

Fairness Embedded Adaptive Recommender System: A Conceptual Framework

Alina Popa

The Doctoral School of Marketing
The Bucharest Academy of Economic Studies, Bucharest, Romania

Abstract—In the current fast paced and constantly changing environment, companies should ensure that their way of interacting with user is both relevant and highly adaptive. In order to stay competitive, companies should invest in state-of-the-art technologies that optimize the relationship with the user using increasingly available data. The most popular applications used to develop user relationship are Recommender Systems. The vast majority of the traditional recommender system considers recommendation as a static procedure and focus on a specific type of recommendation, being not very agile in adapting to new situations. Also, when implementing a Recommender System there is the need to ensure fairness in the way decisions are made upon customer data. In this paper, it is proposed a novel Reinforcement Learning-based recommender system that is highly adaptive to changes in customer behavior and focuses on ensuring both producer and consumer fairness, Fairness Embedded Adaptive Recommender System (FEARS). The approach overcomes Reinforcement Learning's main drawback in recommendation area by using a small, but meaningful action space. Also, there are presented two fairness metrics, their calculation and adaptation for usage with Reinforcement Learning, this way ensuring that the system gets to the optimal trade-off between personalization and fairness.

Keywords—Algorithmic fairness; reinforcement learning; recommender systems; system adaptability

I. INTRODUCTION

In the current constantly changing business environment, companies are required to respond appropriately to challenges that appear and adapt quickly to customers new needs and expectations in order to stay “top of mind” with prospects and clients. In retailing, a family of applications called Recommender Systems (RecSys) can help businesses stay relevant to their customers by leveraging the existing data about users and/or different items in order to help users find the right item for them [1]. One disadvantage of the current approaches of RecSys like Collaborative filtering or Content-based recommendations is that these strategies consider only the two elements, users and items when delivering recommendations, making impossible to detect important patterns that include other elements and to adapt it to the context or changing environment. Also, each of the recommendation approaches has its own limitations. Items recommended through Content-based filtering are always similar to the items previously bought or consumed by the user [2], while Collaborative filtering provides a good solution only under static scenarios when there are many users that bought or consumed the same product [3]. Hybrid

recommender systems combine two or more recommendation strategies in different ways to benefit from their complementary advantages [4] and overcome the limitations of individual components. Another limitation of RecSys, regardless of strategy is the assumption that user's underlying preferences remains unchanged, thus the recommendation procedure is a static process [5, 6].

One of the best-known approaches that allows to include adaptability in a system is Reinforcement Learning (RL) [7-9]. There is a series of publications that explore the usage of RL in the area of RecSys. Out of which there are those that focus on user-item interaction sequence or user's browsing history and use it to create a state that later is fed to the RL model [10-15]. A different approach is to use user and item sets which are obtained from bi-clustering as environmental states [6]. An earlier paper is using both user information and item information vectors and refers to it as context [16]. Important work on integrating negative influence of irrelevant recommendations is done by using negative rewards [12, 13, 15, 17].

Machine Learning technologies are increasing their presence in our daily lives and have serious implication and influence in a lot of areas. Because these systems are used in many sensitive situations and can lead to life-changing, high stake decisions, it is crucial that algorithms' outputs do not reflect discriminatory behavior. Which is why, alongside with developing new ML use-cases, researchers have done efforts to formalize fairness and sources of bias in ML [18-27], propose fairness evaluation metrics [18, 19, 28-33] and processing methods to mitigate undesirable biases in relation to the proposed definitions [19, 28, 31, 34-37].

It is worth noting that based on literature review, there are no works as per knowledge of the author that explore simultaneously the following elements of an efficient and fair recommender system: 1) focus on customer relationship development 2) adaptability to new situations 3) optimization for long term customer engagement using negative rewards where appropriate and 4) awareness of both consumer and provider fairness.

Thus, in this paper, it is presented the design of a Fairness Embedded Adaptive Recommender System (FEARS) that has its main aim the development of customer relationship. The conceptual framework is combining multiple recommendation strategies through leveraging extensive information about user, items and context. The recommendation strategies are combined and used as the action space by a RL Agent. This

way, the system has the ability to automatically learn the optimal policy through trial-and-error; by recommending and receiving reinforcements from user's feedback. This will allow the system to quickly adapt to the changing needs of the customers and will try to come up with a more long-term recommendation strategy to build a fruitful relationship with the client. Not least, the rewards of the RL engine are defined in such a way that both consumer and provider fairness is being ensured.

II. RELATED WORK

In this section, firstly it is described the basic problem of Recommender Systems. Next, each of the recommender strategies specifics, latest advances and limitation are presented. Then, it is introduced the Reinforcement Learning practice and an analysis of it's to date usage and limitations in the recommendation area.

Also, an exploration of the different fairness formalizations for ML and recommendation systems is done in order to give an overview on how these definitions are translated into implementation.

A. Recommender Systems (RecSys)

In the human decision-making process, obtaining recommendations from trusted sources is a critical component. Usually, this role is played by family, friends or subject-matter experts. The goal of a recommender system is to create and give relevant recommendations of items or products to users. Depending on the structure of the learning system, traditionally there are distinguished following types of systems[4]:

- Collaborative Filtering: In this type of systems, a user is recommended items based on the previous ratings of the users that bought/ used the product.
- Content-based Filtering: These systems recommend items that are similar to items the user has liked in the past.
- Hybrid approaches: These methods try to combine both collaborative and content-based approaches into one in order to overcome the individual limitations of each of the approaches.

Currently, the architecture of RecSys and their evaluation on real-world problems is an active area of research.

B. Collaborative Filtering

Collaborative filtering (CF) systems collect user feedback in the form of ratings or ranks and makes recommendations to the active user based on items that other users with similar preferences liked in the past [38].

The aim of any recommendation system is to suggest elements that are relevant to users by extracting latent variables [39-42].

Latest advances in the field include using graph encoding, Stochastic Shared Embeddings, large-scale Pairwise Collaborative Ranking, Sequential Recommendation Via Personalized Transformer [43] that mainly solve the problem of scaling to massive datasets, learn user and item embeddings

and think about the problem as a sequence of actions, not one-shot recommendations. The user and items embeddings are mainly a different way to refer to latent variables and a series of work leverage the power of Neural Networks to try and learn them [44-46].

C. Content-Based Filtering

The intuition behind a content-based recommendation is to suggest to a customer a product similar to those the user has previously purchased. Method tries to extract similar objects. There are two main types of measures used to estimate this relationship: measures of distance and measures of similarity between objects [47].

Most of the advances in the content-based recommendation area is based on finding best ways to represent an item through a vector, or, in other words, get their embeddings [48-53].

D. Recommendation of Complementary Products

When recommending complementary products, the system tries to leverage the transactions history of customers [54, 55] through Association Methods like APRIORI.

The APRIORI algorithm remained essentially unchanged since its introduction to the research community, although there are sporadic efforts to extend it [56, 57].

E. Reinforcement Learning (RL)

Reinforcement Learning is an area of machine learning that has been inspired by behavioural psychology. The field focuses on how a software agent (hereinafter agent) should take actions and how to interact with an environment so as to maximize a total reward function.

An agent can interact with the environment and learn through trial and error, just like humans and animals. Every action that the agent performs in an environment influences the future state of the agent. Also, each action is rewarded with a reward, and this is the only response the learner receives [58]. The mechanism that generates the reward and the transition from one state of the agent to another refers to the dynamics of the environment [47].

The agent's goal is to maximize his total long-term reward in the way he responds to his environment. This can happen if an agent explores the environment and tries to learn its dynamics.

Formally, the environment is a mathematical model known as the Markov Decision Process (MDP) encountered primarily in dynamic programming. The difference between the classical methods of dynamic control and RL is that the latter does not know the MDP model and can be used if these processes are very complex and other methods are unfeasible [59]. The basic MDP model contains the following components:

- A set of environmental states $S_1, \dots, S_n \in S$: These can refer to the inherent characteristics of the agent or objects that surround and interact with it.

- A set of actions that the agent can take, $A_1, \dots, A_m \in A$: These refer to all possible actions that the agent controls.
- Transition function from one state to another: Being a Markov process, the next state of the system depends only on its previous state and the action taken, not on the whole history.
- The reward function represents the value of the reward obtained after acting with A_t in S_t .

An agent is a computer program that is able to observe and interact with the environment defined by the MDP. The agent perceives the environment as a set of observations that define a state. The agent interacts with the environment in a feedback loop pattern by following the steps below:

- 1) The agent observes the characteristics of the environment that define the current state, S_t .
- 2) The agent chooses an action from the set of possible actions, A_t , with which it responds to the environment in the current state S_t .
- 3) The agent enters a waiting state until the characteristics of the environment change with the $S_t + 1$ state and the agent receives the R_{t+1} reward.
- 4) Steps 1-3 are then repeated.

The agent's behaviour or the way he interacts with the environment is described by a function called action policy or simply policy [60]. It specifies the actions to be taken when the agent is in a certain state. The agent's learning goal is to find a policy that maximizes the total reward.

F. Recommender Systems using Reinforcement Learning

As previously mentioned, in the literature there are already RecSys that include an RL engine. It is useful to formalize the problem of RL in the RecSys area and see the differences in the approach of the different research.

As mentioned above, formally, the RL problem can be defined as a mathematical MDP model. For that it is needed to specify the States, Actions and Rewards.

States are defined in different ways in the existing literature. They can reflect a mapping of previous user-item interactions into a hidden state [15], user's recommendation and ad browsing history [13], previous items that a user clicked [12], the sequence of visited and recommended items [10] or a more detailed interaction sequence that contains clicking, purchasing, or skipping, leaving [14]. An interesting approach is to define states as the cluster resulted from the co-clustering or biclustering of users and items [6] or to extend the state to include user demographics [5]. Efforts are as well invested in how to best represent the state in a RL RecSys [61]. Currently, and as to the knowledge of the author, in the current literature there is no approach where the recommendation context, user demographics, behavioral patterns and recent browsing/interaction history is taken into account in the state definition.

Actions are mostly defined as selecting an item to be recommended from the whole discrete action space which

contains the candidate items [12, 14, 15] or even whether to give a recommendation or not, and if yes, what would be the item to recommend [13]. There are authors that consider recommending a list of items [5, 11, 61]. One of the most different approaches it to recommend items from neighboring clusters to the user-items one [6]. As mentioned in multiple articles [62, 63], RL in RecSys has a common issue of efficiency that comes from the fact that the action space is too large, consisting of all candidate items, and thus huge amount of interaction data is required for learning an optimal policy.

The reward function is heavily dependent on user feedback and actions he takes, for example user can click or purchase a recommended item and receive a positive reward or to skip it and get a different reward value [10-13, 15]. Reward can consist of immediate user feedback, but as well as a longer-term objective [14]. Most of the rewards are not deterministic and depend very much on how the user is reacting, but there are also formulations when this is seen deterministically as the Jaccard distance between the user vectors of the time t and $t+1$ state [6].

It is important to note the research direction as well towards using negative rewards. This can help the learning agent into searching for a policy that would be appropriate for overcoming the information fatigue [12, 13, 15, 17].

G. Fairness in Machine Learning

In the same way as people, algorithms are vulnerable to biases that exist in data and can lead to an unfair decision or outcome. More than 20 types of biases in ML were extracted, categorized and explained by researchers [24, 26] in order to motivate and accelerate the process of mitigating them.

Putted simple, in the context of high stakes decision-making, fairness is the absence of any prejudice or favoritism towards an individual or a group based on their inherent or acquired characteristics that are considered sensitive variables. Thus, a fair algorithm is one whose decisions are not skewed towards a particular group of people. GDPR, UK Equality Act, Fair Housing Act and Equal Credit Opportunity Act define protected classes such as race, gender, age or disability and state the fairness and equality principles [64].

The most simple and straightforward definition of fairness is "fairness through unawareness": "A ML model is said to achieve fairness through unawareness if protected attributes are not explicitly used in the prediction process".

Although these variables are not used in developing the ML model, this doesn't mean that the information cannot be retrieved from other variables. Chiappa & Isaac [65] emphasize that fairness should be expressed both in terms of sensitive variables, but also considering correlated or proxy variables. Not considering these proxy variables has been shown to increase the risk of discrimination [27].

Mehrabi et al. [66] distinguish three different types of fairness definitions: 1) Individual Fairness where the system should give similar outputs to similar individuals, 2) Group Fairness where ML system treats different groups equally and 3) Subgroup Fairness which intends to obtain the best properties of the group and individual notions of fairness.

Group fairness equal treatment can be in turn defined through [66, 37]: 1) Equal Opportunity where the probability of a person from a positive outcome class of being assigned a positive outcome should be equal for both protected and unprotected group members, 2) Demographic Parity where the likelihood of a positive outcome should be the same regardless of whether the person is in the protected group or not, 3) Disparate Impact considers the ratio between unprivileged and privileged groups likelihood of a positive outcome. Disparate impact uses the “not less than 80%” rule to define if a process has disparate impact or not.

Main approaches for tackling unfairness are differentiated into three groups: 1) pre-processing, 2) in-processing and 3) post-processing [37]. Pre-processing methods are extracting representations from the data in order to remove undesired biases [27]. Then, this unbiased data is used for model development. Some of the methods in this family are adversarial learning, causal methods, relabeling, perturbations, resampling, reweighing, transformation and variable blinding [27]. The in-processing methods are constraining a model to produce fair outputs by including fairness into the learning mechanism like adversarial learning, bandits, constraint optimization, regularization or reweighing [27]. The post-processing methods are working with model outputs to make them fair using calibration, thresholding and transformation approaches [27].

H. Fairness in Recommender Systems

As presented in a previous section, RecSys make recommendations to support decision making by studying user behavior and historical patterns. Because it is widely applied in various fields like recommending music, books, people to hire or jobs, an impartial view of the system towards any of the involved sides can be detrimental [67, 68].

Since these systems use past data, they are also inheriting the 1) Historical Bias, which is the already existing bias in the world [26], 2) Representation Bias when the used sample from a population is not representative for the whole population [26] and 3) Social Bias when other people’s actions or generated content affect another person’s opinions [69]. Alongside with these biases, the system itself displays: 1) Popularity Bias when items that are more popular tend to be exposed more [70], 2) Algorithmic Bias when the bias is not present in the input data and is added later by the algorithm in the way it works [69], 3) Presentation Bias when the way items are presented impacts the attraction those items get (e.g., users can click only what they see, thus, items presented more often will get more clicks) [69] and 4) Ranking Bias when top ranked items are perceived as more interesting and thus, receive more traffic [71]. In Fig. 1 it is shown how different types of biases feed each other in a RecSys.

Considerations of fairness have been actively studied in the context of recommender systems. Burke [72] introduced the multisided view of fairness in recommender systems. In the case of recommendations, the system is facilitating a transaction between parties [73]. Fairness towards all the involved parties is important and a balanced point should be found. Burke et al. [74] divide stakeholders of any given recommender system into three categories: 1) Consumers are

the individuals that receive recommendations 2) Providers are those that stay behind the recommended items or products and gain from consumer’s choices and 3) System is the platform itself that tries to match providers with consumers and by doing this in a successful way is gaining benefits.

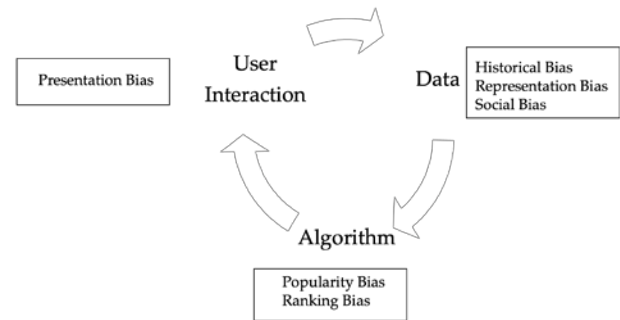


Fig. 1. Different Types of Biases that Appear in a Recommender System.
Source: Adaptation after Fig. 2 [115].

Recommender system’s objective for a consumer is to give the best items for his needs, through personalization, in such a way that these items are not constraining him in getting a higher overall utility compared to people from other groups, thus in a fair way. For a provider, recommender system needs to ensure that his items get sufficient exposure and that items are shown to the consumers that have the highest probability of buying or consuming them, thus in a relevant way. Platform’s utility is also important, because this is the initial motivation of having the recommender system in place. A key issue that arises in recommender systems is the tension between a personalized view of recommendation delivery and fairness objectives [74, 75].

Provider’s fairness in recommender systems is typically defined for the objects or subjects to be ranked. It has been explored and formalized as:

- 1) the bound of the number of items related to each of the protected attributes that are allowed to appear in the top k positions of the ranking [76],
- 2) a sufficient presence of items belonging to different groups [77],
- 3) a consistent treatment of similar items [77],
- 4) a proper representation of items from both protected and unprotected groups [77].
- 5) exposure, disparate exposure and group fairness disparity; all three proportional to the merit of the item defined as relevance to the query [78],
- 6) pairwise fairness that expresses the likelihood of a clicked item being ranked above another relevant unclicked item is the same across both groups [79],
- 7) pairwise statistical parity represents that if two candidates from different groups are compared, then on average each group has an equal chance of being top ranked [79],
- 8) set-based fairness at discrete points in the ranking with logarithmic discount that emphasize the fact that fairness at top ranks is more important than at lower ranks [80],

9) difference in acceptance rates measures whether a relevant item from the advantaged and disadvantaged class are accepted at the same rates [81].

In the area of consumer fairness, there are the following metrics that can be used:

1) value unfairness which occurs when one group of users is consistently given higher or lower predictions than their true preferences [82]. Value unfairness becomes large when predictions for one group are consistently overestimated and predictions for the other group are underestimated.

2) absolute unfairness, which measures inconsistency in absolute estimation error across different user groups [82]. This means that the advantaged group has the unfair advantage of good recommendations, while the other groups have poor recommendations.

3) non-parity unfairness is computed as the absolute difference between the overall average ratings of disadvantaged users and those of advantaged users for recommended items [82].

4) balanced neighborhood that expresses the fact that recommendations for all users are generated from neighborhoods that are balanced with respect to the protected and unprotected classes [74].

5) disparate impact of recommendation [74, 37].

Overall, the consumer fairness is less represented in the literature.

There are efforts in the area of mitigating bias and ensuring fairness in recommender systems by using regularization terms [82, 83, 84], reinforcement learning [78, 85] and neighborhood balancing [74].

Although the literature of methods is rich in methods to mitigate unfairness, not all of them are applicable to the dynamic nature of recommender systems. Ge et al. [85] show that by enforcing fair decisions through static fairness criteria metrics, the system leads to unexpected unfairness in the long run and that fairness cannot be defined in a static setting without considering the long-term impact and evolution. Same as with the need to bring adaptability into the recommender system results, the need to have a dynamic view over fairness can be solved by using reinforcement learning.

The practice of using Reinforcement Learning to ensure fairness is an emerging research area [27]. In terms of implementation, fairness dimension can be given to the RL agent as a reward that can be positive in the case of fair outputs and negative otherwise [86, 87]. Other approaches construct the problem as a Constrained Markov Decision Process [85].

III. PROPOSED APPROACH

In this section, it is proposed a conceptual framework for the Fairness Embedded Adaptive Recommender System that aims to balance between personalization and fairness for long-term customer engagement. Firstly, the objectives of the recommendation system are introduced with possible solutions. Then, based on this, a novel architecture for this

type of systems is proposed. Next, it is described how fairness dimension can be introduced into the RL engine and how the personalization-fairness trade-off can be solved.

A. Fairness Embedded Adaptive Recommender System

In the present paper, the objective is to create a conceptual design of a recommender system that holds a series of requirements:

- System is focusing on customer relationship development.
- System is incorporating an adaptivity functionality.
- System is optimizing for long term customer engagement.
- System is ensuring consumer and provider fairness.
- System is using a small action and state space in the RL engine.
- System is solving the personalization and fairness trade-off.

B. System Overview

Once converted, the relationship with a new customer must be developed for it to become profitable. In simple terms, this means understanding and covering client's needs.

The objective of the application is to extract consumer preferences and use this knowledge to find the most appropriate products and / or content that will be recommended through communication and interaction with the customer. Same time, the recommended content should bring to user the maximum utility and give him equal opportunities compared to people from other social groups.

C. System Components

The framework used follows next steps: 1) Database creation 2) Preprocessing 3) Recommendations generation 4) Recommendation's combination 5) Reinforcement Learning Engine 6) User Recommendation 7) User Feedback incorporation.

1) *Database creation*: The application starts by setting up data sources (Fig. 2, 1). The information considered mandatory in a recommender system application is a) consumer data, b) their past interactions, c) data on provider's items characteristics, d) items' reviews and e) metadata about the current browsing session.

2) *Preprocessing*: The next step is to prepare the tables in the form in which they will be used in different recommendation components. This means that a series of tables having different structures will be created: a) The Customer-Item Matrix Table contains information about the items purchased or consumed by a consumer during the analysis period b) The Transaction-Item matrix is typically stored in transactional format where a transaction contains several rows, c) The items characteristics table contains all the tangible and intangible characteristics of an item as well as

statistics about how many times it was recommended, clicked, bought from recommendation lists etc.

3) *Recommendation Components*: Once all the main tables are prepared (Fig. 2, 2), three recommendation components are developed, and their results are merged and combined to create the action space for the RL Engine (Fig. 2, 3-4).

4) *User-oriented collaborative filtering recommendation* (Fig. 2, 3.1): The method starts from the assumption that similar users have similar preferences [88] and reflects the real situation when recommendations from friends are more effective. Model tries to explain the Customer-Item Matrix Table using a set of latent factors. Latent structures are automatically deduced from the matrix, as long as the number of factors is specified [88]. Once the factors are discovered, the model associates the belonging of an item to a factor and the user's inclination towards the same factor. For each customer, the model will recommend products that have values close to "1" in the reconstructed matrix and have not been purchased in the past.

5) *Content-based recommendation* (Fig. 2, 3.2): In the case of new users or items, because of no prior history, the recommendation strategy is using the Items characteristics table. For example, one can use all available information about the tangible and intangible properties of the item, including embeddings extracted from unstructured data.

6) *Complementary item recommendation* (Fig. 2, 3.3): In this step, rule sets are extracted from transactions using association algorithms. This is extremely useful as it emphasizes the context of using/consuming the initial item. This as well brings completeness to customer's need by saying "if you want to use this, do not forget about that".

Following these recommendation strategies, the outcomes are combined between them in order to create the action space for the RL model (Fig. 2, 4). As it can be seen, there is also the "Random Recommendation" component (Fig. 2, 3.4) and "No Recommendation" (Fig. 2, 3.5) that will bring exploration and novelty into the recommendation landscape as well as will keep the system from harming stakeholders through unfair decisions.

7) *Reinforcement Learning Engine*: The next task of the system is to choose the most appropriate action for a particular use. In other words, the question that needs to be answered is: "For this user, what is the best action to take? Recommendation of a product that corresponds to the latent structures of the user? An item similar to what user consumed/liked before or a complementary item?". The answer can be as well that the best action is no action.

The solution that could combine user information, past behavior and interaction in order to choose the best action is to use a RL engine (Fig. 2, 5). RL problem is defined as a MDP system.

The set of environmental states is represented by the finite clusters over the vector space extracted from the characteristics of the environment:

- The socio-demographic characteristics of the user.
- User past behavior and interaction with focus on indicators like diversification, appetite for novelty, previously liked items.
- The details of the period in which the browsing and recommendation is made. This can include time of the day or year, browsing device, browsing session time etc.

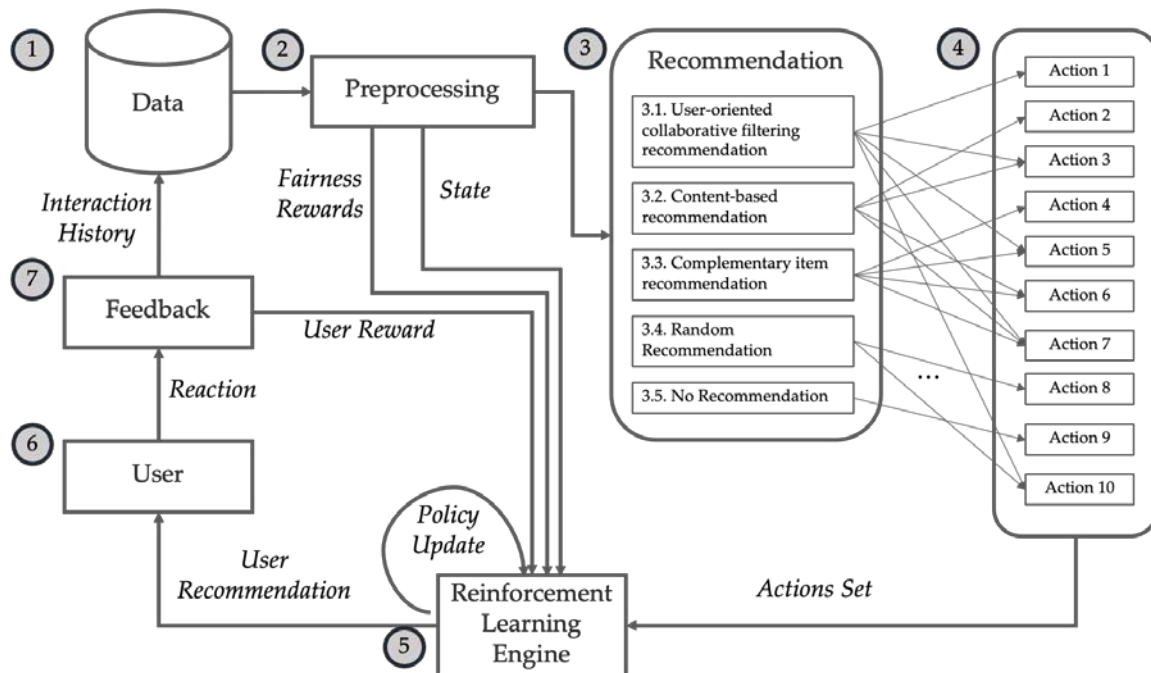


Fig. 2. Design of a Fairness Embedded Adaptive Recommender System.

The set of actions represents all possible actions that the recommendation system controls. As defined before, the set of actions is represented by the individual recommendations or combinations of them (Fig. 2, 4).

The advantages of formulating the actions like this are: 1) the low complexity given by action space: instead of having all the candidate items, 2) keep the personalization as a key focus, 3) include novelty for consumer and fairness for provider (random product action and no recommendation action).

The reward function is conditioned by the user's response to the recommendation received (Fig. 2, 6-7), but also by fairness rewards of the system.

The reward value at time $t+1$, R_{t+1} , after the agent takes A_t in S_t is compounded out of three terms:

- Reward coming from user response
- Reward coming from fairness value towards consumer
- Reward coming from fairness value towards provider

Next, the individual terms are defined, because they are the key for embedding fairness and consumer relationship in the system.

User response rewards are linked to the action that user is taking after seeing the recommended item. There can be distinguished the following actions that a user is taking in response, each with a different associated reward. The associated reward will be decreasing as per user feedback reaching a negative value if user ends the communication with company:

- User clicks on the recommended product, and buys/consumes it.
- User clicks on the recommended product, but does not buy/consume it.
- User is adding the item into wish list/buy latter/favorites lists.
- User marks the recommendation as inappropriate.
- User closes the recommendation/searching session without taking any other action.

Consumer fairness reward is linked to a particular item recommended and is known in advance; thus, it is deterministic. Consumer fairness reward is calculated for each item based on Disparate Impact formula as defined in [37]. This particular formula was used because it is easy to adapt and integrate as a reward into the system. Also, the metric encodes the demographical parity idea in a RL workable manner. But the main reason why this particular metric was chosen out of all presented previously is its unique advantage of being deterministic, thus known before taking any action. Considering the act of recommending a particular item being a treatment, this means that the ratio between the likelihood of a positive outcome (presentation of an item) for different groups should ideally be close to 1:

Disparate Impact(item)

$$= \frac{P(\text{recommend item}=1 \mid \text{group}=\text{protected group})}{P(\text{recommend item}=1 \mid \text{group}=\text{privileged group})} \quad (1)$$

$$= \frac{P(\text{recommend item}=1 \mid \text{group}=\text{protected group})}{P(\text{recommend item}=1 \mid \text{group}=\text{privileged group})} \quad (1)$$

The Disparate Impact for an item is updated every time an item is being recommended using equation 1, and although it is item linked, is actually expressing the consumer fairness. By integrating this metric, it is ensured that individuals with different backgrounds are treated the same, thus they receive same type of items and content recommendation. A good example of a situation that was not fair towards consumer is the study that shows that female users of Google had a lower chance of being recommended and presented hiring ads for high-paying executive jobs [67].

In order to integrate the Disparate Impact into the RL engine, first is needed to adapt it to express the undesirable practice of having either positive or negative bias. Although most of the metrics are focusing on supporting the unprivileged group, this can lead to a turn of the situation, such that the focus here is on not having any type of bias and treating individuals from different groups equally. The equation of calculating the consumer reward is given by equation 2.

$$R_{t+1} \text{ consumer fairness (item)} = -|\text{Disparate Impact(item)} - 1| \quad (2)$$

For the provider fairness, it is desired to ensure that exposure is a function of relevance. For this purpose, difference in acceptance rates is being used (DAR) [81] and is calculated as the pairwise difference of the ratio of true positives divided by the predicted positives for each class. The latest is also called Precision in binary classification evaluation [47] and reflects the fraction of relevant cases. The reason why it was decided to go with this particular metric, is the fact that it is the only metric out of the presented ones that can be used in other recommendation settings besides ranking.

By incorporating DAR into the RL engine there will be ensured that items are presented proportionally to how relevant they are.

$$\text{Relevance (item)} = P(y = 1 \mid \hat{y} = 1) \quad (3)$$

In order to actually use DAR as rewards, for each item, the relevance as defined in equation 3 is calculated. The updated relevance is simulating the situation on recommending an item, while it will not be clicked on. Secondly, DAR (equation 4) is calculated using the newly updated relevance for the item, while maintaining the relevancies for other items *ceteris paribus*. There will be a DAR value associated with each of the items that can potentially be recommended. Finally, this value is taken with a negative sign (equation 5) in order to count for provider fairness. If there are big differences between items relevance, they will sum up and bring to a high negative reward, thus the agent will try to choose an item that leads to a smaller DAR. The overall reward function is given by the equation 6.

$$Difference\ in\ acceptance\ rates\ (item) = \frac{\sum_{i=0}^l \sum_{j=0, j \neq i}^l dist(Relevance(item_i), Relevance(item_j))}{l(l-1)} \quad (4)$$

$$R_{t+1} provider\ fairness\ (item) = -|Difference\ in\ acceptance\ rates(item)| \quad (5)$$

$$(S_t, A_t) = P(R_{t+1} user\ response | S_t, A_t = item) + R_{t+1} provider\ fairness(item) + R_{t+1} consumer\ fairness(item)$$

$$R: S \times A \rightarrow R \quad (6)$$

For solving this problem formulation and extracting the optimal policy, one can use Temporal Difference Methods [89] as they are appropriate for continuous tasks having discrete state and action spaces or Deep Q-Learning Networks [90] if one wants to include the reward values into the state.

IV. RESULTS AND DISCUSSION

In the current paper, it was presented the design of a Fairness Embedded Adaptive Recommender System (FEARS) which allows to create a fruitful relationship with the client by optimizing for long-term goals while making sure to keep both the consumer and provider fairness.

The gaps in the current conceptual practice were presented in the related work chapter. The desired functionalities of the system were stated in the proposed approach section and in the Table I it is shown how requirements were translated into implementation solutions along with potential issues and limitations that should take the form of further work.

The system was designed to use holistic user information including his socio-demographics and past buying behavior patterns. This was integrated into the system by encoding it into the state of the RL MDP model. As per author knowledge, this is a novel addition to the RecSys using RL approach. Another important detail is the inclusion of the recommendation context expressed through browsing time metadata (ex. time of the year, hour of the day). The way states are used in the RL engine, by clustering the initial

vectors, allows overcoming the most common RL limitation of non-efficiency and non-convergence.

Another way to ensure optimal policy convergence is to take control over action state space. This was stated as a clear problem in the literature reviewed and by using an elegant approach of defining actions as recommendation strategies and combination of those, the action space is downside from the number of all items to a maximum of 11 actions.

Although the individual recommendation components are traditional and straightforward, together they are covering the whole scene of consumer interest: similar, complementary, high interest or random products. The way these strategies are combined, namely through an RL engine, brings both adaptivity and ensures reaching long term objectives into the system. A detail that emphasizes the customer relationship health and importance is the practice of using negative rewards into RL component. This means that system will try to optimize for users to be recommended products that they are likely to buy but also play quite safe and not causing information fatigue that can lead to termination of relationship. Another addition is the fair view both towards consumer, but also with respect to items providers. Stated as a clear problem in the reviewed literature, the conflict between personalization, consumer fairness and exploration is solved by introducing not only consumer satisfaction rewards, but also fairness specific rewards.

In this paper, it was presented a conceptual framework that can be adapted to a large range of use cases, from e-commerce companies to both news, article and media items recommendation. Another set of application area may consist of those where the decisions and recommendations are linked to life-changing, high stake situations like hiring, job recommendation or financial lending.

The approach tries to overcome limitations of both individual traditional recommendation systems as well as RL usage in the RecSys by having an integrated view over consumer, a focus on the long-term engagement and a strong enforcing of a sustainable and fair recommendation practice.

TABLE I. SYSTEM REQUIREMENTS AND IMPLEMENTATION SOLUTIONS

System Requirements	Functionality Implementation	Potential issues/ limitations with the solution
System is focusing on customer relationship development	Usage of an RL engine that has the reward function linked to customer relationship goals.	The reward function is not reflecting accurately the desired objective.
System is incorporating an adaptivity functionality	Usage of a RL engine that takes complex states in account when recommending an item.	High complexity and search space that comes with all the additional information.
System is optimizing for long term customer engagement	RL engine is using negative rewards where appropriate in order to decrease information fatigue and optimize for long term objectives.	Negative rewards are too small in comparison with positive rewards affecting their efficiency
System is ensuring consumer and provider fairness	RL engine reward function is containing fairness metrics and outputs a higher reward in case of fair recommendations.	N/A
System is using a small action and state space in the RL engine	Action space is represented by individual or combinations of recommendation strategies. State space is discretized by using clustering techniques.	Oversimplification of the action and state spaces that could lead to pattern loss
System is solving the personalization and fairness trade-off	Reward function contains both recommendation relevance metrics and fairness metrics.	Inappropriate balance between the two objectives

Future research should involve implementing the approach and use it in real-world situations for evaluating the degree in which it reaches its multisided objectives. Also, different streams of work linked to potential issues presented in Table I should be carried.

V. CONCLUSIONS

Major contributions of this paper are presented as follows:

- A reinforcement learning based framework FEARS for better recommendations that focus on both revenues and relationship with the customer was introduced.
- The framework has a holistic view over customer and recommendation landscape ensuring a highly personalized, relevant and positive user interaction.
- Two relevant, adapted fairness metrics are defined and a way to compute them is presented.
- The relevant fairness metrics are embedded into the system as corresponding rewards.
- A RL problem definition was given that overcomes the common RL in RecSys issue of non-efficiency by using a limited, but relevant action space and discretized and clustered state space.

Overall, the system has all the necessary levers to overcome limitations of individual components, solve the personalization-fairness conflict, ensure long-term customer engagement and avoid the typical RL issue.

Same time, the framework should be tested in real-world situations or simulated data and appropriate design changes should be made. This is a conceptual starting point for developing FEARS.

ACKNOWLEDGMENT

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

REFERENCES

- [1] Sammut, C., & Webb, G. I. (2017). Encyclopedia of machine learning and data mining. Springer.
- [2] Mladenic, D. (1999). Text-learning and related intelligent agents: a survey. *IEEE intelligent systems and their applications*, 14(4), 44-54.
- [3] Schafer, J. B., Konstan, J., & Riedl, J. (1999, November). Recommender systems in e-commerce. In *Proceedings of the 1st ACM conference on Electronic commerce* (pp. 158-166).
- [4] Çano, E., & Morisio, M. (2017). Hybrid recommender systems: A systematic literature review. *Intelligent Data Analysis*, 21(6), 1487-1524.
- [5] Liu, F., Tang, R., Li, X., Zhang, W., Ye, Y., Chen, H., Zhang, Y. (2018). Deep reinforcement learning based recommendation with explicit user-item interactions modeling. *arXiv preprint arXiv:1810.12027*.
- [6] Choi, S., Ha, H., Hwang, U., Kim, C., Ha, J. W., & Yoon, S. (2018). Reinforcement learning based recommender system using biclustering technique. *arXiv preprint arXiv:1801.05532*.
- [7] Maqbool, S. D., Ahamed, T. I., & Malik, N. H. (2011, December). Analysis of adaptability of Reinforcement Learning approach. In *2011 IEEE 14th International Multitopic Conference* (pp. 45-49). IEEE.
- [8] Mabu, S., Tjahjadi, A., & Hirasawa, K. (2012). Adaptability analysis of genetic network programming with reinforcement learning in dynamically changing environments. *Expert Systems with Applications*, 39(16), 12349-12357.
- [9] Neftci, E. O., & Averbeck, B. B. (2019). Reinforcement learning in artificial and biological systems. *Nature Machine Intelligence*, 1(3), 133-143.
- [10] Taghipour, N., & Kardan, A. (2008, March). A hybrid web recommender system based on q-learning. In *Proceedings of the 2008 ACM symposium on Applied computing* (pp. 1164-1168).
- [11] Zhao, X., Zhang, L., Xia, L., Ding, Z., Yin, D., & Tang, J. (2017). Deep reinforcement learning for list-wise recommendations. *arXiv preprint arXiv:1801.00209*.
- [12] Zhao, X., Zhang, L., Ding, Z., Xia, L., Tang, J., & Yin, D. (2018, July). Recommendations with negative feedback via pairwise deep reinforcement learning. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (pp. 1040-1048).
- [13] Zhao, X., Gu, C., Zhang, H., Liu, X., Yang, X., & Tang, J. (2019). Deep Reinforcement Learning for Online Advertising in Recommender Systems. *arXiv preprint arXiv:1909.03602*.
- [14] Zou, L., Xia, L., Ding, Z., Song, J., Liu, W., & Yin, D. (2019, July). Reinforcement learning to optimize long-term user engagement in recommender systems. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (pp. 2810-2818).
- [15] Xin, X., Karatzoglou, A., Arapakis, I., & Jose, J. M. (2020, July). Self-Supervised Reinforcement Learning for Recommender Systems. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 931-940).
- [16] Li, L., Chu, W., Langford, J., & Schapire, R. E. (2010, April). A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web* (pp. 661-670).
- [17] Munemasa, I., Tomomatsu, Y., Hayashi, K., & Takagi, T. (2018, March). Deep reinforcement learning for recommender systems. In *2018 international conference on information and communications technology (icoiact)* (pp. 226-233). IEEE.
- [18] Dwork, C., Hardt, M., Pitassi, T., Reingold, O., & Zemel, R. (2012, January). Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference* (pp. 214-226).
- [19] Hardt, M., Price, E., & Srebro, N. (2016). Equality of opportunity in supervised learning. *arXiv preprint arXiv:1610.02413*.
- [20] Chouldechova, A. (2017). Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big data*, 5(2), 153-163.
- [21] Binns, R. (2018, January). Fairness in machine learning: Lessons from political philosophy. In *Conference on Fairness, Accountability and Transparency* (pp. 149-159). PMLR.
- [22] Verma, S., & Rubin, J. (2018, May). Fairness definitions explained. In *2018 IEEE/ACM international workshop on software fairness (fairware)* (pp. 1-7). IEEE.
- [23] Hutchinson, B., & Mitchell, M. (2019, January). 50 years of test (un) fairness: Lessons for machine learning. In *Proceedings of the Conference on Fairness, Accountability, and Transparency* (pp. 49-58).
- [24] Olteanu, A., Castillo, C., Diaz, F., & Kıcıman, E. (2019). Social data: Biases, methodological pitfalls, and ethical boundaries. *Frontiers in Big Data*, 2, 13.
- [25] Saxena, N. A., Huang, K., DeFilippis, E., Radanovic, G., Parkes, D. C., & Liu, Y. (2019, January). How do fairness definitions fare? Examining public attitudes towards algorithmic definitions of fairness. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society* (pp. 99-106).
- [26] Suresh, H., & Guttag, J. V. (2019). A framework for understanding unintended consequences of machine learning. *arXiv preprint arXiv:1901.10002*.
- [27] Oneto, L., & Chiappa, S. (2020). Fairness in machine learning. In *Recent Trends in Learning From Data* (pp. 155-196). Springer, Cham.
- [28] Feldman, M., Friedler, S. A., Moeller, J., Scheidegger, C., & Venkatasubramanian, S. (2015, August). Certifying and removing disparate impact. In *Proceedings of the 21th ACM SIGKDD*

- international conference on knowledge discovery and data mining (pp. 259-268).
- [29] Grgic-Hlaca, N., Zafar, M. B., Gummadi, K. P., & Weller, A. (2016, December). The case for process fairness in learning: Feature selection for fair decision making. In *NIPS Symposium on Machine Learning and the Law* (Vol. 1, p. 2).
- [30] Berk, R., Heidari, H., Jabbari, S., Joseph, M., Kearns, M., Morgenstern, J., Roth, A. (2017). A convex framework for fair regression. *arXiv preprint arXiv:1706.02409*.
- [31] Russell, C., Kusner, M. J., Loftus, J. R., & Silva, R. (2017). When worlds collide: integrating different counterfactual assumptions in fairness. *Advances in Neural Information Processing Systems 30*. Pre-proceedings, 30.
- [32] Corbett-Davies, S., & Goel, S. (2018). The measure and mismeasure of fairness: A critical review of fair machine learning. *arXiv preprint arXiv:1808.00023*.
- [33] Creager, E., Madras, D., Jacobsen, J. H., Weis, M., Swersky, K., Pitassi, T., & Zemel, R. (2019, May). Flexibly fair representation learning by disentanglement. In *International Conference on Machine Learning* (pp. 1436-1445). PMLR.
- [34] Kamishima, T., Akaho, S., Asoh, H., & Sakuma, J. (2012, September). Fairness-aware classifier with prejudice remover regularizer. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases* (pp. 35-50). Springer, Berlin, Heidelberg.
- [35] Corbett-Davies, S., Pierson, E., Feller, A., Goel, S., & Huq, A. (2017, August). Algorithmic decision making and the cost of fairness. In *Proceedings of the 23rd acm sigkdd international conference on knowledge discovery and data mining* (pp. 797-806).
- [36] Agarwal, A., Beygelzimer, A., Dudík, M., Langford, J., & Wallach, H. (2018, July). A reductions approach to fair classification. In *International Conference on Machine Learning* (pp. 60-69). PMLR.
- [37] Caton, S., & Haas, C. (2020). Fairness in Machine Learning: A Survey. *arXiv preprint arXiv:2010.04053*.
- [38] Ricci, F., Rokach, L., & Shapira, B. (2015). Recommender systems: introduction and challenges. In *Recommender systems handbook* (pp. 1-34). Springer, Boston, MA.
- [39] Koren, Y., & Bell, R. (2015). Advances in collaborative filtering. *Recommender systems handbook*, 77-118.
- [40] Cheng, W., Shen, Y., Zhu, Y., & Huang, L. (2018). Explaining Latent Factor Models for Recommendation with Influence Functions. *arXiv preprint arXiv:1811.08120*.
- [41] Su, X., & Khoshgoftaar, T. M. (2009). A survey of collaborative filtering techniques. *Advances in artificial intelligence*, 2009.
- [42] Bokde, D., Girase, S., & Mukhopadhyay, D. (2015). Matrix factorization model in collaborative filtering algorithms: A survey. *Procedia Computer Science*, 49, 136-146.
- [43] Wu, L. (2020). Advances in Collaborative Filtering and Ranking. *arXiv preprint arXiv:2002.12312*.
- [44] Wang, X., Jin, H., Zhang, A., He, X., Xu, T., & Chua, T. S. (2020, July). Disentangled Graph Collaborative Filtering. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 1001-1010).
- [45] Bonet, E. R., Nguyen, D. M., & Deligiannis, N. (2020). Temporal Collaborative Filtering with Graph Convolutional Neural Networks. *arXiv preprint arXiv:2010.06425*.
- [46] Li, X., Zhang, M., Wu, S., Liu, Z., Wang, L., & Yu, P. S. (2021). Dynamic graph collaborative filtering. *arXiv preprint arXiv:2101.02844*.
- [47] Maimon O., Rokach L. (2010), „Data Mining and Knowledge Discovery Handbook”, Springer.
- [48] Grad-Gyenge, L., Kiss, A., & Filzmoser, P. (2017, July). Graph embedding based recommendation techniques on the knowledge graph. In *Adjunct publication of the 25th conference on user modeling, adaptation and personalization* (pp. 354-359).
- [49] Chen, T., Hong, L., Shi, Y., & Sun, Y. (2017). Joint text embedding for personalized content-based recommendation. *arXiv preprint arXiv:1706.01084*.
- [50] Shi, C., Hu, B., Zhao, W. X., & Philip, S. Y. (2018). Heterogeneous information network embedding for recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 31(2), 357-370.
- [51] Grbovic, M., & Cheng, H. (2018, July). Real-time personalization using embeddings for search ranking at airbnb. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (pp. 311-320).
- [52] Zhao, Z., Zhang, X., Zhou, H., Li, C., Gong, M., & Wang, Y. (2020). HetNERec: Heterogeneous network embedding based recommendation. *Knowledge-Based Systems*, 204, 106218.
- [53] Palumbo, E., Monti, D., Rizzo, G., Troncy, R., & Baralis, E. (2020). entity2rec: Property-specific knowledge graph embeddings for item recommendation. *Expert Systems with Applications*, 151, 113235.
- [54] Talwar, K. S., Oraganti, A., Mahajan, N., & Narsale, P. (2015). Recommendation System using Apriori Algorithm. *Int. J. Sci. Res. Dev.*, 3(01), 183-185.
- [55] Agrawal, R., & Srikant, R. (1994, September). Fast algorithms for mining association rules. in *Proc. 20th int. conf. very large data bases, VLDB* (Vol. 1215, pp. 487-499).
- [56] Wang, C., & Zheng, X. (2020). Application of improved time series Apriori algorithm by frequent itemsets in association rule data mining based on temporal constraint. *Evolutionary Intelligence*, 13(1), 39-49.
- [57] Yao, F., Li, A., & Wang, Q. (2020, November). Bi-Apriori-Based Association Discovery via Alarm Logs. In *International Conference on Machine Learning and Big Data Analytics for IoT Security and Privacy* (pp. 621-632). Springer, Cham.
- [58] Kantardzic, M. (2019). *Data mining: concepts, models, methods, and algorithms*, 3rd Edition, John Wiley & Sons.
- [59] Rebal, G., Ravi, A., & Churiwala, S. (2019). Reinforcement Learning Algorithms. in *An Introduction to Machine Learning* (pp. 213-241). Springer, Cham.
- [60] Mohri, M., Rostamizadeh, A., & Talwalkar, A. (2018). *Foundations of machine learning*. MIT press.
- [61] Liu, F., Tang, R., Li, X., Zhang, W., Ye, Y., Chen, H., He, X. (2020). State representation modeling for deep reinforcement learning based recommendation. *Knowledge-Based Systems*, 205, 106170.
- [62] Ie, E., Jain, V., Wang, J., Narvekar, S., Agarwal, R., Wu, R., ... & Boutilier, C. (2019). Reinforcement learning for slate-based recommender systems: A tractable decomposition and practical methodology. *arXiv preprint arXiv:1905.12767*.
- [63] Zhou, S., Dai, X., Chen, H., Zhang, W., Ren, K., Tang, R., Yu, Y. (2020, July). Interactive recommender system via knowledge graph-enhanced reinforcement learning. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 179-188).
- [64] Chen, J., Kallus, N., Mao, X., Svacha, G., & Udell, M. (2019, January). Fairness under unawareness: Assessing disparity when protected class is unobserved. In *Proceedings of the conference on fairness, accountability, and transparency* (pp. 339-348).
- [65] Chiappa, S., & Isaac, W. S. (2018, August). A causal bayesian networks viewpoint on fairness. In *IFIP International Summer School on Privacy and Identity Management* (pp. 3-20). Springer, Cham.
- [66] Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., & Galstyan, A. (2019). A survey on bias and fairness in machine learning. *arXiv preprint arXiv:1908.09635*.
- [67] Tschantz, M. C., & Datta, A. (2015). Automated experiments on ad privacy settings: A tale of opacity, choice, and discrimination. *Proceedings on privacy enhancing technologies*, 2015(1), 92-112.
- [68] Lambrecht, A., & Tucker, C. (2019). Algorithmic bias? an empirical study of apparent gender-based discrimination in the display of stem career ads. *Management Science*, 65(7), 2966-2981.
- [69] Baeza-Yates, R. (2018). Bias on the web. *Communications of the ACM*, 61(6), 54-61.
- [70] Ciampaglia, G. L., Nematzadeh, A., Menczer, F., & Flammini, A. (2018). How algorithmic popularity bias hinders or promotes quality. *Scientific reports*, 8(1), 1-7.

- [71] Lerman, K., & Hogg, T. (2014). Leveraging position bias to improve peer recommendation. *PloS one*, 9(6), e98914.
- [72] Burke, R. (2017). Multisided fairness for recommendation. arXiv preprint arXiv:1707.00093.
- [73] Abdollahpouri, H., Burke, R., & Mobasher, B. (2017, July). Recommender systems as multistakeholder environments. In *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization* (pp. 347-348).
- [74] Burke, R., Sonboli, N., & Ordonez-Gauger, A. (2018, January). Balanced neighborhoods for multi-sided fairness in recommendation. In *Conference on Fairness, Accountability and Transparency* (pp. 202-214). PMLR.
- [75] Lee, E. L., Lou, J. K., Chen, W. M., Chen, Y. C., Lin, S. D., Chiang, Y. S., & Chen, K. T. (2014, August). Fairness-aware loan recommendation for microfinance services. In *Proceedings of the 2014 international conference on social computing* (pp. 1-4).
- [76] Celis, L. E., Straszak, D., & Vishnoi, N. K. (2017). Ranking with fairness constraints. arXiv preprint arXiv:1704.06840.
- [77] Castillo, C. (2019, January). Fairness and transparency in ranking. In *ACM SIGIR Forum* (Vol. 52, No. 2, pp. 64-71). New York, NY, USA: ACM.
- [78] Singh, A., & Joachims, T. (2019). Policy learning for fairness in ranking. arXiv preprint arXiv:1902.04056.
- [79] Narasimhan, H., Cotter, A., Gupta, M., & Wang, S. (2020, April). Pairwise fairness for ranking and regression. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 34, No. 04, pp. 5248-5255).
- [80] Yang, K., & Stoyanovich, J. (2017, June). Measuring fairness in ranked outputs. In *Proceedings of the 29th International Conference on Scientific and Statistical Database Management* (pp. 1-6).
- [81] Zliobaite, I. (2015). On the relation between accuracy and fairness in binary classification. arXiv preprint arXiv:1505.05723.
- [82] Yao, S., & Huang, B. (2017). Beyond parity: Fairness objectives for collaborative filtering. arXiv preprint arXiv:1705.08804.
- [83] Kamishima, T., Akaho, S., & Sakuma, J. (2011, December). Fairness-aware learning through regularization approach. In *2011 IEEE 11th International Conference on Data Mining Workshops* (pp. 643-650). IEEE.
- [84] Kiswanto, D., Nurjanah, D., & Rismala, R. (2018, October). Fairness aware regularization on a learning-to-rank recommender system for controlling popularity Bias in E-commerce domain. In *2018 International Conference on Information Technology Systems and Innovation (ICITSI)* (pp. 16-21). IEEE.
- [85] Ge, Y., Liu, S., Gao, R., Xian, Y., Li, Y., Zhao, X., Zhang, Y. (2021). Towards Long-term Fairness in Recommendation. arXiv preprint arXiv:2101.03584.
- [86] Joseph, M., Kearns, M., Morgenstern, J., Neel, S., & Roth, A. (2016). Fair algorithms for infinite and contextual bandits. arXiv preprint arXiv:1610.09559.
- [87] Liu, Y., Radanovic, G., Dimitrakakis, C., Mandal, D., & Parkes, D. C. (2017). Calibrated fairness in bandits. arXiv preprint arXiv:1707.01875.
- [88] Kotu, V., Deshpande, B. (2018). *Data science: concepts and practice*. Morgan Kaufmann.
- [89] Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.
- [90] Riedmiller, M. (2005, October). Neural fitted Q iteration—first experiences with a data efficient neural reinforcement learning method. In *European Conference on Machine Learning* (pp. 317-328). Springer, Berlin, Heidelberg.