Temporal Modeling of User Preferences in Recommender System

Serhii Chalyi [1][0000-0002-9982-9091], Volodymyr Leshchynskyi [2][0000-0002-8690-5702]

[1] Department of Information Control Systems, Kharkiv National University of Radio Electronics, Nauky Ave. 14, Kharkiv, Ukraine
serhii.chalyi@nure.ua
[2] Department of Software Engineering, Kharkiv National University of Radio Electronics, Nauky Ave. 14, Kharkiv, Ukraine
volodymyr.leshchynskyi@nure.ua

Abstract. A dynamic model of user preferences is proposed for refining recommendations in the recommender system's online mode. The model is distinct by the use of temporal rules for building a temporal pattern of periodic changes in user preferences or a description of user requirements' evolution. Temporal rules define the sequence of events of an item selection by the user. The chain of rules determines the temporal dynamics of the user's preferences for the recommender system. Patterns similarity is determined by comparing the list of rules that make up this pattern and the weights of these rules. The adaptation of the model is performed when describing a temporal pattern of the evolution of user requirements by choosing a time interval within which the initial temporal rules are formed. The model can use the temporal rules for describing the individual behavior of the user, or the rules for choosing the subject. The model allows selecting the data that recorded users' choices with similar changes in preferences over time. It makes possible to reduce the time for building recommendations in the online mode while maintaining their accuracy.

Keywords: recommender system, temporal rule, temporal pattern, personalization of recommendations, user preferences.

1 Introduction

Recommender systems provide support for consumer choice by providing a personalized list of goods and services. This list contains a limited set of items that match the predicted consumer preferences. Therefore, the received recommendation simplifies the user's choice [1].

Recommender systems are widely used to search for goods, services, information, e-commerce systems, health care, banking, and teaching students. Such systems use information about the similarity of user preferences to build a recommended list of items. Information about consumer preferences is formed using the available data on the history of their choice of items that is presented in the recommendation system. The set of initial data is formed using an explicit and implicit feedback from the user.

An explicit feedback is expressed in the form of ratings of goods and services that the user exposes. An implicit feedback is set by information about purchases of items, moving through the site's pages with a description of these items, etc. Item ratings give a five-level preference detailing. However, they can be distorted as a result of shilling attacks [2]. Consumers' financial costs support purchase information, so it more accurately reflects the preferences of users of the recommender system.

The recommender system generates a list of items in offline and online modes. In the first case, calculations of recommendations are performed in batch mode, using all data about the user's selection history. Such recommendations are usually built in advance, before new user interaction. However, user preferences change over time, and offline recommendation results may become outdated. Already developed recommendations are updated online to eliminate this deficiency. Such calculations consider the temporal aspect of user behaviour, which makes it possible to satisfy their needs more accurately.

Existing approaches to building recommendations, considering the temporal aspect, focus either on building temporal patterns, or analysing data streams. Temporal patterns describe cyclical changes in user preferences over time. It is assumed using patterns, that user behaviour repeats at regular intervals. Patterns are formed in offline mode with the possibility of online correction.

The data stream analysis allows building a temporal model of user behaviour. It describes the evolution of user requirements for items offered by the recommender system. The temporal model contains information about the most recent actions of the user. Such a model should be updated online.

The proposed approach combines the capabilities of temporal patterns and temporal models based on data flow processing. Following this approach, a user behaviour model is built from weighted temporal rules. These rules set the order in time for pairs of events associated with the user's selection of items. Rules chain forms either a pattern that describes the cycle of user requirements or a model that describes the evolution of user preferences over time. The rules can be built offline. The rule base is updated as new data about the user's choice becomes available. When adding a set of rules, their weights are specified.

The contribution of this work is as follows: a dynamic model of user preferences in the form of a set of temporal rules is proposed; an approach to the construction and use of a rule-based model is proposed for refining recommendations in the online mode of the recommender system.

2 Literature review

Changes in user preferences over time are due to global and local factors. Global factors, in most cases, cause periodic changes in preferences in groups of people. Local or personal reasons are associated with obtaining an education, changing social status, place of work of a particular consumer. Thus, changes in consumer requirements can vary cyclically due to predominantly global factors, or they can change once due to local reasons for the user. Cyclic changes are considered when building recommendations using temporal patterns. In [3], it is proposed to distinct patterns depending on the length of time intervals during which the interests of the user change. In addition to taking into account the temporal dynamics of users' interests, the taxonomy of objects is also used. In [4], local patterns are used to predict the returning time when the user will select recommended items again.

Temporal patterns are built based on identifying patterns in a sequence of events that describe user actions [5]. When building patterns, only the most recent events can be used to adapt the recommendation model [6]. In this case, the actual data are selected using a sliding window [7]. An alternative approach is based on building dependence on the importance of events at their occurrence. With this approach, all input data are considered, but earlier events have less impact on the received recommendation [8]. In [9], it is proposed to filter data at several time intervals, depending on user preferences' similarity. The resulting data set is used to build recommendations using traditional methods.

Temporal patterns can also be formed using only the sequence of events, without specifying the absolute values of the time of their occurrence [10].

Evolutionary changes in user preferences result from a change in the context of decision making. Information about this context is missing in the recommendation system, making it challenging to build relevant recommendations based on temporal templates. In this case, building of recommendations is carried out using adaptive learning [11]. This approach's main idea is as follows: a set of actual data is highlighted using a sliding window. This set is used to build a model that allows predicting user behaviour. Over time, the set of actual data changes, and the model adapts using reactive and proactive approaches [12]. If the user's preference predicted by the model slightly deviates from his real choice, the model is adapted, taking into account the actual data. Significant differences between the model's predictions and the user's choice indicate a change in the concept of his behaviour. It leads to a significant change in his model [13].

In general, temporal patterns describe repetitive patterns and, therefore, cannot be used in a sharp change in user preferences. Adaptive learning separates the storage of data, models, and algorithms for detecting changes, making it difficult to use them online.

Thus, the task of constructing recommendations that immediately consider the evolution of user preferences requires further research. The solution to this problem is associated with creating a unified representation of data and knowledge describing the temporal dynamics of user requirements. For such a combination, temporal rules [14] can be used, which set the events in time. Such events in the recommendation system occur when a user purchases product, sets ratings and moves through the pages of the site.

3 Dynamic model of user preferences based on temporal rules

The dynamic user model uses adapted temporal rules for the same type of description of cyclical changes in his requirements, as well as the evolution of his preferences.

These rules link in one process a finite set of events that describe the user's choice (purchases, ratings, etc.). Initial sequence of events $E_j = \left\langle e_1, e_2, ..., e_k, ..., e_{|E_j|} \right\rangle$ is a subset of the event log or is contained in the recommendation system database. This sequence includes information about the user's choice. Each event includes data about the user u_j, the selected subject x_i, number of selected items n_i, as well as the moment of choice $\tau_{i,j}$. The event e_k can contain information X_k about the simultaneous selection of several items by the user:

$$e_k = \{u_j, X_k, \tau_i\}, X_k = \{(x_i, n_{i,k})\}. \tag{1}$$

Each pair of events $\langle e_k, e_l \rangle$ordered by the time the user selects items. Then the set of temporal rules $r_{k,l}$ for the case of choosing one item at a time is determined as follows:

$$R = \{r_{k,l} : (\forall k, l) \exists \langle e_k, e_l \rangle : \tau_l > \tau_k, |X_k| = 1, |X_l| = 1\}. \tag{2}$$

Rules (2) set the temporal order for two arbitrary events of the user choosing an object. Such events can occur sequentially in time:

$$r^{(1)}_{k,l} = \langle e_k, e_l \rangle : (\forall m \neq k, m \neq l), \tau_m \notin [\tau_k, \tau_l]. \tag{3}$$

There can also be intermediate events between rule events. Such a rule $r^{(2)}_{k,l}$ combines several rules $r^{(1)}_{k,l}$:

$$r^{(2)}_{k,l} \equiv \left\langle r^{(1)}_{k,m}, ..., r^{(1)}_{s,l} \right\rangle. \tag{4}$$

Rule (4) is intended to describe global dependencies that occur over long-time intervals. For example, this rule makes it possible to specify seasonal changes in users' demand.

Rules (3), (4) are intended for a general description of changing user preferences process when choosing only one item within the framework of one event.

The rule $r^{(3)}_{k,l}(x_i)$ is used to describe user behaviour before the selection of the target subject:

$$r^{(3)}_{k,l}(x_i) = \left\langle r^{(1)}_{k,m}, ..., r^{(1)}_{s,l} \right\rangle : (\forall l)(\forall \tau_k \leq \tau_m < \tau_l) n_{i,m} = 0, n_{i,l} > 0. \tag{5}$$

Rule (5) specifies a sequence of selection from several events, in which the final event e_l contains a choice of a target item x_i. This rule can describe a couple of recent events. It then specifies a local change in user preferences. The complete rule in the form (5) indicates the moment τ_l when, over a long period of time, due to an implicit event, the user's preferences have changed: the first item x_i was selected. The repetition of such rules affects the cyclical pattern of user behaviour of the recommendation system. Users u_j and u_d with similar interests will have overlapping subsets R_j^3 and R_d^3 of rules $r_{k,l}^{(3)}$. A discrete metric of similarity of user preferences based on comparison of rules $r_{k,l}^{(3)}$ is as follows:

$$dsim3(u_j, u_d) = \begin{cases} 1, \, if \; R_j^3 \cap R_d^3 \neq \varnothing, \\ o, otherwise. \end{cases} \tag{6}$$

Metrics of user behavior patterns $dsim1$ and $dsim2$ are defined similarly to (6) based on rules (3) and (4).

The scope of these metrics is different. Metrics $dsim1$ and $dsim2$ define local changes, as well as global cycles of changing user preferences. They are focused on building user-based recommendations. The metric $dsim3$ allows highlighting a group of users with similar behaviour who have shown interest in the subject x_i. Therefore, this metric is focused on building item-based recommendations.

The considered discrete metric act as constraints that allow selecting a subset of users with similar behaviour or with the similar process of selecting a target subject. A preliminary reduction in the amount of processed data will enable to reduce computational costs, which is essential when building recommendations in the recommendation system's online mode.

In case the user selects several items, then the conditions $|X_k| = 1$ и $|X_l| = 1$ are not performed. Then the rules (3) – (5) will bundle batch purchases. Such rules will be unique and challenging to use to predict the user's choice. It is expedient to represent pairs of events that show the choice of several items, by rules $r_{k,l}^{(1)}$, $r_{k,l}^{(2)}$ that set the sequence of selection for pairs of items. These rules, unlike (3) and (4) are combined in parallel.

The rule $r_{k,l}^{(1)}$ when the condition performed $|X_k| > 1$ takes the form:

$$r_{k,l}^{(4)} = \left\{ \left\langle e_k^{(x_1)}, e_l \right\rangle, \left\langle e_k^{(x_1)}, e_l \right\rangle, ..., \left\langle e_k^{\left(x_{|X_k|}\right)}, e_l \right\rangle \right\}. \tag{7}$$

The rule $r_{k,l}^{(2)}$ choosing several items, is denoted as $r_{k,l}^{(5)}$. This rule is detailed similarly to (7).

A model based on temporal rules that describes the individual behaviour of a user when $|X_k| > 1$ includes rules $r_{k,l}^{(1)}$, $r_{k,l}^{(2)}$, $r_{k,l}^{(4)}$, and $r_{k,l}^{(5)}$.

Detailing the rules $r_{k,l}^{(1)}$, $r_{k,l}^{(2)}$ when the condition performed $|X_l| > 1$ is performed in the same way.

The rule $r_{k,l}^{(3)}$ for parallel dependencies is denoted $r_{k,l}^{(6)}$ and is a rule supplemented by the condition of selecting the target item when the last event occurs e_l.

The set of the considered rules forms a dynamic model of user preferences. Varieties of this model differ in the used combination of temporal rules, as well as in the approach to identifying the time period of the actual rules.

The model M_{u_j} based on temporal rules, describing the individual behaviour of the user, includes set of rules $R = \bigcup_{g=1}^{4} R^{(g)}$ where $R^{(g)} = \left\{ r_{k,l}^{(g)} \right\}$:

$$M_{u_j} = R \left| \left(\forall r_{k,l} \in R \right) \exists u_j \in e_k , u_j \in e_l \right. . \tag{8}$$

This model sets the temporal pattern of user u_j preferences. The resulting pattern allows using discrete metrics $dsim1$ and $dsim2$ to select subsets of users with similar temporal dynamic. A discrete metric of user models' similarity can be used to filter the original sets of events. Pre-filtering of data is used not only when working in the online mode, but also when building recommendations in a cyclic cold start situation [15]. The latter is characterized by long cycles of changes in user preferences. As a result, their purchase history becomes outdated, and this user is treated as new or "cold."

Combination of rule sets $G = R^{(5)} \bigcup R^{(6)}$ designed to simulate user behaviour u_j when choosing a target subject x_i :

$$M_{x_i , u_j} = G \left| \left(\forall r_{k,l} \in G \right) \exists \left(u_j \in e_k , u_j \in e_l \right), \exists x_i \in e_l \right. . \tag{9}$$

The use of models (8) and (9) for building temporal patterns and for processing the incoming data stream about the current choice of users is performed in two ways.

First, through the choice of the time interval $\left[\tau_k , \tau_l \right]$ between the first and the last event of all rules in the dynamic user preference model. Moment of time τ_l corresponds to the moment of the last, most relevant user action. When choosing a short interval $\left[\tau_k , \tau_l \right]$ the model will contain the most actual rules. Such a model describes the current input data stream and is intended to formalize user preferences' evolutionary changes. When most of the user's selection history included in this interval, the model is suitable for building temporal patterns.

Secondly, the capabilities of the model change by limiting the list of used rules. When building temporal patterns, the rules $r_{k,l}^{(2)}$ and $r_{k,l}^{(5)}$ are used. These rules cover a chain of several consecutive events, which allows you to describe the cycles in the user's choice. With the current change in user preferences, it is expedient to use only rules $r_{k,l}^{(1)}$ and $r_{k,l}^{(4)}$. These rules only compare the previous and current consumer choices.

Rules $r_{k,l}^{(3)}$ and $r_{k,l}^{(6)}$ are used with similar restrictions:

$$r_{k,l}^{(3)} \left(x_i \right) = \left\langle r_{k,l}^{(1)} \right\rangle : \left(\forall k, l \right) n_{i,k} = 0, n_{i,l} > 0. \tag{10}$$

According to (10), when describing user interests' evolution, only the temporal order for pairs of consecutive events is taken into account.

4 An approach to building dynamic models of user preferences when creating recommendations

When using dynamic models of user preferences, assessing the proximity of models for different users is carried out. Based on this assessment, the recommendations are refined.

The numerical estimation of the proximity of user models uses rule weights. The weights of the rules, depending on the problem being solved, are calculated in two ways. When forming a dynamic user model designed to predict his behaviour, the rules' weight is calculated according to the approach proposed in [16] as a normalized difference of values $n_{i,l}$ and $n_{i,k}$. This approach advantage is low computational costs, which makes it possible to adjust the weight of new rules in online mode.

When implementing an item-based approach using rules $r_{k,l}^{(3)}$ and $r_{k,l}^{(6)}$, then according to the approach [17] it is expedient to use the algorithm based on a random search. In this case, the weights of the same rules for different users are matched taking into account the probability of occurrence of pairs of events for the rule.

The closeness between the dynamic preference models of two users is calculated based on the general rules' weights.

The weights of the general rules calculated by the algorithm [17] will be the same for both compared models. Therefore, the proximity metric $sim1$ is the sum of the weights of these rules:

$$sim1\left(M_{x_i,u_j}, M_{x_i,u_d}\right) = \sum_{r_{k,l} \in G} w_{k,l} \left| \left(\forall r_{k,l} \in G\right) dsim3 = 1, \right. \tag{11}$$

where $w_{k,l}$ – weight of the rule $r_{k,l}$.

The weights of the common rules calculated according to the approach [16] differ since users can choose a different number of items. Accordingly, the degree of closeness is calculated as the arithmetic mean of the total weight of the rules:

$$sim2\left(M_{u_j}, M_{u_d}\right) = \frac{1}{2}\left(\frac{W_{u_j}}{\left|W_{u_j}\right|} + \frac{W_{u_d}}{\left|W_{u_d}\right|}\right) \left| \left(\forall r_{k,l} \in G\right) dsim3 = 1, \right. \tag{12}$$

where $W_{u_j} = \sum_{r_{k,l} \in M_{x_i,u_j}} w_{k,l}$, $W_{u_d} = \sum_{r_{k,l} \in M_{x_i,u_d}} w_{k,l}$ total rule weights in models of users u_j and u_d.

The weights of the rules are normalized, so the resulting estimate will also be normalized.

The sequence of building and using dynamic models of user behaviour in building recommendations includes the following phases and stages.

Phase 1. Building the model

A given time interval $\left[\tau_k, \tau_l\right]$, a set of events $E = \left\{E_{u_j}\right\}$, and a subset of users $U = \left\{u_j\right\}$ are used as the initial data in the first phase.

Stage 1. Formation of a basic set of temporal rules.

The set of rules depends on the requirements for the model (pattern of behaviour or evolution of preferences, user model, or model of choosing an object). The pattern of user preferences' temporal dynamics is set using a combination of rules (3) and (4). The evolution of user requirements is determined using rules (3). The temporal pattern that sets the dynamics of the choice of an object x_i uses rules (5). The evolution of requirements for an item is determined using rules (10).

Stage 2. Building models for users from a subset in accordance with (8) or (9).

Phase 2. Using the model.

Stage 3. Pre-filtering rules from the resulting models using discrete metrics $dsim1$ and $dsim2$, or $dsim3$.

The first two metrics are used for rules (3) and (4), and the third is used for rules (5) and (10).

Stage 3. Calculating weights for temporal rules that meet criteria $dsim1 = 1$ and $dsim2 = 1$, or $dsim3 = 1$.

When using rules (3) and (4), weights are calculated based on the approach considered in [16]. In the case of using rules (5) and (10), the weights are calculated within the frame-work presented in [17].

Stage 4. Calculation of model proximity metrics according to (11) for individual user models or (12) for item selection models.

Stage 5. The selection of close models provided that the proximity indicator exceeds the threshold value.

The selected models of the temporal dynamics of users allow them to promptly adjust the recommendations by filtering the set of input data. The purpose of filtering is to select from the general input dataset the events of only those users whose preferences satisfy the temporal dependencies. In the future, recommendations are formed on the filtered dataset using traditional methods.

5 Experiments and results

Experimental verification of the dynamic model of user preferences and the approach to its use was performed based on the “Online Retail dataset” from the UCI repository.

This set contains a log of sales events of a wholesale chain of gift stores. Wholesale purchases are periodically repeated, since the interest in gifts changes cyclically, depending on the holidays. During the experiment, two subsets containing 6 and 10-

week gift sales records were formed. Such small subsets were allocated for testing the duration of the cycles of changing user interests.

The purpose of the experimental test was to evaluate the possibility of filtering data based on a temporal model and check the effect of filtering on the resulting recommendations.

Two experiments were performed. The first experiment is focused on the construction of temporal templates describing the cyclical change in the user's interest in gifts.

In the second experiment, the purchase history was partially removed to reflect user preferences' current evolution. In other words, in the framework of the second experiment, a cold start situation was simulated.

The data were filtered using the user preference models. After pre-filtering, recommendations were formed using the collaborative filtering method.

The experiment results were evaluated using the AUC (Area Under the Curve) value. The results of the first experiment are shown in the Table. 1.

Table 1. The results of the experiment with temporal patterns

Algorithms for comparison	AUC	
	6 weeks	10 weeks
Collaborative filtering	0.81	0,85
Temporal pattern and Collaborative filtering	0.79	0.86

From the Table. 1 shows that for the first subset of data, the AUC indicator slightly worsened due for the recommended list of items to filtering, and for the second, it increased. This means that in the first case, data essential for collaborative filtering was removed from the original set. Pre-filtering worsened the resulting recommendation as the user's preference cycle is longer than four weeks. The results of the recommendations in the second subset show that this cycle is at least ten weeks. To further improve the recommendations' accuracy, it is necessary to increase the size of the original data subset.

The results of the second experiment are shown in the Table. 2.

Table 2. The results of the experiment with evolution of preference

Algorithms for comparison	AUC	
	6 weeks	10 weeks
Collaborative filtering	0.7	0,73
Temporal pattern and Collaborative filtering	0.714	0.74

In this experiment, the size of the dataset decreased significantly due to the partial deletion of the purchase history. Removing most of the event history for targeted users led the collaborative filtering algorithm to select the most popular products. This significantly reduced the AUC indicator. Pre-filtering made it possible to select data that affect the overall assessment.

The small size of the initial subsets does not allow achieving high AUC values, but it is possible to assess the effect of temporal filtering on the final recommendation. This experiment represents the impact of temporal dependencies on the resulting recommendations, even when using a small sample size.

This approach uses a set of typical rules, which allows scaling the temporal model of user preferences, in contrast to the methods considered in [8].

A distinctive feature of this approach is significant reduction in the sample size due to pre-filtering, which reduces computational costs. There is no loss of accuracy, which allows using the temporal approach to supplement the recommendations online. Increasing the accuracy of recommendations requires improving the algorithm for calculating weights.

6 Conclusion

A dynamic model of user preferences in a recommendation system based on the use of temporal rules is proposed. The temporal rule sets the order in time for a pair of events. Each of these events describes a user's choice. The main difference of the proposed model is that the typical set of temporal rules determines both temporal patterns for cycles of changing user preferences due to global factors and the evolution of user requirements as a result of local factors. The model is adapted by changing the time interval for the formation of temporal rules and selecting rules of the corresponding types. The model provides the ability to filter data taking into account the temporal characteristics of user behaviour, which makes it possible to reduce the time for building recommendations in the online mode, as well as to adapt recommendations in a cold start situation of the recommendation system.

References

1. Aggarwal, C. C.: Recommender Systems: The Textbook. Springer, New York (2017)
2. Chala, O., Novikova, L., Chernyshova, L.: Method for detecting shilling attacks in e-commerce systems using weighted temporal rules. EUREKA: Physics and Engineering 5, 29-36 (2019). doi: 10.21303/2461-4262.2019.00983
3. Raza, S., Ding C.: News Recommender System Considering Temporal Dynamics and News Taxonomy. In: 2019 IEEE International Conference on Big Data, 920-929 (2019). doi: 10.1109/BigData47090.2019.9005459 ·
4. Nan, D., Yichen, W., Niao, H., Le, S.: Time-Sensitive Recommendation From Recurrent User Activities. In: NIPS'15: Proceedings of the 28th International Conference on Neural Information Processing Systems, vol. 2, pp. 3492–3500, (2015)
5. Campos, P. G., Diez, F., Cantador, I.: Time-aware recommender systems: a comprehensive survey and analysis of existing evaluation protocols. User Modeling and User-Adapted Interaction, 24 (1-2), pp. 67-119 (2014). doi: 10.1007/s11257-012-9136-x

6. Ma, Y., Narayanaswamy, B., Lin, H., Ding, H.: Temporal-Contextual Recommendation in Real-Time. In: KDD '20: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 2291–2299 (2020). doi: 10.1145/3394486.3403278
7. Rabiu, I., Naomie, S., Aminu, D., Akram, O.: Recommender System Based on Temporal Models: A Systematic Review. Applied Sciences, 10(7), 2204. (2020). doi: 10.3390/app10072204
8. Vinagre, J., Jorge, A. M., Gama, J.: An overview on the exploitation of time in collaborative filtering. WIREs Data Mining and Knowledge Discovery, 5(5), pp. 195-215 (2015). doi: 10.1002/widm.1160
9. Hidasi, B., Quadrana, M., Karatzoglou, A., Tikk, D.: Parallel recurrent neural network architectures for feature-rich session-based recommendations. In: 10th ACM Conference on Recommender Systems (RecSys'16), pp. 241–248. Boston, USA (2016). doi: 10.1145/2959100.2959167
10. Chalyi, S., Pribylnova, I.: The method of constructing recommendations online on the temporal dynamics of user interests using multilayer graph. EUREKA: Physics and Engineering 3, 13-19 (2019). doi: 10.21303/2461-4262.2019.00894
11. Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., Bouchachia, H.: A survey on concept drift adaptation. In: ACM Computing Surveys (CSUR), vol. 46 (4) (2014). doi: 10.1145/2523813
12. Kraus, M., Fischbach, F., Jansen, P., Minker, W.: A Comparison of Explicit and Implicit Proactive Dialogue Strategies for Conversational Recommendation. In: Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), pp. 429–435 (2020).
13. Ghossein, M., Abdessalem, T., Barré, A.: Dynamic Local Models for Online Recommendation. In: Proceedings of the 2018 World Wide Web Conference, pp. 1419–1423 (2018). doi: 10.1145/3184558.3191586
14. Levykin, V., Chala, O.: Development of a method for the probabilistic inference of sequences of a business process activities to support the business process management. Eastern-European Journal of Enterprise Technologies, 5/3(95), 16-24 (2018). doi: 10.15587/1729-4061.2018.142664
15. Chalyi, S., Leshchynskyi, V.: Method of constructing explanations for recommender systems based on the temporal dynamics of user preferences. EUREKA: Physics and Engineering 3, 43-50 (2020). doi: 10.21303/2461-4262.2020.001228
16. Chalyi, S., Leshchynskyi, V., Leshchynska, I.: Method of forming recommendations using temporal constraints in a situation of cyclic cold start of the recommender system. EUREKA: Physics and Engineering 4, 34-40 (2019). doi: http://dx.doi.org/10.21303/2461-4262.2019.00952
17. Gogate, V., Domingos, P.: Probabilistic theorem proving: A Unifying Approach for Inference in Probabilistic Programming. Communications of the ACM, 59(7), 107-115 (2016). doi: 10.1145/2936726