OPEN ACCESS
International Journal of Innovations in Science & Technology

Contrasting Impact of Start State on Performance of A Reinforcement Learning Recommender System

Sidra Hassan, Mubbashir Ayub*, Muhammad Waqar, and Tasawer Khan
Department of Software Engineering, University of Engineering and Technology, Taxila, Pakistan
***Correspondence:** mubbashir.ayub@uettaxila.edu.pk

Citation | Hassan. S, Ayub, M, Waqar. M, Khan. T, "Contrasting Impact of Start State on Performance of a Reinforcement Learning Recommender System", IJIST, Vol. 6 Issue. 2 pp 565-581, May 2024

Received | May 10, 2024 **Revised |** May 19, 2024 **Accepted |** May 25, 2024 **Published |** May 28, 2024.

A recommendation problem and RL problem are very similar, as both try to increase user satisfaction in a certain environment. Typical recommender systems mainly rely on history of the user to give future recommendations and doesn't adapt well to current changing user demands. RL can be used to evolve with currently changing user demands by considering a reward function as feedback. In this paper, recommendation problem is modeled as an RL problem using a squared grid environment, with each grid cell representing a unique state generated by a biclustering algorithm Bibit. These biclusters are sorted according to their overlapping and then mapped to a squared grid. An RL agent then moves on this grid to obtain recommendations. However, the agent has to decide the most pertinent start state that can give best recommendations. To decide the start state of the agent, a contrasting impact of different start states on the performance of RL agent-based RSs is required. For this purpose, we applied seven different similarity measures to determine the start state of the RL agent. These similarity measures are diverse, attributed to the fact that some may not use rating values, some may only use rating values, or some may use global parameters like average rating value or standard deviation in rating values. Evaluation is performed on ML-100K and FilmTrust datasets under different environment settings. Results proved that careful selection of start state can greatly improve the performance of RL-based recommender systems.

Keywords: Recommender Systems, Reinforcement Learning, Collaborative Filtering, Similarity Measures, Start State, Q-Learning.

Introduction:

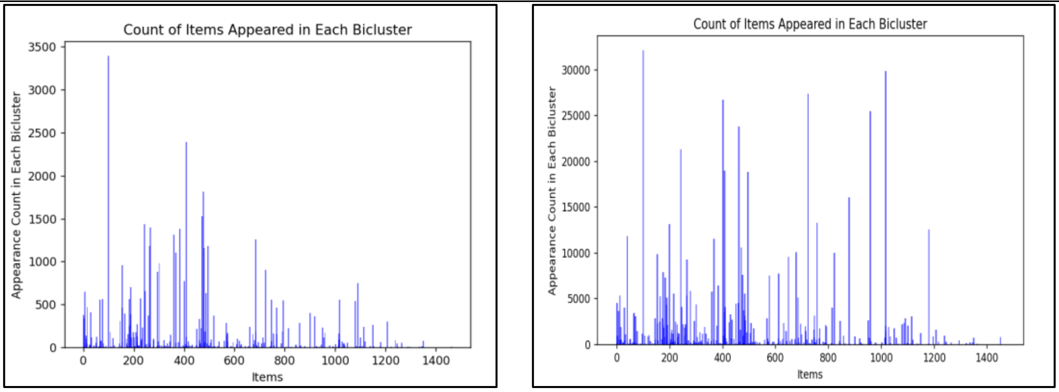
In this era of the digital world, Recommender Systems (RSs) are used as a tool to address information overload problems [1]. Conventional recommender systems use Collaborative Filtering (CF) or content-based filtering to generate recommendations. However, these methods rely on statistical inherence of features to generate predictions and suffer from problems of data sparsity, cold start, and absence of exploration [2]. Modeling a conventional recommendation problem as an RL problem offers several advantages including prolonged user engagement, diverse forms of user-item interactions, encompassing actions such as clicks, and purchases, balancing exploration and exploitation, and adapting to changing user preferences [3]. Reinforcement Learning (RL) is a good option due to its adaptability to dynamic settings. To model the recommendation problem as an RL framework, we have to define an environment for the RL problem. In this study, we adopt the squared grid environment proposed in [4]. An RL problem is characterized by an environment that includes a state space, an action space, a reward function, a state transitioning function, and a goal state [5].

Considerations for starting an RL-based RS using a squared grid environment is pivotal. This research paper addresses fundamental questions such as where to begin, what rewards to expect from different starting points, and how to navigate through the grid environment to optimize recommendations. Our study focuses on two distinct grid sizes: 6×6 and 7×7 (as shown in Figure 2), providing insights into how grid dimensions impact recommendation performance within an RL framework. The 6×6 grid environment comprises 36 states, while the 7×7 grid environment comprises 49 states. Any state can be a start state but we want to find such a start state that can lead to optimized and accurate recommendations for user. To achieve this, we employed seven different start-state selection methods to identify the most effective approach for optimized and accurate recommendations.. The rest of the paper is structured as follows; section 2 is the literature review section, section 3 highlights the methodology and working of the proposed method, section 4 discusses observed outcomes and section 5 concludes our work and gives future directions.

Literature Review:

Collaborative Filtering (CF) is widely used in e-commerce websites where users give ratings to products that they purchased or have viewed. This approach consists of making recommendations by looking for correlations between "liked" and "disliked" products among users of the system. For example, a movie Recommender System will search for users similar to the target user; and only movies well rated by these users will be recommended to the target user. In this way, the system will recommend the same item to a set of users having a similar taste as this user [6]. This problem is considered a clustering process since it aims to classify a set of users/items into homogeneous groups. This clustering relies on calculating the semantic distance or similarity between elements within a group. As elements share more common features, the similarity value increases [7]. Hence, selecting an appropriate similarity measure among a very large set of available measures is considered a crucial task when implementing a CF RS. Therefore, the quality of similarity measures directly impacts the accuracy of CF RSs [8].

In the RL field, an agent's objective is to perform actions in the environment to maximize its cumulative reward. Unlike unsupervised learning, where the focus is on discerning differences and similarities among data points, RL centers on determining actions that, when taken in the environment, lead to maximizing the agent's cumulative reward. In RL, goal is to find actions that when performed in the environment, maximize the cumulative reward of the agent. The solution to an RL problem is a policy which is a set of actions that leads to a goal state [9]. An RL problem is formulated as a Markov Decision Process (MDP) comprising five components. A set of possible states (S), a set of available actions (A), a reward function (R), transition probability (P) of moving from one state to another, and discount factor ϒ [10].



(a)ML-100K (b)Film Trust

Figure 1: Count of items that appeared in each Bicluster, (a) ML-100K dataset (b) Film Trust dataset

The primary objective of clustering is to identify hidden patterns within the dataset based on certain similarities. Clustering algorithms typically operate on one data dimension, either rows or columns, but not both [11]. Moreover, for these clustering algorithms, a single user can only exist in one cluster. Clustering algorithms are effective at identifying global patterns in a dataset but may struggle to detect local patterns within subsets of rows and columns. In contrast, biclustering algorithms [12], have the capacity to simultaneously group a subset of rows and columns, allowing for the extraction of local patterns within a dataset.

Iftikhar et al. [13] proposed ITR similarity to work in sparse CF RSs. Special property of ITR operates on all items of both users, while other measures use only common items. The robustness of ITR was also vetted by F. Fkih [6]. Work in [1] used ITR similarity measure to determine the start state for a squared grid environment. Choi et al. [14] used Jaccard similarity to determine the start state in a squared grid environment. Work in [10] used the Cosine similarity measure to determine the start state. So, the authors of this study aim to find the best similarity measure to select a start from an abundance of available similarity measures, in a squared grid environment.

Dataset Description and Methodology:

In this research, we utilized two standard datasets of recommender systems whose detail is given below.

Movie Lens-100K (ML-100K) Dataset [15]:

This dataset is comprised of 943 users who rated different movies on a scale of 1.0 (worst rating) to 5.0 (best rating). The most-rated value of this dataset is 4.0. This dataset includes 100000 user ratings over 1682 movies and each user rated at least 20 movies, having a sparsity of 93.70%.

Film Trust Dataset [16][17]:

Film Trust is a trust-based social site in which users can rate and review movies. The rating dataset contains 1508 users, 2071 movies, and 35,497 ratings. The ratings take values from 0.5 to 4.0 with the step of 0.5. The sparsity of this dataset is 98.86%. Out of 35,497 rating values, 4.0 is the maximum repeated value having a count of 9,320. In this study, we employed the squared grid environment proposed in [1] as an RL environment for our proposed system. As typical datasets are comprised of a large number of users and items it can result in a very large state space for RL problems. To reduce state space, we applied Bibit biclustering algorithm to generate biclusters of datasets. The Biclustlib-master Python library facilitated the generation of these biclusters [18]. After the generation of biclusters, we observed significant overlapping in the user and item set of each bicluster. We detected and plotted this overlapping for ML-100K dataset in Figure 1(a). Bibit algorithm requires two input parameters for the generation of biclusters which are a minimum number of rows and columns in the required bicluster. After inputting these two parameters, Bibit generates biclusters that have rows and columns greater than a minimum number of rows and columns. At minimum rows =20 and minimum columns =10, 4,406 biclusters are generated. Figure 1(a), illustrates that out of the 1,682 movies in the dataset, 1,385 unique movies were present within these biclusters, encompassing a total of 51,097 items. Notably, it is observed that few items/movies occurred in almost all biclusters, like movie id 100, which occurred in 3394 biclusters. This analysis sheds light on the structural patterns and distribution of items within the generated biclusters, offering insights into the dataset's organization and potential clustering strategies for RL-based systems.

The FilmTrust dataset's item overlapping across different biclusters is visualized in Figure 1(b). At minimum rows =10 and minimum columns =05, for Bibit algorithm, the number of biclusters found was 60,971 and contained overall 700,533 items. Figure 1(b) highlights that several movies appeared in more than 30,000 biclusters. Out of 2,071 movies of the dataset, 1560 unique movies appeared in one or more biclusters. This observation led us to reconsider the sorting criteria for biclusters. Instead of prioritizing biclusters based solely on their SMSR (Scaled Mean Square Residual) quality value as done in [1], we proposed sorting them based on their overlapping in terms of movies, specifically focusing on movie overlaps. Biclusters on the squared grid had minimum overlapping with each other. A quality measure like SMSR quantifies the homogeneity of values within a bicluster. We aimed to recommend items to the user, not their rating value so measuring homogeneity is irrelevant in our case. We computed item overlapping of biclusters to achieve our objectives. To compute item overlapping, we selected a bicluster having the largest movies set (designated as pivot bicluster) and determined item overlapping of this pivot bicluster from all other biclusters. In this way, we obtained a sorted list where biclusters having low overlapping came at the start, while biclusters having maximum overlapping came at the end of the sorted list. This sorted list is then used to place biclusters on a squared grid in a cantor diagonal fashion as shown in figure 2. Squared grid arrangements for both 6×6 (top 36 biclusters) and 7×7 grid arrangements (top 49 biclusters) are shown in Figure 2.

6X6 Grid

B0	B1	B5	B7	B13	B17
B2	B4	B6	B15	B14	B27
B3	B8	B9	B21	B16	B32
B10	B11	B20	B22	B31	B29
B12	B19	B24	B25	B26	B30
B18	B23	B33	B28	B34	B35

7X7 Grid

B0	B1	B5	B6	B14	B15	B27
B2	B4	B9	B11	B16	B26	B28
B3	B7	B13	B17	B25	B29	B38
B10	B12	B18	B24	B30	B37	B39
B8	B19	B23	B31	B36	B40	B45
B20	B22	B32	B35	B41	B44	B46
B21	B33	B34	B42	B43	B47	B48

Figure 2: 6×6 and 7×7 squared grid arrangements.

The flow diagram of the proposed methodology is shown in Figure 3. The complete methodology is composed of six steps. In step 1 biclusters are generated from the user-item ratings matrix. In step 2, grid size is chosen and in step 3, biclusters are sorted according to their overlapping of items. In step 4, seven similarity measures are applied to determine the start state one by one, and results are recorded in step 6. After gathering the results of all start state determination methods, the best method is selected for implementing an RL-based recommender system using a square grid environment.

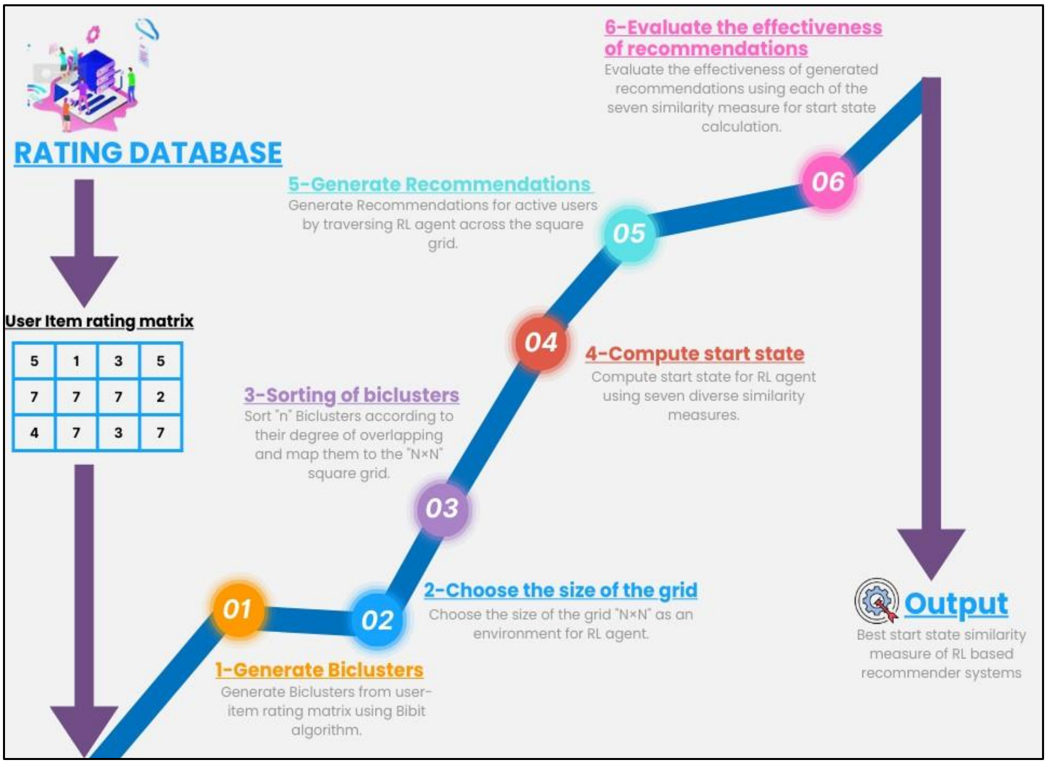


Figure 3: Flow Diagram of the proposed methodology

Mathematical detail of similarity measures is used to determine the start state. In 1998, Koutrika and Bercovitz introduced the Jaccard function for the calculation of the relationship between two users [19]. The Jaccard function only considers the number of co-rated items between two users without using the actual value of the ratings [11]. Despite several benefits of Jaccard measure, ignoring the actual rating value is a key drawback of Jaccard measure. Jaccard measure to compute the start state is given in Eq. (1).

$$Start_{(u,bic)}^{JACCARD} = \frac{|I_u \cap I_{bic}|}{|I_u \cup I_{bic}|} \quad (1)$$

Here I_u denotes items rated by user u and I_{bic} denotes items in a bicluster bic. $I_u \cap I_{bic}$ denotes commonly rated items of user u and bicluster bic, whereas $I_u \cup I_{bic}$ denotes the union of items of user u and bicluster bic. The cosine similarity measure is basically concerned with computing the angle difference of two ratings vectors [20].

$$Start_{(u,bic)}^{COSINE} = \frac{\Sigma_{i \in I_u \cap I_{bic}} (R_{u,i}) * (R_{bic,i})}{\sqrt{\Sigma_{i \in I_u} (R_{u,i})^2} * \sqrt{\Sigma_{j \in I_{bic}} (R_{bic,i})^2}} \quad (2)$$

In Eq. (2) $R_{u,i}$ denotes the rating score given by user u on item i and $R_{bic,i}$ denotes the average rating score of items i in bicluster bic. Pearson Correlation Coefficient (PCC) is a classical measure to compute similarities between two objects. Hereby, we utilized PCC to determine start state by computing the similarity of the target user with all biclusters in the squared grid environment and is given in Eq. (3).

$$Start_{(u,bic)}^{Pearson} = \frac{\Sigma_{i \in I_u \cap I_{bic}} (R_{u,i} - \overline{R}_u) * (R_{bic,i} - \overline{R}_{bic})}{\sqrt{\Sigma_{i \in I_u} (R_{u,i} - \overline{R}_u)^2} * \sqrt{\Sigma_{i \in I_{bic}} (R_{bic,i} - \overline{R}_{bic})^2}} \quad (3)$$

In Eq. (3) $\overline{R}_u$ denotes the average rating score of users u and $\overline{R}_{bic}$ denotes overall rating average of bicluster. In 2017, [21] introduced a similarity function named TMJ made from the combination of Triangle and Jaccard similarities/. The Triangle function uses the length and angle between two rating vectors while the Jaccard function considers the number of co-rated items as given in Eq. (4) and (5).

$$sim_{(u,bic)}^{TRIANGLE} = 1 - \frac{\sqrt{\sum_{i\in I_u \cap I_{bic}} (R_{u,i} - R_{bic,i})^2}}{\sqrt{\sum_{i\in I_u} {R_{u,i}}^2} + \sqrt{\sum_{i\in I_{bic}} {R_{bic,i}}^2}} \quad (4)$$

$$Start_{(u,bic)}^{TMJ} = sim_{(u,bic)}^{TRIANGLE} * sim_{(u,bic)}^{JACCARD} \quad (5)$$

Iftikhar et al. [13] introduced a triangle-based similarity metric (ITR), by utilizing both the ratings of common rated items and the uncommon items from pairs of users, while the user rating preference behavior is complemented by the obtained similarity in giving rating preferences. The proposed similarity metric achieves adequate accuracy when compared to existing similarity metrics in CF environment. The mathematical formulation of ITR is given from Eqn. (6) to Eqn. (9).

$$P = \{i \in I_u \cup I_{bic}\} \quad (6)$$

$$sim_{(u,bic)}^{TRIANGLE\prime} = 1 - \frac{\sqrt{\sum_{i\in P} (R_{u,i} - R_{bic,i})^2}}{\sqrt{\sum_{i\in P} {R_{u,i}}^2} + \sqrt{\sum_{i\in P} {R_{bic,i}}^2}} \quad (7)$$

$$sim_{(u,bic)}^{urp} = 1 - \frac{1}{1+exp(-|\overline{R}_u - \overline{R}_{bic}|.|\sigma_u - \sigma_{bic}|)} \quad (8)$$

$$sim_{(\mathrm{u,bic})}^{ITR} = sim_{(\mathrm{u,bic})}^{TRIANGLE\prime} * sim_{(\mathrm{u,bic})}^{urp} \quad (9)$$

In Eq. (8) σ_u denotes standard deviation in rating scores provided by the user u. Euclidian measure is a classical measure to compute the distance between two objects. We used Euclidian measure here to decide the start state for the target user. The mathematical formulation of Euclidian measure to compute the start state is given in Eq. (10) and Eq. (11).

$$ED_{(u,bic)} = \sqrt{\sum_{i\in I_u \cap I_{bic}} (R_{u,i} - R_{bic,i})^2} \quad (10)$$

$$Start_{(u,bic)}^{ED} = 1/ED_{(u,bic)} \quad (11)$$

The mathematical formulation of Manhattan measure to compute the start state is given in Eq. (12) and Eq. (13).

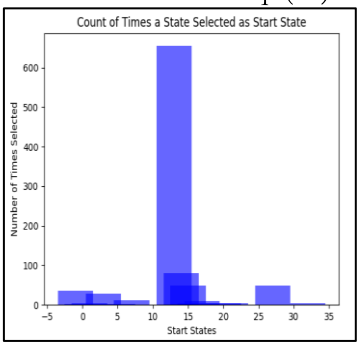
$$ManD_{(u,bic)} = \sum_{i\in I_u \cap I_{bic}} |R_{u,i} - R_{bic,i}| \quad (12)$$

$$Start_{(u,bic)}^{ManD} = 1/ManD_{(u,bic)} \quad (13)$$

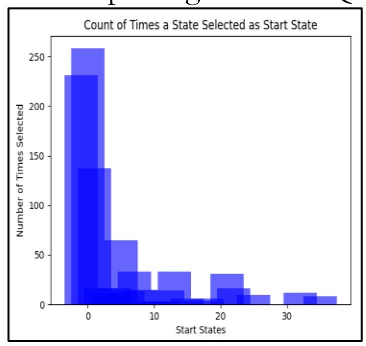
When a user seeking recommendations enters the environment, its rating vector is compared with all the biclusters of the grid environment, and the comparison result is stored in a list. This list is then sorted from maximum to minimum similarity. A bicluster having maximum similarity with user's rating vector is selected as the start state for the movement of the user within the grid environment. Reward function is given in Eq. (14). As we have no or minimum overlapping of items/ movies, so reward function computes only user overlapping in the current and next state. State having major overlapping was preferred. In Eq. (14) U_{st} and U_{st+1} denotes users in state st and $st + 1$.

$$Reward\ (R) = \frac{|U_{st} \cap U_{st+1}|}{|U_{st} \cup U_{st+1}|} \tag{14}$$

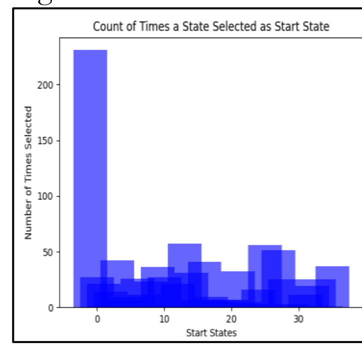
To balance the exploration and exploitation tradeoff, we used Epsilon-greedy algorithm [22]. Epsilon-greedy algorithm performs large exploration in the start and then decreases exploration and increases exploitation. Q-learning was used to update the quality values of each action in each state. Eq. (15) denotes the updating model of Q-learning.



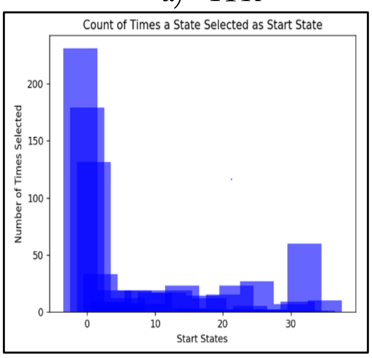
a) ITR



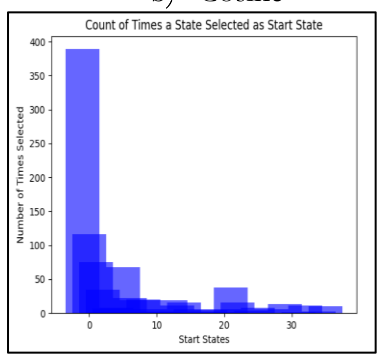
b) Cosine



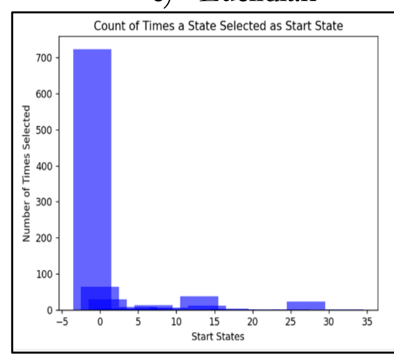
c) Euclidian



d) Jaccard



e) PCC



f) TMJ

Figure 4 (a-f): ML-100K dataset, Grid size=6×6, Depiction of how many times a state is selected as start state by a particular similarity measure

$$Q(s,a) = Q(s,a) + \alpha * (Reward + \Upsilon * \max(Q(s',a')) - Q(s,a)) \tag{15}$$

In Eq. (15) $Q(s,a)$ represents the expected reward for taking action a in state s. The actual reward received for that action was referenced by R while s' refers to the next state. The learning rate is α and γ is the discount factor. The highest expected reward for all possible actions a′ in state s' is represented by $max(Q(s',a'))$.

Objective of RL is to find the optimal policy $\pi^*(s)$ that maximizes the expected cumulative reward. In RL, the optimal policy can be learned by a state-action value function $Q_\pi(s,a)$which means the expected value of the cumulative reward obtained from episodes starting from a certain start state s with the action a. $Q_\pi(s,a)$can be expressed as follows:

$$Q_\pi(s,a) = E_\pi\{\sum_{k=0}^{\infty} \gamma^k R_{t+k} | s_t = s, a_t = a\} \tag{16}$$

Here π is the policy and k is the number of episodes, γ denotes a discount factor on reward, having value 0 to 1. Discount factor γ gives more emphasis to current reward value and suppresses past reward value. Optimal policy $\pi^*(s)$ is a set of actions having highest Q value. Each action results in a state visit, where each state represents a bicluster, thus items belonging to the item set of that bicluster are recommended to the user.

$$\pi^*(s) = argmax_{a \in A} Q_\pi(s,a) \tag{17}$$

Observation, Analysis, and Results:

The following evaluation measures were used for the analysis of the performance of different start states on both sized grid arrangements for used datasets.

State Selection Count:

This measure assesses the robustness of a particular start state selection method by counting how many times each state is selected as the start state. State selection count for each selected state by the particular measure is given by the following notation, {State selected: State count}, for example {2:21} indicates that state 2 is selected as 21 times as the start state. A state value of -1 indicates failure in the determination of the start state for a particular user. For example, {-1:20} indicates for 20 users, the proposed system is unable to determine a start state.

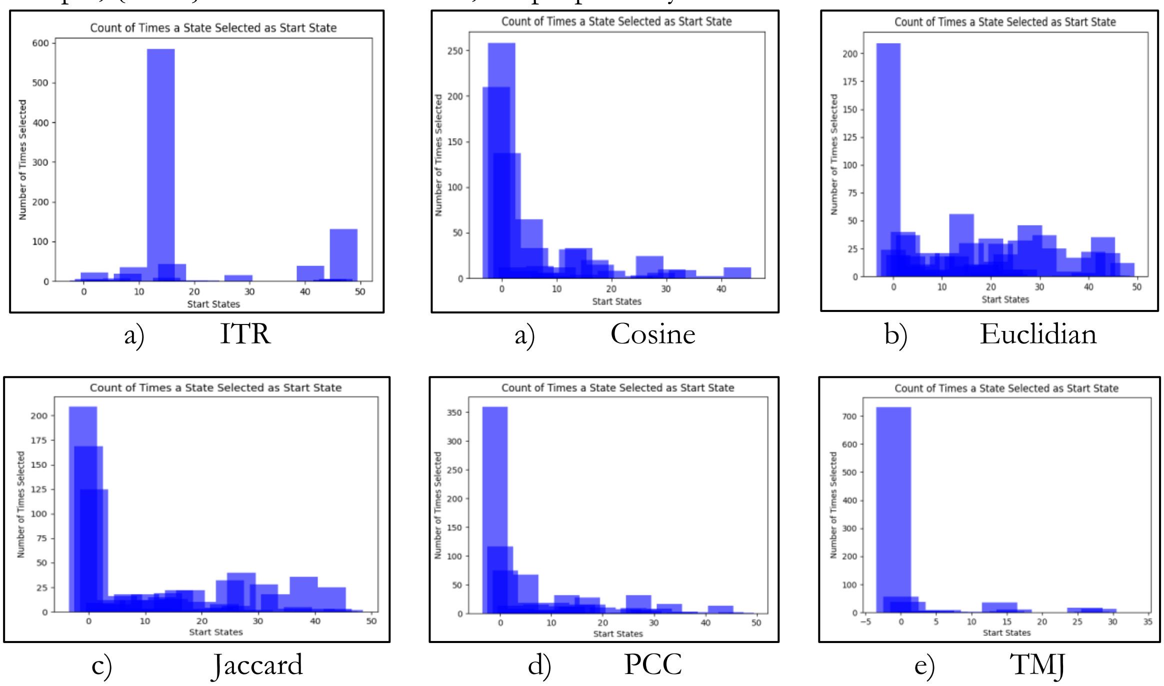


Figure 5 (a-f): For ML-100K dataset, Grid size=7×7, Depiction of how many times a state is selected as start state by a particular similarity measure

User Coverage:

Indicates the percentage of users for which the proposed method was able to generate recommendations.

Item Coverage:

Indicates the percentage of items for which our proposed method was able to generate recommendations. A state value of -1 indicates that RL agent is unable to determine a start state using a particular similarity function, thus no recommendations are being generated for that user, thus decreasing user coverage and item coverage.

Precision:

Precision is the amount of overlap between predicted items and actual items of test user w.r.t to predicted items set. A higher value of precision is desirable for better efficiency of a method.

Recall:

A recall is the amount of overlap between the predicted item set and the actual items set of the test user w.r.t actual items set of the test user.

F-Measure: To better understand the results of precision and recall, F-measure is used which is the harmonic mean of both measures.

Return:

Return is the reward earned by the RL agent by applying the learned policy. A greater return value indicates more efficiency of the algorithm.

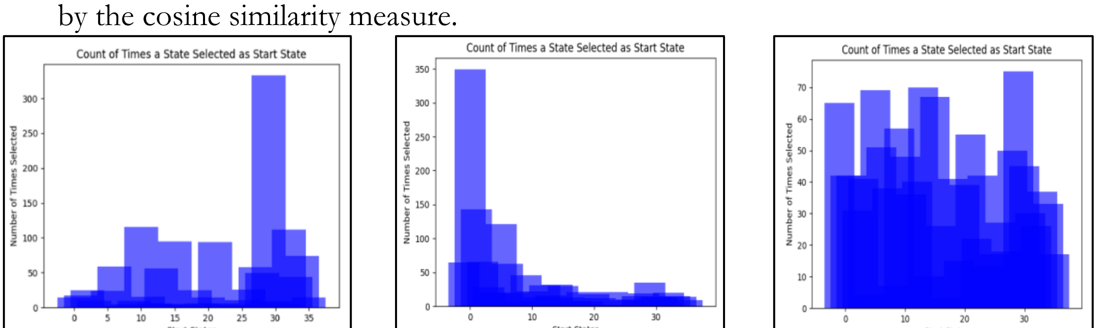
Measuring the Impact of State Selection Count for Different Start States:

This section gives an insight into how different states were selected as start states when the start state is determined by a particular similarity measure. As Euclidian and Manhattan measure possess almost similar results, so in figures 3,4,5, and 6 only Euclidian measure results are shown. Observations and analysis for ML-100K dataset regarding state selection count on 6×6 Grid Size (figure 4) are below.

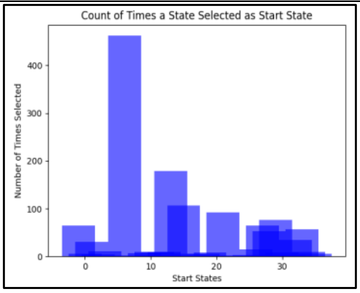
- For ITR measure (figure 4(a)), had as low as 35 counts of -1 state, resulting in user coverage of 96.28% and item coverage of 92.74%. State 13 was selected highly as 655 times as start state for 655 users out of 943 users. States which were selected as start state along with state selection count are {13: 655 ,14: 81 ,27: 48 ,21: 4 ,17: 9 ,3: 29 ,15: 48, -1: 35 ,16: 4 ,32: 4 ,0: 2 ,20: 5 ,7: 11 ,1: 5 ,9: 1 ,5: 2}. In total 16 states were selected out of 37 states implying that 21 states were not selected by the ITR similarity measure.
- For Cosine similarity (figure 4(b)), state -1 had highest count of 231, implying a failure to generate recommendations for 231 (24.95%) users out of 943 also resulting in a reduced item coverage of 71.64. Most frequent selected state was state 0 with a highest count of 258. Start state selection count for cosine measure = {5: 8 ,14: 23 ,-1: 231 ,1: 131 ,0: 179 ,7: 19 ,33: 1 ,32: 60 ,17: 14 ,25: 27 ,13: 19 ,2: 33 ,12: 17 ,4: 19 ,22: 23 ,31: 9,24: 5 ,6: 12 ,8: 19 ,16: 2 ,10: 17 ,18: 12 ,26: 2 ,9: 7 ,23: 2 ,3: 9 ,20: 15 ,29: 1 ,35: 10 ,27: 2 ,30: 7 ,34: 1 ,19: 1 ,15: 3 ,28: 1 ,21: 2 }. Except state 11, all other states were selected got the chance of selection as start state.
- Euclidian and Manhattan (figure 4(c)) showed almost similar performance. Both have highest count of 231 for state -1, implying failure to generate recommendations for 231 users (24.95%). As both methods have similar performance so this makes it clear that we can use any one when we have to choice from both. For Manhattan measure, state selection count = { 14: 31 ,1: 21 ,-1: 231 ,25: 56 ,27: 51 ,22: 5 ,18: 5 ,13: 57 ,35: 37 ,20: 7 ,17: 9 ,6: 26 ,0: 27 ,7: 11 ,16: 40 ,3: 42 ,2: 15 ,33: 25 ,8: 23 ,10: 24 ,28: 25 ,31: 12 ,12: 21 ,21: 30 ,5: 9 ,9: 37 ,19: 7 ,4: 5 ,32: 21 ,15: 7 ,29: 5 ,24: 16 ,23: 3 ,34: 1 ,11: 1 ,}. For Euclidian measure, state selection count = { 14: 31 ,1: 21 ,-1: 231 ,25: 56 ,27: 51 ,22: 5 ,18: 5 ,13: 57 ,35: 37 ,20: 7 ,17: 9 ,6: 26 ,0: 27 ,7: 11 ,16: 41 ,3: 42 ,2: 14 ,33: 25 ,8: 23 ,10: 27 ,28: 25 ,31: 11 ,12: 21 ,21: 32 ,5: 9 ,9: 36 ,19: 7 ,4: 6 ,32: 19 ,15: 7 ,29: 3 ,24: 16 ,23: 3 ,34: 1 ,11: 1 }. Both methods were unable to select state 26 and 30 as start state.
- TMJ (figure 4(f)) and PCC (figure 4(e)) similarity proved to be worst. TMJ was Unable to generate recommendations for 722 users (76.57%). Thus, giving a user coverage of 23.43% and item coverage of 21.94%. After state -1, highly selected state is state 0, with a state count of 64. For PCC measure count of -1 states was found to be 389 leading to a user coverage and item coverage of 58.74% and 55.46%. For TMJ measure, state selection count = {5: 9 ,1: 29,-1: 722 ,3: 4 ,17: 4 ,13: 37 ,0: 64 ,9: 6 ,14: 11 ,2: 3 ,27: 23 ,32: 2 ,25: 1 ,7: 13 ,21: 2 ,8: 3 ,6: 5 ,4: 2 ,16: 2 ,15: 1 }. For PCC measure, state selection count ={1: 75 ,-1: 389 ,14: 16 ,24: 4 ,0: 116 ,4: 8 ,20: 6 ,6: 22 ,25: 8 ,2: 35 ,7: 19 ,5: 68 ,12: 19 ,27: 6 ,29: 13 ,21: 38 ,13: 11 ,32: 11 ,8: 20 ,35: 10 ,15: 6 ,18: 5 ,22: 16 ,3: 7 ,10: 2 ,19: 1

,34: 2 ,17: 2 ,9: 6 ,11: 1 ,31: 1}. In total 31 unique states are selected by the PCC measure and 20 unique states are selected by the TMJ measure.

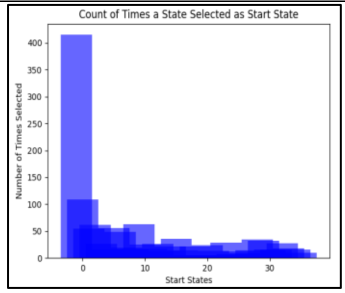
- For Jaccard measure (figure 4(d)), state -1 had a count of 231, thus achieving a user coverage 75.50%. This is almost same as compared to cosine measure. Also both cosine and Jaccard achieved an item coverage of 71.64%. But Jaccard only computes number of common items and cosine computes difference in angle of rating vectors common items. Similar performance of both here implies that it makes no difference whether we use item intersection or rating values. State selection count of Jaccard measure = {5: 8 ,14: 23 ,-1: 231 ,1: 131 ,0: 179 ,7: 19 ,33: 1 ,32: 60 ,17: 14 ,25: 27 ,13: 19 ,2: 33 ,12: 17 ,4: 19 ,22: 23 ,31: 9 ,24: 5 ,6: 12 ,8: 19 ,16: 2 ,10: 17 ,18: 12 ,26: 2 ,9: 7 ,23: 2 ,3: 9 ,20: 15 ,29: 1 ,35: 10 ,27: 2 ,30: 7 ,34: 1 ,19: 1 ,15: 3, 28: 1 ,21: 2 }. Only state 11 remain unselected, all other states get selected by Jaccard measure, but on the other hand cosine was able to select 23 states.
- Thus we observe that for ML-100K dataset, 6×6 Grid size, overall for all seven measures if we sort their inability to determine start state then we have following {ITR: 35, Cosine: 231, Euclidian:231, Manhattan:231, Jaccard:231, PCC:389, TMJ:722, }. As ITR takes into account complete set of items of both target user and bicluster, so it is indecisive for just 35 times. As all other measures work on co-rated items which may be hard to be available in both target user rating vector and bicluster, so they are unable to select a start state many times.
- Following is observed regrading state selection count for ML-100K dataset, 7×7 Grid size (figure 5).
- For ITR measure (figure 5(a)), had as low as 52 counts of -1 state, resulting in user coverage of 94.48% and item coverage of 90.91%. State 14 was selected highly as 615 times as start state for 615 users out of 943 users. Out of 50 states only 17 states get selected as start state. These states along with their count are {14: 615 ,16: 87 ,28: 38 ,17: 5 ,15: 8 ,-1: 52 ,2: 29 ,3: 31 ,9: 54 ,13: 1 ,0: 2 ,6: 11 ,1: 5 ,26: 1 ,5: 2 ,10: 1 ,7: 1 }. Remaining 33 states, which are two third of total states didn't get selected as start state.
- For Cosine similarity (figure 5(b)), state 0 had highest count of 258, after which comes state -1 with a state selection count of 210, implying a failure to generate recommendations for 210 users (22.26%) out of 943, resulting in a reduced user coverage of 77.73% and item coverage of 73.93%. 27 states were selected as start state having a selection state count = {0: 258 ,-1: 210 ,1: 137 ,14: 33 ,27: 24 ,6: 33 ,31: 10 ,43: 12 ,5: 65 ,18: 15 ,11: 12 ,33: 9 ,7: 13 ,29: 6 ,13: 32 ,8: 6 ,17: 20 ,3: 1 ,10: 2 ,9: 6 ,2: 12 ,4: 8 ,38: 2 ,20: 8 ,15: 4 ,26: 3 ,25: 2 }. This implies that 23 states can't be selected as start state by the cosine similarity measure.



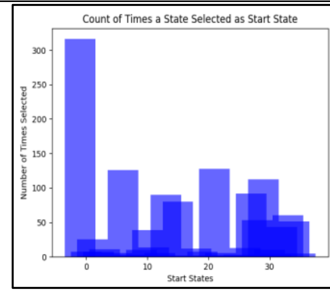
a) ITR b) Cosine c) Euclidian



d) Jaccard



e) PCC



f) TMJ

Figure 6 (a-f): For the Film Trust dataset, Grid size=6×6, Depiction of how many times a state is selected as start state by a particular similarity measure

- The performance of Euclidian and Manhattan (figure 5(c)), is almost similar as in case of 6x6 Grid environment. Both have highest count of 209 for state -1, implying failure to generate recommendations for 209 users (22.17%) and yielding an item coverage of 74.25%. As both methods have similar performance so this make it clear that we can use any one when we have to choice from both. For both measures 42 states out of 50 were selected as start sate, having a state selection count= {16: 29 ,1: 19, -1: 209 ,41: 22 ,28: 45 ,35: 1 ,8: 5 ,14: 56 ,20: 34 ,25: 2 ,15: 9 ,11: 21 ,43: 35 ,33: 25 ,6: 11 ,3: 37 ,47: 38 ,2: 40 ,0: 24 ,44: 21 ,7: 21 ,21: 13 ,23: 20 ,24: 6 ,13: 18 ,17: 9 ,5: 9 ,26: 32 ,39: 3 ,27: 6 ,10: 18 ,4: 17 ,31: 37 ,19: 10 ,9: 6 ,38: 17 ,22: 8 ,36: 4 ,18: 3 ,45: 1 ,46: 1 ,12: 1}.
- TMJ (figure 5(f)) and PCC similarity (figure 5(e)) proved to be worst. TMJ was Unable to generate recommendations for 731 users (77.51%). Thus giving a user coverage of 22.49% and item coverage of 20.74%. After state -1, highly selected state is state 0, with a state count of 57. Selected states count for TMJ= { -1: 731 ,1: 37 ,2: 4 ,14: 35 ,0: 57 ,26: 18 ,16: 11 ,4: 2 ,5: 7 ,6: 9 ,28: 12 ,10: 1 ,3: 5 ,31: 1 ,17: 1 ,15: 4 ,7: 3 ,27: 2 ,11: 2 ,8: 1}. Only 20 states out of 50 were selected by TMJ measure. For PCC measure count of -1 states was found to be 359 leading to a user coverage and item coverage of 61.93% and 58.60%. After state -1, state 0 is highly selected state with a count of 116. State selection count of PCC measure = {1: 75 ,-1: 359 ,16: 16 ,36: 3 ,0: 116 ,27: 32 ,25: 7 ,11: 19 ,43: 13 ,31: 8 ,6: 19 ,5: 68 ,13: 32 ,33: 17 ,28: 6 ,12: 12 ,14: 11 ,29: 4 ,47: 1 ,7: 16 ,20: 10 ,2: 13 ,8: 5 ,18: 15 ,10: 4 ,17: 28 ,21: 1 ,9: 6 ,4: 8 ,26: 9 ,38: 2 ,45: 2 ,15: 2 ,3: 2 ,24: 1 ,35: 1}. In this state selection count of PCC measure 36 unique states out of 50 got selected.
- For Jaccard measure (figure 5(d)), state -1 has a count of 209, thus achieving a user coverage 77.83%. This is almost same as compared to cosine measure. A lesser state count of state -1 here (i.e. 209) as compared to 231 for 6x6 grid implies that increasing Grid size may result in better user coverage. State selection count of Jaccard measure = { 5: 8 ,16: 22 ,-1: 209 ,1: 125 ,0: 169 ,33: 18 ,19: 1 ,38: 36 ,15: 13 ,31: 28 ,43: 25 ,14: 19 ,27: 40 ,4: 12 ,13: 15 ,18: 22 ,24: 8 ,11: 12 ,7: 18 ,3: 2 ,12: 3 ,10: 18 ,6: 16 ,44: 3 ,8: 12 ,30: 2 ,26: 6 ,37: 5 ,25: 32 ,2: 9 ,20: 10 ,41: 3 ,28: 2 ,21: 3 ,36: 3 ,23: 4 ,46: 2 ,45: 1 ,39: 1 ,9: 3 ,17: 2 ,40: 1}. Here Jaccard measure was able to select 42 unique states a start state, but on the other hand cosine was able to select 27 unique states.
- Thus we observe that for ML-100K dataset, 7×7 Grid size, overall for all seven measures if we sort their inability to determine start state then we have following {ITR: 52, Cosine: 210, Euclidian:209, Manhattan:209, Jaccard:209, PCC:359, TMJ:731}. As ITR takes into account complete set of items of both target user and bicluster, so it is indecisive for just 52 users. As all other measures work on co-rated items which may be hard to be available in both target user rating vector and bicluster, so they are unable to select a start state many times than ITR.

Following observations are made regrading state selection count for FilmTrust dataset 6×6 Grid size (figure 6).

- For all users ITR (figure 6(a)), was able to determine a start state, implying 0 count of state -1 and resulting in a user coverage of 100.00% and item coverage of 91.41%. State 29 was selected highly as 334 times as start state for 334 users out of 1349 users. Out of 36 states 34 were selected as start states for different test users whereas state 4, and state 17 remained un-selected. These selected states along with their count are {31: 46 ,20: 14 ,10: 117 ,29: 334 ,15: 93 ,28: 49 ,32: 120 ,13: 44 ,21: 117 ,11: 15 ,5: 24 ,33: 44 ,7: 6 ,27: 59 ,26: 26 ,6: 58 ,9: 12 ,2: 28 ,12: 6 ,18: 27 ,23: 1 ,25: 13 ,1: 17 ,3: 12 ,35: 16 ,8: 6 ,14: 7 ,16: 5 ,0: 14 ,24: 3 ,22: 3 ,30: 9 ,34: 3 ,19: 1}.
- For Cosine similarity (figure 6(b)), state 0 has highest count of 349, after which comes state 1 with a state selection count of 143. State -1 had a count of 65 implying a failure to generate recommendations for 65 users (4.81%) out of 1349, resulting in a user coverage of 95.19% and item coverage of 86.57%. All 37 states including -1 state were selected as start state having a selection state count = { 2: 66 ,17: 8 ,5: 121 ,0: 349 ,1: 143 ,34: 14 ,16: 13 ,-1: 65 ,6: 63 ,15: 30 ,29: 35 ,9: 46 ,26: 10 ,18: 21 ,30: 14 ,13: 16 ,27: 15 ,25: 9 ,14: 32 ,12: 31 ,33: 15 ,7: 20 ,11: 13 ,23: 21 ,28:19 ,3: 28 ,4: 20 ,20: 18 ,8: 7 ,21: 12 ,31: 13 ,10: 22 ,32: 20 ,35: 9 ,19: 7 ,22: 2 ,24: 2}.
- Performance of Euclidian and Manhattan measures (figure 6(c)), is almost similar as previously found in case of ML-100K dataset. Both have a count of 65 for state -1, implying failure to generate recommendations for 65 users (4.81%) and yielding an item coverage of 86.59%. As both methods have similar performance so this make it clear that we can use any one when we have to choice from both. State count of Euclidian and Manhattan measure = {29: 73 ,20: 42 ,23: 36 ,4: 12 ,30: 47 ,32: 26 ,13: 88 ,21: 45 ,8: 8 ,5: 70 ,28: 55 ,9: 53 ,10: 51 ,12: 38 ,-1: 65 ,31: 12 ,6: 51 ,0: 41 ,3: 51 ,15: 60 ,2: 32 ,11: 33 ,34: 34 ,16: 28 ,7: 21 ,22: 28 ,25: 19 ,24: 13 ,18: 48 ,17: 3 ,33: 43 ,35: 16 ,26: 29 ,1: 34 ,27: 16 ,19: 18 ,14: 10 }
- TMJ (figure 6(f)) and PCC (figure 6(e)), similarity proved to be worst. TMJ was Unable to generate recommendations for 345 users (26.58%). Thus giving a user coverage of 74.42% and item coverage of 66.43%. Selected states count for TMJ= { 15: 76 ,-1: 345 ,13: 77 ,21: 138 ,5: 6 ,27: 97 ,28: 53 ,6: 97 ,10: 38 ,29: 109 ,11: 16 ,33: 60 ,14: 7 ,3: 12 ,32: 48 ,26: 13 ,31: 47 ,23: 2 ,35: 6 ,18: 13 ,1: 24 ,7: 8 ,0: 9 ,30: 10 ,8: 3 ,9: 10 ,25: 5 ,22: 4 ,2: 4 ,12: 6 ,34: 2 ,20: 2 ,19: 1 ,16: 1 ,}. 34 states out of 37 are selected by TMJ measure. For PCC measure count of -1 states was found to be 415 leading to a user coverage and item coverage of 69.23% and 61.45%. After state -1, state 0 is highly selected state with a count of 109. State selection count of PCC measure = {9: 62 ,-1: 415 ,5: 56 ,0: 109 ,1: 55 ,16: 14 ,3: 26 ,6: 48 ,26: 8 ,15: 36 ,34: 27 ,18: 22 ,29: 36 ,21: 11 ,32: 26 ,8: 17 ,2: 61 ,14: 17 ,33: 18 ,7: 11 ,28: 35 ,11: 17 ,23: 27 ,20: 23 ,12: 27 ,4: 21 ,24: 7 ,10: 25 ,13: 19 ,30: 16 ,22: 7 ,27: 14 ,19: 6 ,17: 5 ,25: 12 ,35: 8 ,31: 5 }. All states were selected by PCC measure as start state and no state is left over.
- For Jaccard measure (figure 6(d)), state -1 had a count of 65, thus achieving a user coverage 95.18%. This is almost same as compared to cosine measure. Also both cosine and Jaccard achieved an item coverage of 86.60%. State selection count of Jaccard measure = {6: 462 ,13: 179 ,15: 106 ,32: 34 ,21: 92 ,-1: 65 ,28: 53 ,11: 9 ,33: 57 ,27: 65 ,29: 77 ,10: 9 ,3: 11 ,26: 15 ,12: 9 ,1: 31 ,23: 1 ,35: 5 ,19: 8 ,31: 3 ,34: 10 ,16: 5 ,0: 6 ,30: 8 ,8: 2 ,14: 2 ,7: 1 ,9: 7 ,25: 3 ,22: 1 ,2: 4 ,5: 2 ,18: 6 ,20: 1 } showing that state 6 is highly selected state . 32 unique states out of 37 are selected by Jaccard measure, but on the other hand cosine selected all 37 states as start state.

- Thus, we observed that for the FilmTrust dataset, 6×6 Grid size, overall, for all seven measures if we sort their inability to determine start state then we have the following {ITR: 0, Cosine: 65, Euclidian:65, Manhattan:65, Jaccard:65, TMJ:345, PCC:415}. As ITR takes into account a complete set of items of both target user and bicluster, so it is able to select the start state for all users. As all other measures work on co-rated items which may be hard to be available in both target user rating vector and bicluster, so they are unable to select a start state for many test users.

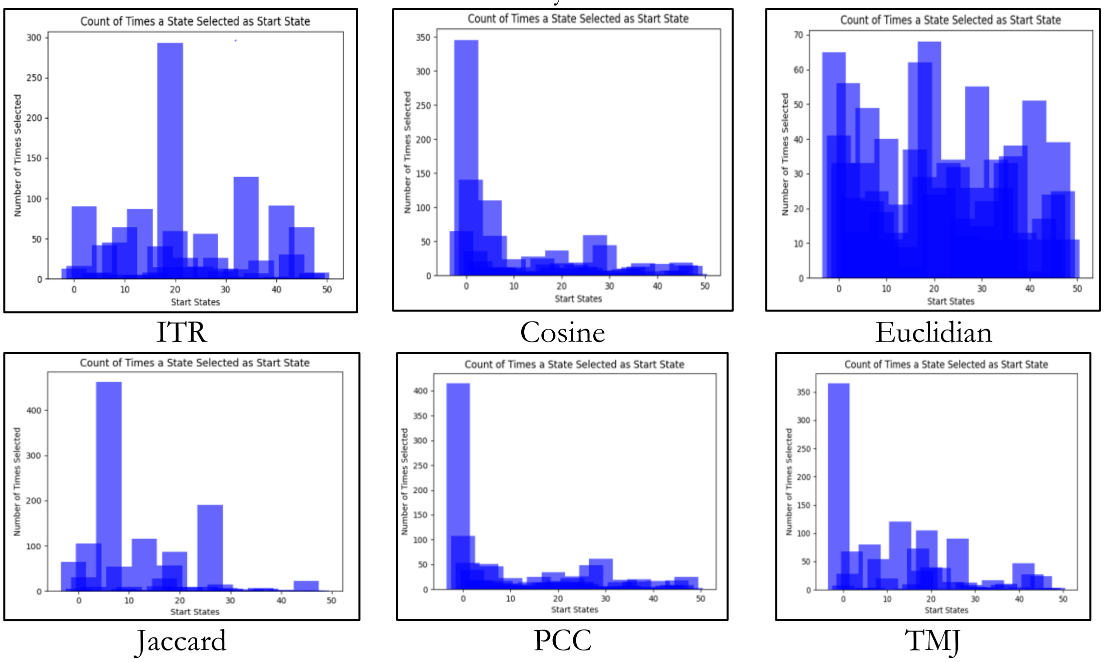


Figure 7 (a-f): For Film Trust dataset, Grid size=7×7, Depiction of how many times a state is selected as start state by a particular similarity measure

Following is observed regrading state selection count for Film Trust dataset 7×7 Grid size (figure 7).

- For all users ITR (figure 7(a)), was able to determine a start state, implying 0 count of state -1 and resulted in a user coverage of 100.00% and item coverage of 91.26%. State 19 was selected highly as 293 times as start state for 293 users out of 1349 users. Out of 50 states, 45 were selected as start states for different test users whereas state {-1, 11, 15, 38, 42} remained un-selected. These selected states along with their count are {22: 26 ,23: 14 ,34: 127 ,19: 293 ,2: 90 ,10: 64 ,8: 45 ,41: 91 ,17: 40 ,13: 87 ,25: 15 ,5: 8 ,43: 30 ,20: 59 ,26: 56 ,45: 64 ,28: 26 ,6: 42 ,29: 9 ,18: 14 ,48: 8 ,37: 23 ,24: 1 ,4: 7 ,1: 16 ,16: 8 ,32: 12 ,40: 3 ,9: 5 ,0: 13 ,47: 7 ,30: 13 ,14: 5 ,44: 1 ,27: 2 ,31: 3 ,39: 2 ,12: 4 ,7: 1 ,35: 2 ,21: 2 ,36: 7 ,3: 1 ,33: 1 ,46: 2}
- For Cosine similarity (figure 7(b)), state 0 has highest count of 349, after which comes state 1 with a state selection count of 143. State -1 has a count of 65 implying a failure to generate recommendations for 65 users (4.81%) out of 1349, resulting in a user coverage of 95.19% and item coverage of 86.50%. State selection count of selected states= {27: 59 ,15: 6 ,5: 110 ,0: 345 ,1: 140 ,35: 12 ,9: 10 ,-1: 65 ,41: 17 ,6: 58 ,2: 35 ,19: 36 ,29: 44 ,28: 7 ,37: 18 ,36: 13 ,17: 16 ,26: 11 ,12: 6 ,14: 28 ,47: 14 ,18: 11 ,7: 12 ,25: 13 ,24: 19 ,44: 7 ,8: 12 ,16: 25 ,11: 12 ,23: 16 ,4: 9 ,20: 18 ,22: 10 ,3: 21 ,33: 9 ,10: 24 ,45: 19 ,43: 7 ,34: 10 ,30: 7 ,13: 5 ,38: 6 ,46: 15 ,32: 4 ,31: 1, 48: 2 ,21: 4 ,39: 1}. Except state 40 and 42 all other states are selected as start state.
- Performance of Euclidian and Manhattan (figure 7(c)), was almost similar as found previously in case of ML-100K dataset. Both have a count of 65 for state -1, implying

failure to generate recommendations for 65 users (4.81%) and yielding an item coverage of 86.59%. Both methods showed similar performance. State count of Manhattan measure and Euclidian measure ={19: 68 ,23: 33 ,41: 51 ,35: 33 ,48: 11 ,45: 24 ,11: 11 ,46: 39 ,4: 13 ,5: 33 ,26: 17 ,29: 55 ,10: 40 ,44: 11 ,12: 13 ,-1: 65 ,6: 49 ,0: 41 ,39: 13 ,17: 62 ,2: 56 ,34: 26 ,33: 34 ,25: 32 ,20: 24 ,9: 19 ,8: 25 ,7: 22 ,31: 22 ,24: 34 ,16: 37 ,15: 3 ,18: 29 ,27: 12 ,37: 38 ,30: 15 ,13: 21 ,36: 35 ,3: 23 ,22: 26 ,28: 26 ,47: 25 ,1: 33 ,32: 11 ,43: 17 ,38: 11 ,14: 9 ,40: 2 }. Both were able to select 48 unique states as start state.

- TMJ (figure 7(f)), and PCC (figure 7(e)), similarity proved to be worst. TMJ was Unable to generate recommendations for 365 users (27.06%). Thus, giving a user coverage of 72.94% and item coverage of 64.87%. Selected states count for TMJ = {2: 68 ,-1: 365 ,17: 73 ,13: 120 ,26: 90 ,8: 54 ,6: 80 ,34: 17 ,19: 105 ,25: 14 ,43: 27 ,20: 40 ,5: 3 ,10: 20 ,16: 10 ,45: 24 ,28: 13 ,18: 33 ,22: 39 ,24: 1 ,41: 47 ,15: 1 ,30: 5 ,37: 11 ,1: 28 ,42: 2 ,40: 4 ,0: 7 ,36: 9 ,32: 4 ,47: 3 ,14: 2 ,7: 3 ,29: 8 ,31: 4 ,48: 3 ,35: 2 ,33: 2 ,3: 1 ,23: 2 ,46: 2 ,38: 1 ,44: 2} . 43 states out of 50 were selected by TMJ measure. For PCC measure count of -1 states was found to be 415 leading to a user coverage and item coverage of 69.23% and 61.39%. After state -1, state 0 is highly selected state with a count of 109. State selection count of PCC measure = { 29: 61 ,-1: 415 ,5: 51 ,0: 107 ,1: 53 ,9: 12 ,16: 25 ,6: 46 ,28: 5 ,2: 39 ,46: 9 ,37: 20 ,19: 35 ,22: 13 ,41: 17 ,32: 5 ,27: 48 ,14: 14 ,47: 25 ,18: 14 ,7: 7 ,25: 16 ,24: 26 ,23: 23 ,44: 8 ,8: 17 ,11: 10 ,21: 6 ,34: 19 ,4: 18 ,3: 18 ,20: 17 ,10: 23 ,39: 5 ,17: 13 ,33: 14 ,45: 18 ,36: 15 ,13: 6 ,31: 5 ,43: 7 ,26: 11 ,38: 4 ,15: 4 ,48: 3 ,30: 6 ,35: 7 ,12: 8 ,40: 1 ,}. All states were selected by PCC measure as start state and no state is left over.
- For Jaccard measure (figure 7(d)), state -1 has a count of 65, thus achieving a user coverage 95.18%. This is similar to cosine measure. The 6X6 grid arrangement also had a state count of 65 for -1, which implies that increased size of grid environment has no effect on determining start state. State selection count of Jaccard measure = { 6: 462 ,17: 28 ,2: 105 ,13: 116 ,19: 87 ,-1: 65 ,8: 54 ,25: 9 ,18: 57 ,26: 191 ,10: 9 ,16: 11 ,28: 15 ,3: 9 ,1: 31 ,24: 1 ,30: 5 ,22: 8 ,40: 3 ,20: 9 ,9: 5 ,0: 6 ,36: 7 ,32: 2 ,47: 1 ,14: 2 ,7: 1 ,29: 7 ,12: 3 ,45: 23 ,31: 1 ,34: 4 ,41: 2 ,37: 6 ,33: 1 ,23: 1 ,44: 1 ,46: 1 ,}. Here Jaccard measure was able to select 38 unique states a start state, but on the other hand cosine was able to select 48 unique states out of 50.
- We observed that for the FilmTrust dataset and a 7×7 grid size, when sorting the measures based on their inability to determine the start state, we have the following results: {ITR: 0, Cosine: 65, Euclidean: 65, Manhattan: 65, Jaccard: 65, TMJ: 365, PCC: 415}. The ITR measure, which considers the complete set of items of both the target user and bicluster, was able to select the start state for all users. In contrast, the other measures work on co-rated items, which may not be readily available in both the target user's rating vector and bicluster, leading to their inability to select a start state for many test users..

Measuring Impact of Different Start States on Performance:

Table 1 shows impact of different start states on performance of proposed system. For ML-100K dataset 6×6 grid, ITR gave best performance for all evaluation measures. Worst performance was reported by the TMJ and PCC measures, While TMJ and PCC measures also earned lowest reward. Whereas performance of Euclidian, Manhattan, cosine and Jaccard was almost similar. For Grid size 7×7 we observed that, user coverage, item coverage and return decreased as compared to 6×6 grid size, while precision, recall and F-measure yielded better value as compared to 6×6 grid size, for ITR measure. For Euclidian, Manhattan, Cosine, Jaccard and PCC all evaluation measures result of 7×7 grid is better than 6×6 grid size. TMJ performance for evaluation measures was decreased when we increased grid size from 6×6 to 7×7. One reason for the better performance of ITR as compared to other measures is its ability to work on non-co-rated items. While all other measures work on co-rated items and availability of co-rated items across biclusters is minimal.

For the Film Trust dataset, ITR measures user coverage and item coverage is the same for both grid sizes implying that grid size has no effect on these parameters. But the return of small small-sized grid (6×6) is greater than a large-sized grid (7×7). Precision, Recall, and F-measure results of 7×7 grid are better than 6×6 grid implying larger grid size can improve accuracy. Performance of Cosine, Euclidian, Manhattan, and Jaccard is almost similar with each other and for both sizes We can also observe that for a similarity measure like ITR, giving better user coverage can also increase item coverage, thus in turn resulting in a better return value. On both datasets when grid size is increased Return decreases. One reason for this can be regarded by the fact that as the number of biclusters in the grid environment increases either reward value decreases or the number of steps taken in the optimal policy may be decreased. But this observation can be strengthened on another dataset. The return value of ITR, Cosine, Euclidian, Manhattan and TMJ increases for the large-sized grid on ML-100K dataset. Whereas increased grid size on Film Trust dataset decreases return value.

Table 1: Evaluation measure results

ML-100K Dataset, Grid Size=6x6							
	Similarity Measure						
	ITR	**Cosine**	**Euclidian**	**Manhattan**	**Jaccard**	**PCC**	**TMJ**
User Coverage	96.288	75.504	75.504	75.504	75.504	58.749	23.436
Item Coverage	92.748	71.692	71.902	71.791	71.647	55.461	21.943
Precision	49.541	47.812	49.602	49.712	48.857	47.287	47.493
Recall	78.813	77.330	78.112	76.561	78.255	75.606	74.593
F-Measure	58.829	57.253	58.830	57.593	58.166	56.510	56.619
Return	10.546	7.634	7.939	7.895	7.757	6.006	2.434
ML-100K Dataset, Grid Size=7x7							
	ITR	**Cosine**	**Euclidian**	**Manhattan**	**Jaccard**	**PCC**	**TMJ**
User Coverage	94.486	77.731	77.837	77.837	77.837	61.930	22.163
Item Coverage	90.914	73.932	74.258	74.183	73.962	58.606	20.747
Precision	51.572	50.799	50.579	50.653	50.875	50.324	48.416
Recall	80.440	79.139	78.835	78.753	79.121	77.592	74.385
F-Measure	60.904	59.177	58.962	59.009	59.207	58.526	56.521
Return	9.954	7.561	8.254	8.300	7.790	6.224	2.210
Film Trust Dataset, Grid Size = 6x6							
	ITR	**Cosine**	**Euclidian**	**Manhattan**	**Jaccard**	**PCC**	**TMJ**
User Coverage	100.000	95.182	95.182	95.182	95.182	69.236	74.426
Item Coverage	91.416	86.575	86.591	86.588	86.610	61.453	66.437
Precision	78.584	76.606	76.591	76.594	76.572	75.783	75.989
Recall	94.605	93.883	93.690	93.767	93.678	70.106	75.943
F-Measure	82.889	80.925	80.905	80.907	80.870	71.387	74.758
Return	1.109	1.072	1.076	1.077	1.047	0.786	0.817
Film Trust Dataset, Grid Size = 7x7							
	ITR	**Cosine**	**Euclidian**	**Manhattan**	**Jaccard**	**PCC**	**TMJ**
User Coverage	100.000	95.182	95.182	95.182	95.182	69.236	72.943
Item Coverage	91.269	86.507	86.490	86.542	86.463	61.399	64.875
Precision	79.731	76.674	76.691	76.640	76.719	75.838	76.068
Recall	95.514	92.913	93.235	92.928	93.384	69.463	74.601

F-Measure	84.107	80.008	80.040	79.950	80.092	78.450	78.854
Return	0.859	0.825	0.816	0.820	0.830	0.609	0.633

Based on our analysis, it can be concluded that any of the four measures including Cosine, Euclidean, Manhattan, and Jaccard can be effectively utilized when selecting a similarity measure for determining the start state. Additionally, it is noteworthy that measures such as PCC and TMJ performed poorly specifically with the Film Trust dataset, indicating their unsuitability for certain contexts.

Conclusion and Future Work:

In this study, we applied seven different similarity measures (ITR, Cosine, Jaccard, Euclidian, Manhattan, PCC and TMJ) to determine start state for two different-sized squared grid RL environments. The objective was to determine a start state that can increase recommendation performance. We observed that ITR has a minimum state count of state -1 for ML-100K dataset as compared to all other start state determination measures, for both grid environments, but on the other hand, ITR is unable to select more than half of states as start state. Although currently, it has no impact on user coverage and item coverage it seems that it can result in restricted movement of RL agents and can reduce item diversity and user personalization if tested. For FilmTrust dataset, ITR measure has a state count of -1 state as zero for both grid environments, whereas all other measures have a state count of -1 greater than zero. This implies that ITR achieved user coverage of 100% for FilmTrust dataset. As sparsity of FilmTrust dataset is greater than ML-100K dataset and user coverage of ITR for FilmTrust dataset is 100%, we can conclude that ITR has the ability to work well with sparse datasets. On both datasets, worst performance is given by the TMJ measure, giving minimum user and item coverage as compared to other measures. After TMJ measure, PCC measure gives the worst performance. In terms of balanced state selection distribution Euclidian performed best for both datasets (figure 4, 5,6, 7).

In this work, the start state is determined on a squared grid environment. But how to determine this start state in a different environment like a tree environment is still a matter of exploration. Seven different similarity measures are used to determine the start state but in recent years' researchers have proposed many novel similarity measures that can also be tested. Currently used similarity measures use only ratings information of the user to determine start state, demographic features can also be incorporated to determine a more precise start state. Future work can be done to get a deeper dive into the performance of these similarity measures, by taking into consideration more robust evaluation measures like MAE, RMSE, novelty, and personalization. Demographic or context information can also be used for careful selection of a start state. In the future we also intend to determine the internal characteristics of highly selected states/biclusters.

Acknowledgment:

- First two authors contributed equally.
- Third and Fourth author helped in problem formulation and modelling.

References:

[1] M. Ahmadian, S. Ahmadian, and M. Ahmadi, "RDERL: Reliable deep ensemble reinforcement learning-based recommender system," Knowledge-Based Syst., vol. 263, p. 110289, Mar. 2023, doi: 10.1016/J.KNOSYS.2023.110289.

[2] F. Tahmasebi, M. Meghdadi, S. Ahmadian, and K. Valiallahi, "A hybrid recommendation system based on profile expansion technique to alleviate cold start problem," Multimed. Tools Appl., vol. 80, no. 2, pp. 2339–2354, Jan. 2021, doi: 10.1007/S11042-020-09768-8/METRICS.

[3] K. Sivamayil, E. Rajasekar, B. Aljafari, S. Nikolovski, S. Vairavasundaram, and I. Vairavasundaram, "A Systematic Study on Reinforcement Learning Based Applications,"

Energies 2023, Vol. 16, Page 1512, vol. 16, no. 3, p. 1512, Feb. 2023, doi: 10.3390/EN16031512.
[4] A. Iftikhar, M. A. Ghazanfar, M. Ayub, S. Ali Alahmari, N. Qazi, and J. Wall, "A reinforcement learning recommender system using bi-clustering and Markov Decision Process," Expert Syst. Appl., vol. 237, p. 121541, Mar. 2024, doi: 10.1016/J.ESWA.2023.121541.
[5] Y. Ge et al., "Toward pareto efficient fairness-utility trade-off in recommendation through reinforcement learning," WSDM 2022 - Proc. 15th ACM Int. Conf. Web Search Data Min., pp. 316–324, Feb. 2022, doi: 10.1145/3488560.3498487.
[6] F. Fkih, "Similarity measures for Collaborative Filtering-based Recommender Systems: Review and experimental comparison," J. King Saud Univ. - Comput. Inf. Sci., vol. 34, no. 9, pp. 7645–7669, Oct. 2022, doi: 10.1016/J.JKSUCI.2021.09.014.
[7] S. Benkessirat, N. Boustia, and R. Nachida, "A New Collaborative Filtering Approach Based on Game Theory for Recommendation Systems," J. Web Eng., vol. 20, no. 2, pp. 303–326, Mar. 2021, doi: 10.13052/JWE1540-9589.2024.
[8] Q. Zhang, J. Lu, and Y. Jin, "Artificial intelligence in recommender systems," Complex Intell. Syst., vol. 7, no. 1, pp. 439–457, Feb. 2021, doi: 10.1007/S40747-020-00212-W/FIGURES/1.
[9] M. M. Afsar, "Personalized Recommendation Using Reinforcement Learning," 2022, doi: 10.11575/PRISM/39785.
[10] R. . Sargar, "Recommender System using Reinforcement Learning," Arizona State Univ., 2020.
[11] J. Leskovec, A. Rajaraman, and J. D. Ullman, "Mining of Massive Datasets," Min. Massive Datasets, Jan. 2020, doi: 10.1017/9781108684163.
[12] E. Saravana Kumar, K. Vengatesan, R. P. Singh, and C. Rajan, "Biclustering of gene expression data using biclustering iterative signature algorithm and biclustering coherent column," Int. J. Biomed. Eng. Technol., vol. 26, no. 3–4, pp. 341–352, 2018, doi: 10.1504/IJBET.2018.089968.
[13] A. Iftikhar, M. A. Ghazanfar, M. Ayub, Z. Mehmood, and M. Maqsood, "An Improved Product Recommendation Method for Collaborative Filtering," IEEE Access, vol. 8, pp. 123841–123857, 2020, doi: 10.1109/ACCESS.2020.3005953.
[14] S. Choi, H. Ha, U. Hwang, C. Kim, J.-W. Ha, and S. Yoon, "Reinforcement Learning based Recommender System using Biclustering Technique," Jan. 2018, Accessed: May 19, 2024. [Online]. Available: https://arxiv.org/abs/1801.05532v1
[15] F. M. Harper and J. A. Konstan, "The MovieLens Datasets," ACM Trans. Interact. Intell. Syst., vol. 5, no. 4, Dec. 2015, doi: 10.1145/2827872.
[16] G. Guo, J. Zhang, and N. Yorke-Smith, "A Novel Bayesian Similarity Measure for Recommender Systems".
[17] "librec.net." Accessed: May 19, 2024. [Online]. Available: http://ww12.librec.net/datasets.html?usid=15&utid=28471851619
[18] V. A. Padilha and R. J. G. B. Campello, "A systematic comparative evaluation of biclustering techniques," BMC Bioinformatics, vol. 18, no. 1, pp. 1–25, Jan. 2017, doi: 10.1186/S12859-017-1487-1/FIGURES/15.
[19] H. Al-Bashiri, M. A. Abdulgabber, A. Romli, and H. Kahtan, "An improved memory-based collaborative filtering method based on the TOPSIS technique," PLoS One, vol. 13, no. 10, p. e0204434, Oct. 2018, doi: 10.1371/JOURNAL.PONE.0204434.
[20] H. Liu, Z. Hu, A. Mian, H. Tian, and X. Zhu, "A new user similarity model to improve the accuracy of collaborative filtering," Knowledge-Based Syst., vol. 56, pp. 156–166, Jan. 2014, doi: 10.1016/J.KNOSYS.2013.11.006.
[21] S. B. Sun et al., "Integrating Triangle and Jaccard similarities for recommendation," PLoS One, vol. 12, no. 8, p. e0183570, Aug. 2017, doi: 10.1371/JOURNAL.PONE.0183570.
[22] "Sutton & Barto Book: Reinforcement Learning: An Introduction." Accessed: May 19, 2024. [Online]. Available: http://incompleteideas.net/book/the-book-2nd.html