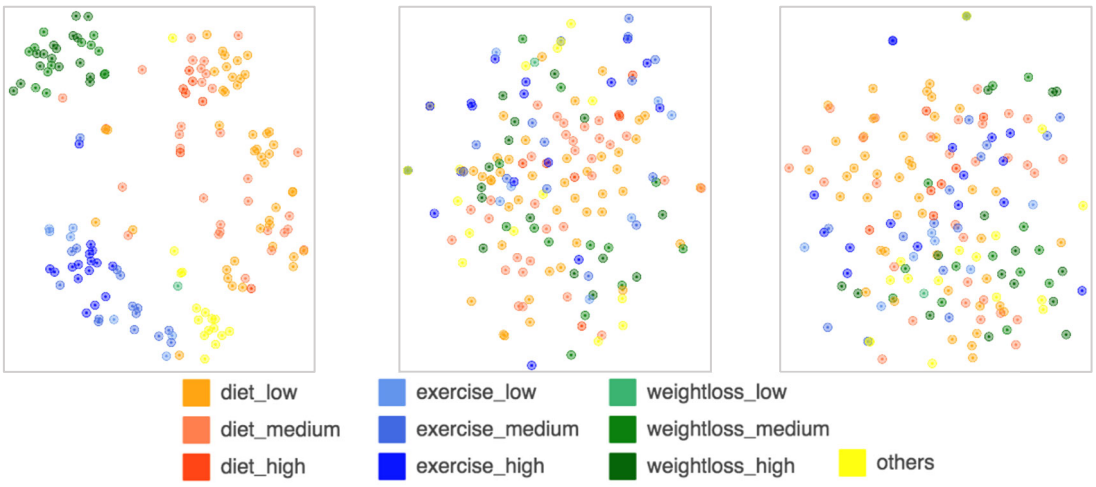
Table A7 Robustness Check on Challenge Selection Dataset

Model	Precision@10	Recall@10	nDCG@10	MAP@10	DR@10	Simu@10
DLDE-MAB	**0.5081**	**0.0647**	**0.7902**	**0.0418**	**0.4942**	**0.4445**
FAST	0.4014 ***	0.0501 ***	0.6104 ***	0.0291 ***	0.3796 ***	0.3850 ***
SLi_Rec	0.3959 ***	0.0457 ***	0.2356 ***	0.0245 ***	0.3913 ***	0.3988 ***
Caser	0.4006 ***	0.0463 ***	0.2015 ***	0.0255 ***	0.4066 ***	0.4016 ***
GRU4Rec	0.3937 ***	0.0454 ***	0.2290 ***	0.0252 ***	0.4004 ***	0.3995 ***
A2SVD	0.3921 ***	0.0453 ***	0.1961 ***	0.0255 ***	0.3953 ***	0.3973 ***
NextItNet	0.3892 ***	0.0449 ***	0.2292 ***	0.0251 ***	0.4034 ***	0.3886 ***
LSTUR	0.4396 ***	0.0539 ***	0.2337 ***	0.0347 ***	0.4247 ***	0.4209 ***
NPA	0.4053 ***	0.0509 ***	0.2062 ***	0.0322 ***	0.4334 ***	0.4393 **
CACF	0.3149 ***	0.0412 ***	0.5043 ***	0.0228 ***	0.4272 ***	0.4324 **
SCF	0.3970 ***	0.0495 ***	0.7026 ***	0.0305 ***	0.3977 ***	0.4024 ***
PMF	0.2147 ***	0.0269 ***	0.5183 ***	0.0120 ***	0.3637 ***	0.4010 ***
CAMF	0.2653 ***	0.0342 ***	0.6718 ***	0.0210 ***	0.3365 ***	0.3670 ***
CB	0.1690 ***	0.0209 ***	0.5173 ***	0.0075 ***	0.4002 ***	0.4324 ***
hybrid_pure	0.1712 ***	0.0214 ***	0.6268 ***	0.0109 ***	0.3919 ***	0.4283 ***
hybrid_cacf	0.2062 ***	0.0263 ***	0.4879 ***	0.0118 ***	0.4071 ***	0.4345 ***
UCB	0.2920 ***	0.0427 ***	0.7253 ***	0.0259 ***	0.3904 ***	0.4245 ***
ε-greedy	0.3823 ***	0.0579 ***	0.7634 ***	0.0324 ***	0.4113 ***	0.4180 ***

Note: Asterisk in superscript denotes that a benchmark model performs significantly worse than the proposed model. Significance levels are: $^{*}p < 0.1$, $^{**}p < 0.05$, $^{***}p < 0.01$.

A9.3 T-SNE Visualization of Benchmark Embeddings

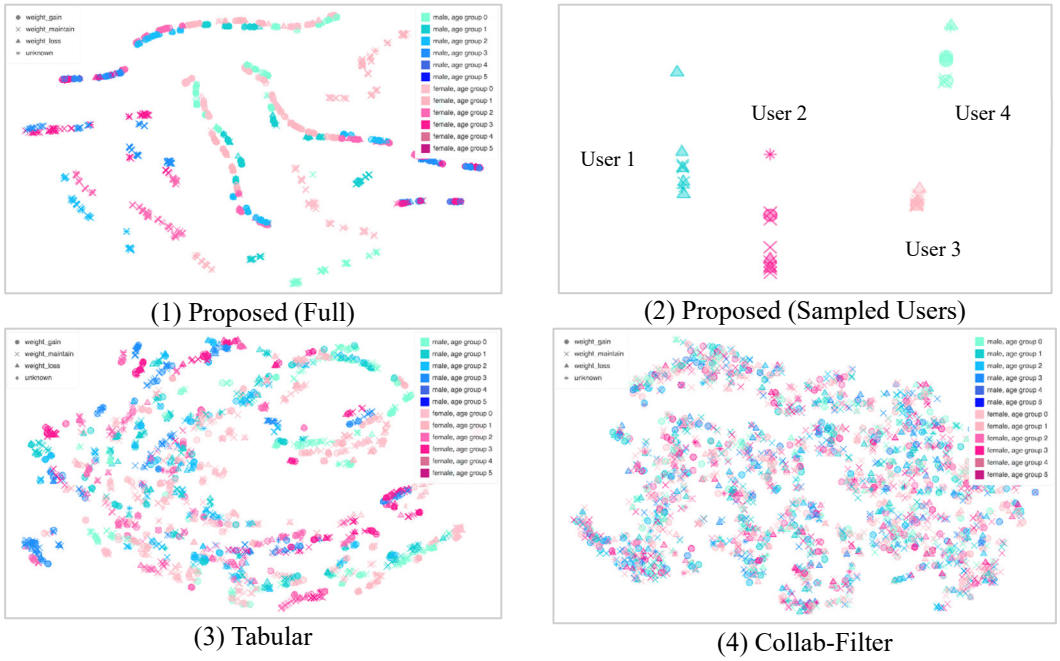
In this section, we show the t-SNE visualization results for the benchmark embeddings to provide a comparison with our proposed deep context representation. In Figure A5, we show the t-SNE plots for our constructed challenge embeddings, the BERT embeddings, and the FastText embeddings. Similar to what we have done in the main text, we use different color codes to distinguish challenge types and intensities so that the data patterns can be shown clearly. As can be seen, the visualization for the benchmark embeddings does not contain a clear clustering pattern, indicating that the benchmark models are less effective in capturing the key intervention attributes.



(1) Proposed (2) BERT (3) FastText

Figure A5 Challenge Embedding t-SNE Visualization

Figure A6 shows the comparison results for our proposed user embeddings and the embeddings constructed from Collab_Filter and Tabular model. We use the same color and marker code described in the paper to represent user characteristics. The results show that as compared to our proposed user representation, the Collab_Filter model and the Tabular model do not effectively distinguigh user patterns. These results are in line with our evaluation results.



(1) Proposed (Full) (2) Proposed (Sampled Users)

(3) Tabular (4) Collab-Filter

Figure A6 User Embedding t-SNE Visualization

A9.4 Evaluation Results for Dynamic Users

Table A8 Results for Dynamic Test Users (Top-5 Recommendations)

Model	Precision@5	Recall@5	nDCG@5	MAP@5	DR@5	Simu@5
DLDE-MAB	**0.5728**	**0.0329**	**0.7645**	**0.0263**	**0.5210**	**0.4390**
FAST	0.3867 ***	0.0246 ***	0.5422 ***	0.0126 ***	0.3733 ***	0.2800 ***
SLi_Rec	0.3256 ***	0.0213 ***	0.1585 ***	0.0151 ***	0.3396 ***	0.3438 ***
Caser	0.3311 ***	0.0217 ***	0.1284 ***	0.0154 ***	0.3429 ***	0.3319 ***
GRU4Rec	0.3317 ***	0.0218 ***	0.1543 ***	0.0155 ***	0.3276 ***	0.3281 ***
A2SVD	0.3350 ***	0.0220 ***	0.1204 ***	0.0155 ***	0.3467 ***	0.3256 ***
NextItNet	0.3283 ***	0.0215 ***	0.1583 ***	0.0151 ***	0.3422 ***	0.3194 ***
LSTUR	0.4133 ***	0.0267 ***	0.2080 ***	0.0244 ***	0.3667 ***	0.3489 ***
NPA	0.3578 ***	0.0234 ***	0.0952 ***	0.0241 ***	0.3756 ***	0.3444 ***
CACF	0.3239 ***	0.0215 ***	0.5365 ***	0.0157 ***	0.4278 ***	0.4344 *
SCF	0.4217 ***	0.0272 ***	0.6934 ***	0.0194 ***	0.4033 ***	0.4106 ***
PMF	0.1772 ***	0.0117 ***	0.4207 ***	0.0067 ***	0.3583 ***	0.3917 ***
CAMF	0.3067 ***	0.0204 ***	0.6322 ***	0.0150 ***	0.3539 ***	0.3822 ***
CB	0.2067 ***	0.0132 ***	0.3963 ***	0.0070 ***	0.3783 ***	0.4244 **
hybrid_pure	0.3067 ***	0.0208 ***	0.6564 ***	0.0132 ***	0.3606 ***	0.4373
hybrid_cacf	0.2606 ***	0.0176 ***	0.4920 ***	0.0110 ***	0.4189 ***	0.4367
UCB	0.3444 ***	0.0226 ***	0.6715 ***	0.0205 ***	0.4320 ***	0.4105 ***
ε-greedy	0.4542 ***	0.0291 ***	0.6918 ***	0.0212 ***	0.4993 **	0.4353

Note: Asterisk in superscript denotes that a benchmark model performs significantly worse than the proposed model. Significance levels are: * $p < 0.1$, ** $p < 0.05$, *** $p < 0.01$.

Table A9 Results for Dynamic Test Users (Top-10 Recommendations)

Model	Precision@10	Recall@10	nDCG@10	MAP@10	DR@10	Simu@10
DLDE-MAB	**0.5435**	**0.0617**	**0.8076**	**0.0413**	**0.5080**	**0.4645**
FAST	0.3800 ***	0.0496 ***	0.6203 ***	0.0239 ***	0.3767 ***	0.3533 ***
SLi_Rec	0.3514 ***	0.0461 ***	0.2289 ***	0.0228 ***	0.3530 ***	0.3409 ***
Caser	0.3555 ***	0.0468 ***	0.1993 ***	0.0234 ***	0.3385 ***	0.3418 ***
GRU4Rec	0.3516 ***	0.0462 ***	0.2300 ***	0.0232 ***	0.3261 ***	0.3571 ***
A2SVD	0.3508 ***	0.0459 ***	0.1938 ***	0.0231 ***	0.3294 ***	0.3384 ***
NextItNet	0.3472 ***	0.0457 ***	0.2345 ***	0.0225 ***	0.3363 ***	0.3400 ***
LSTUR	0.3981 ***	0.0529 ***	0.2870 ***	0.0407	0.3717 ***	0.3586 ***
NPA	0.3328 ***	0.0444 ***	0.1530 ***	0.0395 ***	0.3725 ***	0.3647 ***
CACF	0.2989 ***	0.0408 ***	0.5398 ***	0.0231 ***	0.4256 ***	0.4372 ***
SCF	0.3658 ***	0.0471 ***	0.6893 ***	0.0273 ***	0.3897 ***	0.4144 ***
PMF	0.1981 ***	0.0257 ***	0.5122 ***	0.0109 ***	0.3528 ***	0.3828 ***
CAMF	0.2400 ***	0.0324 ***	0.6528 ***	0.0199 ***	0.3289 ***	0.3794 ***
CB	0.2317 ***	0.0310 ***	0.4843 ***	0.0120 ***	0.3839 ***	0.4303 ***
hybrid_pure	0.2336 ***	0.0315 ***	0.6866 ***	0.0164 ***	0.4017 ***	0.4412 ***
hybrid_cacf	0.2639 ***	0.0354 ***	0.4967 ***	0.0173 ***	0.4094 ***	0.4369 ***
UCB	0.3516 ***	0.0463 ***	0.7171 ***	0.0276 ***	0.4036 ***	0.4225 ***
ε-greedy	0.4595 ***	0.0592 ***	0.7359 ***	0.0367 ***	0.4350 ***	0.4271 ***

Note: Asterisk in superscript denotes that a benchmark model performs significantly worse than the proposed model. Significance levels are: * $p < 0.1$, ** $p < 0.05$, *** $p < 0.01$.

References

Adomavicius G, Tuzhilin A (2005) Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering* 17(6):734-749.

--- (2011) Context-aware recommender systems. *Recommender systems handbook* (Springer), 217-253.

An M, Wu F, Wu C, Zhang K, Liu Z, Xie X (2019) Neural news recommendation with long-and short-term user representations. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 336-345.

Auer P, Cesa-Bianchi N, Fischer P (2002) Finite-time analysis of the multiarmed bandit problem. *Machine Learning* 47(2-3):235-256.

Baltrunas L, Ludwig B, Ricci F (2011) Matrix factorization techniques for context aware recommendation. *Proceedings of the Fifth ACM Conference on Recommender Systems*:301-304.

Barkan O, Koenigstein N (2016) Item2vec: neural item embedding for collaborative filtering. *2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP)* (IEEE), 1-6.

Beygelzimer A, Langford J, Li L, Reyzin L, Schapire R (2011) Contextual bandit algorithms with supervised learning guarantees. *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics* (JMLR Workshop and Conference Proceedings), 19-26.

Bieliková M, Kompan M, Zeleník D (2012) Effective hierarchical vector-based news representation for personalized recommendation. *Computer Science and Information Systems* 9(1):303-322.

Burke R (2002) Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction* 12(4):331-370.

Chen A (2005) Context-aware collaborative filtering system: Predicting the user's preference in the ubiquitous computing environment. *International Symposium on Location-and Context-Awareness* (Springer), 244-253.

Cohen JD, McClure SM, Yu AJ (2007) Should I stay or should I go? How the human brain manages the trade-off between exploitation and exploration. *Philosophical Transactions of the Royal Society B: Biological Sciences* 362(1481):933-942.

Dharia S, Jain V, Patel J, Vora J, Yamauchi R, Eirinaki M, Varlamis I (2016) PRO-Fit: Exercise with friends. *2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)* (IEEE), 1430-1433.

Donkers T, Loepp B, Ziegler J (2017) Sequential user-based recurrent neural network recommendations. *Proceedings of the eleventh ACM conference on recommender systems*, 152-160.

Dudík M, Langford J, Li L (2011) Doubly robust policy evaluation and learning. *arXiv preprint arXiv:1103.4601*.

Eskandanian F, Mobasher B (2018) Detecting changes in user preferences using hidden markov models for sequential recommendation tasks. *arXiv preprint arXiv:1810.00272*.

Gittins JC (1979) Bandit processes and dynamic allocation indices. *Journal of the Royal Statistical Society: Series B (Methodological)* 41(2):148-164.

Hertz U, Bahrami B, Keramati M (2018) Stochastic satisficing account of confidence in uncertain value-based decisions. *PLOS One* 13(4):e0195399.

Hidasi B, Karatzoglou A, Baltrunas L, Tikk D (2016) Session-based recommendations with recurrent neural networks. *ICLR*.

Howard J, Gugger S (2020) Fastai: a layered API for deep learning. *Information* 11(2):108.

Jiang N, Krishnamurthy A, Agarwal A, Langford J, Schapire RE (2017) Contextual decision processes with low bellman rank are pac-learnable. *International Conference on Machine Learning* (PMLR), 1704-1713.

Johnson PE, Veazie PJ, Kochevar L, O'connor PJ, Potthoff SJ, Verma D, Dutta P (2002) Understanding variation in chronic disease outcomes. *Health Care Management Science* 5(3):175-189.

Kim J-H, Lee J-H, Park J-S, Lee Y-H, Rim K-W (2009) Design of diet recommendation system for healthcare service based on user information. *2009 Fourth International Conference on Computer Sciences and Convergence Information Technology* (IEEE), 516-518.

Kim MJ, Lim AE (2015) Robust multiarmed bandit problems. *Management Science* 62(1):264-285.

Lattimore T, Szepesvári C (2020) *Bandit algorithms* (Cambridge University Press).

Lei H, Tewari A, Murphy SA (2017) An actor-critic contextual bandit algorithm for personalized mobile health interventions. *arXiv preprint arXiv:1706.09090*.

Li L, Chu W, Langford J, Schapire RE (2010) A contextual-bandit approach to personalized news article recommendation. *Proceedings of the 19th International Conference on World Wide Web* (ACM), 661-670.

McLean V (2011) Motivating patients to use smartphone health apps. *PR Web* 113.

Mehlhorn K, Newell BR, Todd PM, Lee MD, Morgan K, Braithwaite VA, Hausmann D, Fiedler K, Gonzalez C (2015) Unpacking the exploration–exploitation tradeoff: A synthesis of human and animal literatures. *Decision* 2(3):191.

Misra K, Schwartz EM, Abernethy J (2019) Dynamic Online Pricing with Incomplete Information Using Multiarmed Bandit Experiments. *Marketing Science*.

Mnih A, Salakhutdinov RR (2008) Probabilistic matrix factorization. *Advances in Neural Information Processing Systems*:1257-1264.

Mnih V, Kavukcuoglu K, Silver D, Rusu AA, Veness J, Bellemare MG, Graves A, Riedmiller M, Fidjeland AK, Ostrovski G (2015) Human-level control through deep reinforcement learning. *nature* 518(7540):529-533.

Morid MA, Sheng ORL, Dunbar J (2021) Time Series Prediction using Deep Learning Methods in Healthcare. *ACM Transactions on Management Information Systems*.

Phanich M, Pholkul P, Phimoltares S (2010) Food recommendation system using clustering analysis for diabetic patients. *2010 International Conference on Information Science and Applications* (IEEE), 1-8.

Pon RK, Cardenas AF, Buttler D, Critchlow T (2007) Tracking multiple topics for finding interesting articles. *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 560-569.

Sahoo N, Singh PV, Mukhopadhyay T (2012) A hidden Markov model for collaborative filtering. *MIS Quarterly*:1329-1356.

Sani A, Lazaric A, Munos R (2012) Risk-aversion in multi-armed bandits. *Advances in Neural Information Processing Systems*:3275-3283.

Sedhain S, Sanner S, Braziunas D, Xie L, Christensen J (2014) Social collaborative filtering for cold-start recommendations. *Proceedings of the 8th ACM Conference on Recommender Systems*:345-348.

Silver D, Huang A, Maddison CJ, Guez A, Sifre L, Van Den Driessche G, Schrittwieser J, Antonoglou I, Panneershelvam V, Lanctot M (2016) Mastering the game of Go with deep neural networks and tree search. *nature* 529(7587):484-489.

Speekenbrink M, Konstantinidis E (2015) Uncertainty and exploration in a restless bandit problem. *Topics in Cognitive Science* 7(2):351-367.

Subramaniyaswamy V, Manogaran G, Logesh R, Vijayakumar V, Chilamkurti N, Malathi D, Senthilselvan N (2019) An ontology-driven personalized food recommendation in IoT-based healthcare system. *The Journal of Supercomputing* 75(6):3184-3216.

Sutton RS, Barto AG (2018) *Reinforcement learning: An introduction* (MIT press).

Tang J, Wang K (2018) Personalized top-n sequential recommendation via convolutional sequence embedding. *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, 565-573.

Tang L, Jiang Y, Li L, Li T (2014) Ensemble contextual bandits for personalized recommendation. *Proceedings of the 8th ACM Conference on Recommender Systems* (ACM), 73-80.

Tewari A, Murphy SA (2017) From ads to interventions: Contextual bandits in mobile health. *Mobile Health* (Springer), 495-517.

Tomkins S, Liao P, Klasnja P, Murphy S (2021) IntelligentPooling: practical Thompson sampling for mHealth. *Machine Learning*:1-43.

Unger M, Bar A, Shapira B, Rokach L (2016) Towards latent context-aware recommendation systems. *Knowledge-Based Systems* 104:165-178.

Vasile F, Smirnova E, Conneau A (2016) Meta-prod2vec: Product embeddings using side-information for recommendation. *Proceedings of the 10th ACM Conference on Recommender Systems*, 225-232.

Wendel S, Dellaert BG, Ronteltap A, Van Trijp HC (2013) Consumers' intention to use health recommendation systems to receive personalized nutrition advice. *BMC health services research* 13(1):1-13.

Wu C, Wu F, An M, Huang J, Huang Y, Xie X (2019) NPA: neural news recommendation with personalized attention. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2576-2584.

Yu Z, Lian J, Mahmoody A, Liu G, Xie X (2019) Adaptive User Modeling with Long and Short-Term Preferences for Personalized Recommendation. *IJCAI*, 4213-4219.

Yuan F, Karatzoglou A, Arapakis I, Jose JM, He X (2019) A simple convolutional generative network for next item recommendation. *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, 582-590.

Zaman N, Li J (2014) Semantics-enhanced recommendation system for social healthcare. *2014 IEEE 28th International Conference on Advanced Information Networking and Applications* (IEEE), 765-770.

Zhang S, Yao L, Sun A, Tay Y (2019) Deep learning based recommender system: A survey and new perspectives. *ACM Computing Surveys (CSUR)* 52(1):1-38.

Zhao X, Chen W, Yang F, Liu Z (2016) Improving diversity of user-based two-step recommendation algorithm with popularity normalization. *International Conference on Database Systems for Advanced Applications* (Springer), 15-26.

Zhou Z, Wang Y, Mamani H, Coffey DG (2019) How do tumor cytogenetics inform cancer treatments? dynamic risk stratification and precision medicine using multi-armed bandits. *Dynamic Risk Stratification and Precision Medicine Using Multi-armed Bandits (June 17, 2019)*.