*Journal of Social Sciences* *Vol. IV, no. 2 (2021), pp. 84 - 97*
*Fascicle* *Social Science* *ISSN 2587-3490*
*Topic* *Economics and Management* *eISSN 2587-3504*

**https://doi.org/10.52326/jss.utm.2021.4(2).09**
**UDC 339.5:004.738.5**

# DESIGNING A HOLISTIC ADAPTIVE RECOMMENDER SYSTEM (HARS) FOR CUSTOMER RELATIONSHIP DEVELOPMENT: A CONCEPTUAL FRAMEWORK

Alina Popa*, ORCID ID: 0000-0001-8727-1687

*Bucharest Academy of Economic Studies, 6 Piața Romană, Bucharest, Romania*
**popa.alina.alexei@gmail.com*

Received: 04. 11. 2021
Accepted: 05. 15. 2021

**Abstract.** With the recent COVID-19 pandemic, the world we knew changed significantly. The buying behavior shifted as well and is reflected by a growing transition to online interaction, higher media consumption and massive turn to online shopping. Companies that aim to remain top of mind to customers should ensure that their way of interacting with user is both relevant and highly adaptive. Companies should invest in state-of-the-art technologies that help manage and optimize the relationship with the client based on both online and offline data. One of the most popular applications that companies use to develop the client relationship is a Recommender System. The vast majority of traditional recommender systems consider the recommendation as a static procedure and focus either on a specific type of recommendation or on some limited data. In this paper, it is proposed a novel Reinforcement Learning-based recommender system that has an integrative view over data and recommendation landscape, as well as it is highly adaptive to changes in customer behavior, the Holistic Adaptive Recommender System (HARS). From system design to detailed activities, it was attempted to present a comprehensive way of designing and developing a HARS system for an e-commerce company use-case as well as giving a suite of metrics that could be used for its evaluation.

**Keywords:** *Recommender Systems, Customer Engagement, Reinforcement Learning, Framework, Integrated Customer View.*

**Rezumat.** Odată cu recenta pandemie COVID-19, lumea pe care o cunoșteam s-a schimbat semnificativ. Comportamentul cumpărătorului s-a schimbat și se reflectă printr-o tranziție în creștere către interacțiunea online, un consum mai mare de mass-media și o transformare masivă în cumpărături online. Companiile care își propun să rămână la îndemâna clienților ar trebui să se asigure că modul lor de interacțiune cu utilizatorul este atât relevant, cât și adaptabil. Companiile ar trebui să investească în tehnologii de ultimă generație care să ajute la gestionarea și optimizarea relației cu clientul atât pe baza datelor online, cât și offline. Una dintre cele mai populare aplicații pe care companiile le utilizează pentru a dezvolta relația cu clientul este un sistem de recomandare. Marea majoritate a sistemelor tradiționale de recomandare consideră recomandarea ca o procedură statică și se concentrează fie pe un anumit tip de recomandare, fie pe unele date limitate. În articol se

propune un nou sistem de recomandare bazat pe învățarea de consolidare, care are o viziune integrativă asupra datelor și peisajului de recomandare, precum și foarte adaptabil la schimbările de comportament ale clienților, Holistic Adaptive Recommender System (HARS). S-a încercat prezentarea modului de proiectare și dezvoltare a unui sistem HARS pentru un caz de utilizare al unei companii de comerț electronic, precum și oferirea unei suite de valori care ar putea fi utilizate pentru evaluarea acestuia.

**Cuvinte cheie:** *sisteme de recomandare, implicarea clienților, învățare consolidată, cadru, vizualizare integrată a clienților*.

**Introduction**

The year 2020 was a year of disruption, business challenges and change in consumer behavior patterns. A series of expert reports from big consulting companies show the main shifts in the buying patterns and consumer habits. In an early pandemic report from Accenture [1], it was brought into focus the e-commerce phenomenon on which the demand for online shopping has surged at unprecedented rates for all the categories of goods. The growth is coming mainly from the "new to online" segments of consumers. Another research [2] shows that this pattern is going to stick as consumers report an intent to shop online even after the crisis. Same publication highlights another trend called "shock to loyalty". This means that consumers are switching brands at increasing rates and try new shopping behaviors leading to the need of brands to convey value. The already existing overload of information on the Internet, new customer segments and the decrease in loyalty towards brands require companies to respond appropriately to these challenges in order to stay "top of mind" with prospects and clients and adapt quickly to their new needs and expectations.

Especially in the virtual environment, Data Mining applications can help solve issues and even take advantage of the current situation by efficiently managing the relationship with the customers [3].

In online 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 [4]. Today, RecSys are used extensively by big companies as Netflix [5, 6], Google [7, 8], Amazon [9]. 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 both 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 [10], while Collaborative filtering provides a good solution only under static scenarios when there are many users that bought or consumed the same product [11]. Hybrid recommender systems combine two or more recommendation strategies in different ways to benefit from their complementary advantages [12] and overcome the limitations of individual components. Another limitation of RecSys, regardless of strategy, that make them encounter challenges in rapidly changing environments is the assumption that user's underlying preferences remains unchanged, thus the recommendation procedure is a static process [13, 14].

One of the best-known approaches that allows to include adaptability in a system is Reinforcement Learning (RL) [15 - 17], being used successfully in robotics for changing environments [18], sustainable energy, heating and electric systems [19], intelligent educational system [20] and smart manufacturing systems [21].

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 [22 - 27]. A different approach is to use user and item sets which are obtained from bi-clustering as environmental states [14]. An earlier paper is using both user information and item information vectors and refers to it as context [28]. Important work on integrating negative influence of irrelevant recommendations is done by using negative rewards [24, 25, 27, 29].

To the knowledge of the author, none of the works present in the field combine in one approach the following elements of an efficient recommendation: 1) user information and past behavior, 2) item's characteristics 3) recommendation context, 4) consumption context, 5) adaptivity and 6) the idea of negative rewards.

Thus, in this paper, it is presented the design of a Holistic Adaptive Recommender System with focus on customer relationship development. The conceptual framework is combining three recommendation strategies leveraging extensive information about user, items and context using RL to automatically learn the optimal combination of these via trial-and-error. This will allow the system to quickly adapt to the changing needs of the customers and focus to build a fruitful relationship with the client.

## Literature Review

In this section, we first describe the basic problem of RecSys and their types. We then go more in depth into each of the recommender strategies specifics, latest advances and limitation. Next, we introduce RL practice and analyze it's to date usage and limitations in the recommendation area. Also, we will explore the different fairness formalization for ML systems and recommendation in particular and give an overview on how these definitions were translated into implementation.

### *Recommender Systems (RecSys)*

In the human decision-making process, obtaining recommendations from trusted sources is a critical component. Usually, this role is played my family, friends or subject-matter experts. Online space is offering a multitude of options and information that has the drawback of being too overwhelming to the user.

The goal of a recommender system is to create and give relevant recommendations of items or products to users. This will both increase customer satisfaction and will create a qualitative interaction with company's items or products. The design of a recommendation engine it is highly dependent on the data available and final goal of the system. RecSys can help customers find the items they want to purchase as well as improve cross-sells by suggesting additional products for the customer to acquire [11].

RecSys show differences in the way they analyze data sources and the way they conceptualize a good recommendation to be [4]. Depending on the structure of the learning system, we traditionally distinguish [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.

A different Machine Learning approach that tries to recommend an item or product to client is based on association methods. Although these methods are not referred to as recommendation strategies, the aim of the system is very similar. We refer to these methods as recommendation of complementary products later in the text.

Additionally, the system may have access to user-specific, context specific and item specific attributes such as demographics, browsing history and product description and reviews that allow to extend functionality of the traditional structure presented above.

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

***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 [30].

A method extensively used in modelling user behavior and preferences in the area of CF is Singular Value Decomposition (SVD). SVD gained extreme popularity after Netflix (a film and television company) hosted a competition in 2007 to develop a recommendation system based on the user's viewing history and preferences.

The aim of any recommendation system is to suggest elements that are relevant to users [31]. These can be TV shows or movies, as in the case of Netflix or food, clothing and books. The principle remains the same for any field, namely, creating a pleasant user experience that brings additional revenue to the company.

Because solutions that included the use of SVD or other matrix factorization and decomposition methods [32] achieved the highest performance at Netflix Prize, the method is mainly used in such applications.

The basic idea of SVD is to extract the latent dimensions of users and products starting from the matrix represented by the user notes given to the products used in the past.

Latent variables of users are usually considered their preferences in terms of product characteristics, e.g. product type, brand or functionality. At the same time, the latent variables of the products are the characteristics mentioned earlier.

The question SVD is trying to answer is "If consumer X were to use product Y, how would he rank it?" [33]. Before making this type of prediction, we need to model the ranks given by users as a function between characteristics and preferences, in other words between the latent variables of users and those of items. With the help of SVD, the “user-product” matrix can be decomposed into a matrix product [34], each of which represents a certain type of hidden variables.

Latest advances in the field include using graph encoding, Stochastic Shared Embeddings, large-scale Pairwise Collaborative Ranking, Sequential Recommendation Via Personalized Transformer [35] 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 [36 - 38].

***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.

This type of recommendation is useful in the case of a new user. If user has bought at least one product, we can make recommendations for products similar to the one bought. And if the company has a new product, the features can be extracted regardless of whether it was purchased or not, respectively it can be recommended to a user if it is similar to another product already purchased.

Since this method tries to extract similar objects, it is necessary to use a measure that can determine whether two objects are similar or different.

There are two main types of measures used to estimate this relationship: measures of distance and measures of similarity between objects [39].

To efficiently use both measures, an item characteristic should be transformed in a vectorial form containing only numerical values.

The intuition behind the use of distance as a measure of similarity is expressed by the idea that objects of the same kind are close to each other in the vector space, and different ones are at a greater distance.

There are a series of distances that can be used for this objective, like Euclidian, Manhattan, Minkowski [39]. Regarding Similarity metrics, these can be Jaccard, Pearson or Cosine [40].

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 [41 - 46].

***Complementary Product Recommendation***

When recommending complementary products, systems try to leverage the transaction history of customers [47]. Also, products often bought together explains a lot the consumption context.

Association algorithms are a group of unsupervised methods that are very popular in Data Mining projects.

These result in a set of association rules, which represent the hidden structures in the initial data.

The term Association Rule was first introduced by [48] in the context of shopping cart analysis. The methods try to measure the importance of the co-appearance of some elements or values from the data set.

Association Methods and APRIORI in particular are used in Recommendation Systems for educational courses and e-learning [49, 50], social media [51], library and books [52, 53] and e-commerce [54].

Still, they are not very popular in this area, although the methods are powerful in extracting knowledge and consumption context of the products or services.

The APRIORI algorithm remained essentially unchanged since its introduction to the research community, although there are sporadic efforts to extend it, for example adapting

it for time series of frequent items [55] or to make it more efficient by solving the problem of frequently scanned dataset and generation of large number of candidates [56].

***Reinforcement Learning (RL)***

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

As mentioned, 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 [57]. The mechanism that generates the reward and the transition from one state of the agent to another refers to the dynamics of the environment [39].

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. At first, an agent will not be able to correctly predict the outcome of his actions, but as he interacts with the environment and observes the consequences of his actions, he can adapt his behavior. Thus, learning in RL refers to the development of a tactic or policy that would help the agent to make the best decision in a situation.

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 [58]. The basic MDP model contains the following components:

• A set of environmental states S1, ..., Sn ∈ S: These can refer to the inherent characteristics of the agent or objects that surround and interact with it. The sum of the states of the environment is equal to the product of the number of values that each characteristic of the environment can take. The set of all possible states is known as the state space. The state space for an environment in which all characteristics are categorical is a finite or discrete space. If there are continuous features, this space is infinite.

• A set of actions that the agent can take, A1, ..., Am ∈ A: These refer to all possible actions that the agent controls. Both the set of actions and the set of states - can be finite or infinite.

• 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 of the situations and actions taken.

• The reward function represents the value of the reward obtained after acting with At in St.

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.

1. The agent interacts with the environment in a feedback loop pattern by following the steps below:

2. The agent observes the characteristics of the environment that define the current state, St.

3. The agent chooses an action from the set of possible actions, At, with which it responds to the environment in the current state St.

4. The agent enters a waiting state until the characteristics of the environment change with the St + 1 state and the agent receives the Rt+1 reward.
5. Repeat steps 1 - 3.

The agent's behavior or the way he interacts with the environment is described by a function called action policy or simple policy [59]. 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.

***Recommender Systems using Reinforcement Learning***

The intuition in which RecSys aim to provide recommendations to users with the objective of maximizing the long-term user satisfaction with the company requires a RL component that would keep track and optimize the system for this fruitful engagement and relationship with the client.

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 we need to specify the States, Actions and Rewards.

States are defined differently in the existing research literature. For example, they can reflect a mapping of previous user-item interactions into a hidden state [27], user's recommendation and ad browsing history [25], previous items that a user clicked [24], the sequence of visited and recommended items [22] or a more detailed interaction sequence that contains clicking, purchasing, or skipping, leaving [26]. An interesting approach is to define states as the cluster resulted from the co-clustering or bi-clustering of users and items [14] or to extend the state to include user demographics [13]. Efforts are as well invested in how to best represent the state in a RL RecSys [60]. 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 candidate items [24, 26, 27] or even whether to give a recommendation or not, and if yes, what would be the item to recommend [25]. There are authors that consider recommending a list of items [13, 23, 60]. One of the most different approaches it to recommend items from neighboring clusters to the user-items one [14]. As mentioned in multiple articles [61, 62], 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. This can be overcome by having a smaller action space through a step of pre-selected item types or recommendation strategies.

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 [22 - 25, 27). Reward can consist of immediate user feedback, but as well as a longer-term objective [26].

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 [14]. 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 [24, 25, 27, 29].

## Methodology

In this section, we propose a Holistic Adaptive Recommender System conceptual framework for customer relationship development that aims to close the gaps in the currently existing literature. To be more specific, we first set the objectives for our recommendation system and then propose a novel adaptive architecture, which is combining three recommendation strategies by using an RL engine. Then, we discuss step by step all the activities involved in such a system.

### *Holistic Adaptive Recommender System Objectives*

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

1. Using user information
2. Leveraging past customer behavior patterns
3. Including the recommendation context
4. Extracting value from using consumption context and patterns
5. Incorporating an adaptivity functionality
6. Optimizing for long term customer engagement using negative rewards where appropriate
7. Covering information about item's characteristics or recommended content

For exemplification purposes, we will use the case of an e-commerce company that is selling products and adapt the following concepts to it.

### *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 the client's needs.

The objective of the system 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.

The proposed recommendation system design is shown schematically in Figure 1.

### *System components*

*Data Ingestion*

The application starts by setting data sources (Figure 1, 1). The information considered mandatory is customer data, their past purchases, product data and external data on online reviews of the company's products.

*Data preparation*

The next step is to prepare the tables in the form in which they will be used in different components (Figure 1, 5):

- *The Customer-Product matrix:* The customer-product matrix contains information about the products purchased by a customer during the analysis period.
- *The Transaction-Product matrix:* Typically, data on customer purchases is stored in transactional format. Thus, a transaction contains several rows.

- *The product characteristics table:* In order to create the product feature table, we use the initial *Product Data Table* which contains all the tangible and intangible characteristics of an item. In addition to this information, we extract the main topics from reviews related to a certain product (Figure 1, 3).

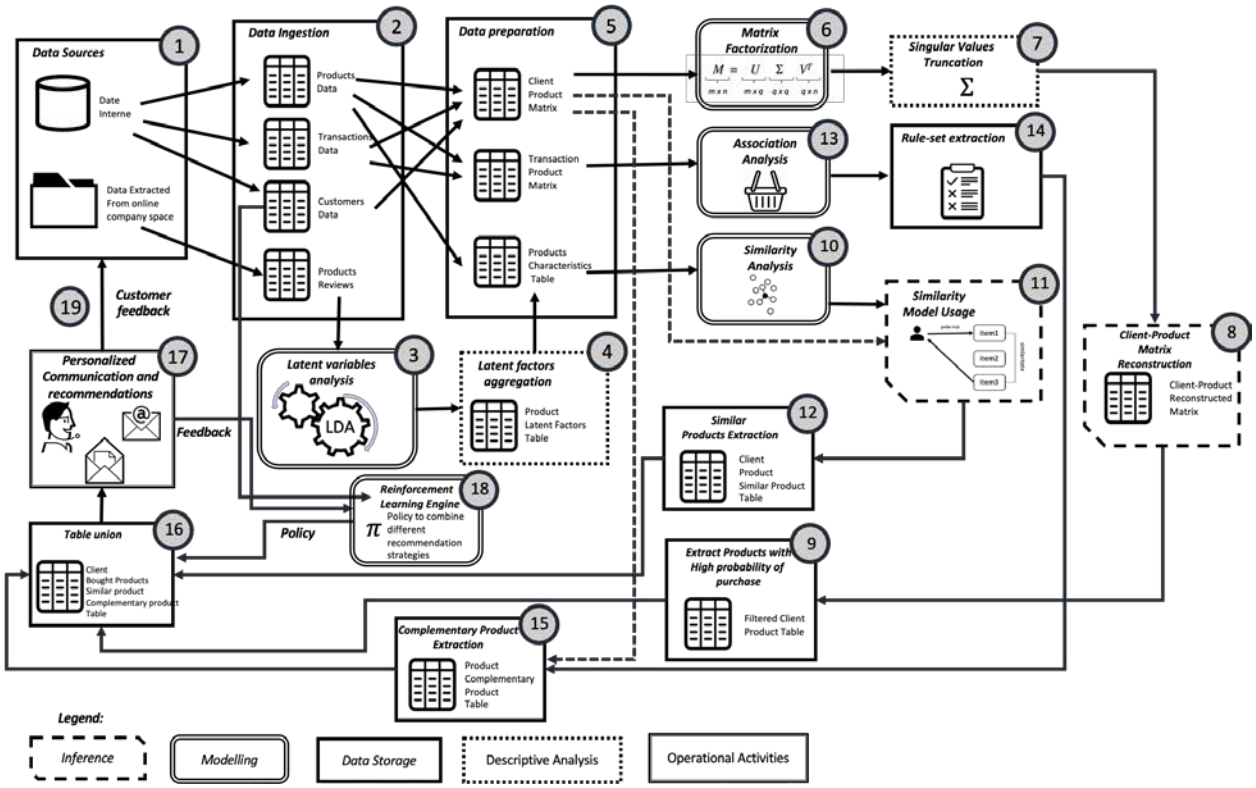


**Figure 1.** Design of a Holistic Adaptive Recommender System.

*Recommendation Components*

Once all the main tables are prepared (Figure 1, 5), three recommendation components are developed, and their results are merged in a later step based on a RL policy (Figure 1, 16).

*User-oriented collaborative filtering recommendation (*Figure 1, 6 - 9)*:* The method starts from the assumption that similar users have similar preferences [63] and reflects the real situation when recommendations from friends are more effective (since friends share preferences).

Step (Figure 1, 6) shows a model that tries to explain the customer-product shopping matrix using a set of latent factors. Latent structures are automatically deduced from the matrix, as long as the number of factors is specified/fixed [63]. 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.

*Content-based recommendation* (Figure 1, 10-12): In the case of new customers or products, instead of the matrix of products purchased by a customer, the *Product Characteristics Table* is used. For example, one can use all available information about the tangible and intangible properties of the product.

*Complementary product recommendation* (Figure 1, 13-15): In Association Analysis step, we extract rule sets from transactions. 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".

*Reinforcement Learning Engine*

The next task of the marketer is to choose the recommendation strategy for a type of customer. In other words, the question that needs to be answered is: "For this type of client, what is the most appropriate action to take? Recommendation of a product that corresponds to the latent structures of the client? A product similar to what he/she bought before or a complementary product?".

One solution that could combine customer information, purchasing behaviour in order to choose the best action is to use a RL component (Figure 1, 18).

Here, we need to define the RL problem as an MDP system:

- *The set of environmental states* is represented by the finite clusters over the vector space defined by the characteristics of the environment:
  - The socio-demographic characteristics of the customer
  - Its buying behavior segment created on data about diversification, focus on a product, appetite for novelty, past bought categories.
  - The characteristics 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.
- The *set of actions* represents all possible actions that the recommendation system controls. As there are three recommendation strategies, the set of actions is represented by the individual recommendations or combinations of them (Figure 1, 16), for example recommending a product that has a high probability of being purchased or recommending together a similar product and a complementary product.
  The advantage of formulizing the actions like this is the low complexity given by action space: instead of having all the candidate items, we have to just make the decision on how to combine the items given by traditional recommendation strategies.
- The *reward function* is conditioned by the client's response to the recommendation received (Figure 1, 19). A series of responses from client can be distinguished like "The customer clicks on the recommended product and buys it" or "The client opens the communication message, but does not click", each with an associated reward. The associated reward is set in such a way that reaches a negative value if user ends the communication with company.
- The *policy* is extracted using Temporal Difference Methods [64] as they are appropriate for continuous tasks having discrete space and action spaces.

*System Evaluation Metrics*

The efficiency of the recommender system can be determined using metrics specific to marketing campaigns:

- Click Through Rate (CTR) reflects the ratio of the number of people who clicked on a recommended product:

$$CTR = = \frac{Number\ of\ clicks}{Number\ of\ recommendations} \quad (1)$$

- The success rate is the ratio between the number of products purchased from those recommended:

$$Success\ rate = \frac{Number\ of\ products\ purchased\ from\ recommended}{Number\ of\ total\ recommended\ products} \quad (2)$$

- Return on Investment is a performance measure used to evaluate the efficiency of the overall investment in the developed system:

$$ROI_{application} = \frac{Revenues\ from\ Recommended\ Products\ Bought - System\ Development\ Cost}{System\ Development\ Cost} \quad (3)$$

It is taken into consideration only the revenues from recommended products and think of the system development cost expenses containing both the acquisition of technology, IT consulting services, but also the operational expenses for maintaining the application system.

**Results and discussions**

In the current paper, it was presented the design of a Holistic Adaptive Recommender System (HARS) for Customer Development, which allows to create a fruitful relationship with the client by optimizing for long-term goals.

First, the gaps in the current conceptual practice were extracted, then, it was designed a system that has a holistic view both towards user, but also with respect to recommendation strategies landscape. The way these strategies are combined, namely through an RL engine, brings both adaptivity and ensure 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. Two ways to ensure optimal policy convergence is to take control over action and state spaces. This was stated as a clear problem in the reviewed literature and by using an elegant approach of defining actions as recommendation strategies alone or combination of those, the action space was downsized from the number of all products in portfolio to a maximum of seven actions. The state definition aimed to use all available customer data, as well as recommendation context expressed through browsing time metadata. The way we decided to go to ensure optimal policy convergence, is to discretize the states vector space through clustering. As per author knowledge, both the holistic data inclusion, and actions defined as recommendation strategies are novel additions to the RecSys-RL approach.

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 news and media items recommendation. It tries to overcome limitations of both individual traditional recommendation systems as well as RL usage in the area by having an integrated view over customer and focus on the long-term engagement.

It is interesting to see this approach being implemented and used in real-world situations and evaluate how it compares with existing approaches.

**Conclusion**

Below, there is a summary of the major contributions:

- Identification of challenges in developing the customer relationship in the online space and proposal of a principled approach for better customer engagement.

- Proposal of a reinforcement learning based framework, HARS, for better recommendations that focus on both revenues and relationship with the customer.
- The framework has a holistic view over customer and recommendation landscape ensuring a highly personalized, relevant and positive user interaction.
- Novel RL problem definition 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.

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 a HARS type application.

**References**


1. Accenture (2020), "How COVID-19 will permanently change consumer behavior", accessed on 1 December 2020 at: https://www.accenture.com/_acnmedia/PDF-134/Accenture-COVID19-Consumer-Behaviour-Survey-Research-PoV.pdf#zoom=40
2. McKinsey (2020), accessed on 1st December at: https://www.mckinsey.com/~/media/McKinsey/Business%20Functions/Marketing%20and%20Sales/Our%20Insights/The%20great%20consumer%20shift/ten-charts-show-how-us-shopping-behavior-is-changing.pdf?shouldIndex=false
3. Chorianopoulos, A. (2016). Effective CRM using predictive analytics. John Wiley & Sons.
4. Sammut C., & Webb G. I. (2017). Encyclopedia of machine learning and data mining. Springer.
5. Gomez-Uribe, C. A., & Hunt, N. (2015). The Netflix recommender system: Algorithms, business value, and innovation. ACM Transactions on Management Information Systems (TMIS), 6(4), 1-19.
6. Amatriain X., & Basilico J. (2015). Recommender systems in industry: A Netflix case study. In Recommender systems handbook (pp. 385-419). Springer, Boston, MA.
7. Das A. S., Datar M., Garg A., & Rajaram S. (2007, May). Google news personalization: scalable online collaborative filtering. In Proceedings of the 16th international conference on World Wide Web (pp. 271-280).
8. Grasso A., Glance N. S., & Meunier J. L. (2008). U.S. Patent No. 7,386,547. Washington, DC: U.S. Patent and Trademark Office.
9. Smith B., & Linden G. (2017). Two decades of recommender systems at Amazon. com. Ieee internet computing, 21(3), 12-18.
10. Mladenic D. (1999). Text-learning and related intelligent agents: a survey. IEEE intelligent systems and their applications, 14(4), 44-54.
11. 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).
12. Çano E., & Morisio M. (2017). Hybrid recommender systems: A systematic literature review. Intelligent Data Analysis, 21(6), 1487-1524.
13. 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.
14. 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.
15. 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.
16. 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.
17. Neftci E. O., & Averbeck B. B. (2019). Reinforcement learning in artificial and biological systems. Nature Machine Intelligence, 1(3), 133-143.
18. Polydoros A. S., & Nalpantidis L. (2017). Survey of model-based reinforcement learning: Applications on robotics. Journal of Intelligent & Robotic Systems, 86(2), 153-173.
19. Yang T., Zhao, L., Li W., & Zomaya A. Y. (2020). Reinforcement learning in sustainable energy and electric systems: A survey. Annual Reviews in Control.

20. Iglesias A., Martínez P., Aler R., & Fernández F. (2009). Learning teaching strategies in an adaptive and intelligent educational system through reinforcement learning. Applied Intelligence, 31(1), 89-106.
21. Epureanu B. I., Li X., Nassehi A., & Koren Y. (2020). Self-repair of smart manufacturing systems by deep reinforcement learning. CIRP Annals, 69(1), 421-424.
22. 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).
23. 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.
24. 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).
25. 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.
26. 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).
27. 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).
28. 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).
29. 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.
30. Ricci F., Rokach L., & Shapira B. (2015). Recommender systems: introduction and challenges. In Recommender systems handbook (pp. 1-34). Springer, Boston, MA.
31. Koren Y., & Bell R. (2015). Advances in collaborative filtering. Recommender systems handbook, 77-118.
32. Cheng W., Shen Y., Zhu Y., & Huang L. (2018). Explaining Latent Factor Models for Recommendation with Influence Functions. arXiv preprint arXiv:1811.08120.
33. Su X., & Khoshgoftaar T. M. (2009). A survey of collaborative filtering techniques. Advances in artificial intelligence, 2009.
34. Bokde D., Girase S., & Mukhopadhyay D. (2015). Matrix factorization model in collaborative filtering algorithms: A survey. Procedia Computer Science, 49, 136-146.
35. Wu L. (2020). Advances in Collaborative Filtering and Ranking. arXiv preprint arXiv: 2002.12312.
36. 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).
37. Bonet E. R., Nguyen D. M., & Deligiannis N. (2020). Temporal Collaborative Filtering with Graph Convolutional Neural Networks. arXiv preprint arXiv:2010.06425.
38. Li X., Zhang M., Wu S., Liu Z., Wang L., & Yu P. S. (2021). Dynamic graph collaborative filtering. arXiv preprint arXiv:2101.02844.
39. Maimon O., Rokach L. (2010), „Data Mining and Knowledge Discovery Handbook", Springer
40. Strehl A., Ghosh J., & Mooney R. (2000, July). Impact of similarity measures on web-page clustering. în Workshop on artificial intelligence for web search (AAAI 2000) (Vol. 58, p. 64).
41. 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).
42. Chen T., Hong L., Shi Y., & Sun Y. (2017). Joint text embedding for personalized content-based recommendation. arXiv preprint arXiv:1706.01084.
43. 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
44. 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).

45. Zhao Z., Zhang X., Zhou H., Li C., Gong M., & Wang Y. (2020). HetNERec: Heterogeneous network embedding based recommendation. Knowledge-Based Systems, 204, 106218.
46. 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.
47. Talwar K. S., Oraganti A., Mahajan, N., & Narsale, P. (2015). Recommendation System using Apriori Algorithm. Int. J. Sci. Res. Dev, 3(01), 183-185.
48. Agrawal R., & Srikant R. (1994, September). Fast algorithms for mining association rules. în Proc. 20th int. conf. very large data bases, VLDB (Vol. 1215, pp. 487-499).
49. Aher, S. B., & Lobo, L. (2012, August). Applicability of data mining algorithms for recommendation system in e-learning. In Proceedings of the International Conference on Advances in Computing, Communications and Informatics (pp. 1034-1040).
50. Zhang H., Huang, T., Lv, Z., Liu S., & Zhou Z. (2018). MCRS: A course recommendation system for MOOCs. Multimedia Tools and Applications, 77(6), 7051-7069.
51. Liu L., Yu S., Wei X., & Ning Z. (2018). An improved Apriori–based algorithm for friends recommendation in microblog. International Journal of Communication Systems, 31(2), e3453.
52. More N., & More N. P. (2014). Recommendation of books using improved apriori algorithm. Int. J. Innov. Res. Sci. Technol, 1(4), 80-82.
53. Zhou Y. (2020). Design and implementation of book recommendation management system based on improved Apriori algorithm. Intelligent Information Management, 12(3), 75-87.
54. Ying H. (2012). Application of Improved Apriori Algorithm in E-Commerce Recommendation System. Computer & Digital Engineering, 08.
55. 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.
56. 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.
57. Kantardzic, M. (2019). Data mining: concepts, models, methods, and algorithms, 3rd Edition, John Wiley & Sons.
58. Rebala, G., Ravi, A., & Churiwala, S. (2019). Reinforcement Learning Algorithms. în An Introduction to Machine Learning (pp. 213-241). Springer, Cham.
59. Mohri M., Rostamizadeh A., & Talwalkar A. (2018). Foundations of machine learning. MIT press.
60. 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.
61. 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.
62. 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).
63. Kotu V., Deshpande B. (2018). Data science: concepts and practice. Morgan Kaufmann.
64. Sutton R. S., & Barto A. G. (2018). Reinforcement learning: An introduction. MIT press.