Journal of Iranian Association of Electrical and Electronics Engineers
Vol. 22 No. 1, 2025
DOI: 10.61186/jiaeee.22.1.151

Optimal Modeling of Interactions between Users and Items in Recommender Systems Using an Improved Deep Reinforcement Learning Method

Saber Alinejad[1] , Behrooz Koohestani[2] , Mohammad Reza Feizi Derakhshi[3]

[1] MSc student, Faculty of Electrical and Computer Engineering, University of Tabriz, Tabriz, Iran
alinezhadsaber@gmail.com
[2] Associate Professor, Faculty of Electrical and Computer Engineering, University of Tabriz, Tabriz, Iran
b.koohestani@tabrizu.ac.ir
[3] Professor, Faculty of Electrical and Computer Engineering, University of Tabriz, Tabriz, Iran
mfeizi@tabrizu.ac.ir

Abstract :

Recommender systems are one of the most important topics in academia and industry. With the increase in the volume of information and data, it has become confusing and sometimes impossible for users to access the required services without using recommender systems. So far, various techniques have been proposed for this purpose such as collaborative filtering, matrix factorization, logistic regression, neural networks, etc. However, most of these methods suffer from two limitations: (1) considering the recommendation as a static procedure and ignoring the dynamic interactive nature between users and the recommender systems; (2) focusing on the immediate feedback of recommended items and neglecting the long-term rewards. In this research, the modeling of interactions between users and items is done using an improved deep reinforcement learning method which can consider both the dynamic adaptation and long term rewards. The results of the experiments show that the proposed algorithm performs better than other methods.

Keywords: Recommender systems, Deep reinforcement learning, Artificial intelligence, User item interactions.

Article Type: Research

Received: 20. 08. 2023

Revised: 18. 02. 2024

Accepted: 01. 08. 2024

Corresponding author: Behrooz Koohestani

Corresponding author's address: Faculty of Electrical and Computer Engineering, University of Tabriz, Tabriz, Iran

[Downloaded from jiaeee.com on 2026-02-15]

[DOI: 10.61186/jiaeee.22.1.121]

Copyright © 2025 The Authors. Published by Iranian Association of Electrical and Electronics Engineers.
This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International license (https://creativecommons.org/licenses/by-nc/4.0/). Non-commercial uses of the work are permitted, provided the original work is properly cited.

1. Motivation of the work

With the rise of online services like shopping, news, and social networks, it has become very convenient to buy goods, books, videos, and news through the Internet or mobile devices. Despite this convenience, the large number of items available also presents a considerable challenge for users in finding items that interest them. Recommendation is a commonly utilized remedy and different sets of methods have been suggested in this area, including content-based collaborative filtering [1], matrix factorization based methods [2] and deep learning models [3]. The studies mentioned have two significant limitations in common [4]. Firstly, the recommendation procedure is generally seen as a static process, meaning they assume that the user's underlying preference remains unchanged. Secondly, the studies mentioned earlier are trained by maximizing the instant rewards of suggestions, focusing solely on whether the suggested items are clicked or used, and ignoring the long-term impact the items can have. This leads to a considerable decrease in the performance of recommender systems. The motivation behind the current work was to find a solution to the problem mentioned earlier by presenting an improved deep reinforcement learning method.

2. Contributions

The paper's key contributions can be outlined as follows: We present a recommendation framework based on deep reinforcement learning. The proposed framework differs from traditional approaches by utilizing an Actor-Critic architecture and viewing recommendation as a sequential decision-making process that considers both immediate and long-term rewards. Also, the interactions between users and items can be explicitly modeled. One of the most important advantages of the proposed system is its more effective network updating. In fact, the proposed system includes a public network and several separate actor and critic networks, each of which is executed in different threads and obtains its experience from the interaction between the user and the item, and after improvement, copies its weight to the public network. In this step, a set of transfer steps are extracted from the buffer so that the algorithm's update mechanism starts and finally desired items to the user can be recommended.

3. rocedures

The current study's procedure consists of the following steps: 1) Data preparation, 2) Generating a model for user-item interactions, 3) Generating a model for predicting the user's score to the item, 4) Designing actor network, 5) Designing critic network, 6) Designing state representation module, 7) Designing experience replay module. Fig. 1 illustrates the structure of the proposed recommendation system.

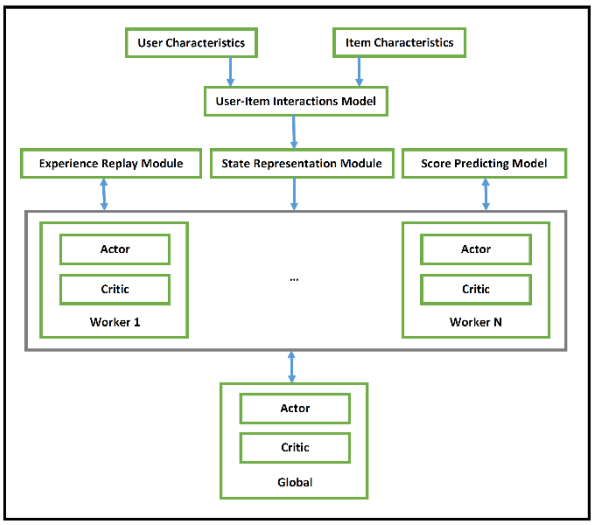


Fig. 1 Flowchart of the proposed recommendation system.

4. Findings

The experiments were conducted using the following real-world publicly available datasets: MovieLens (100k), MovieLens (1M) and Jester. Precision@k and NDCG@k were utilized as metrics to evaluate the performance of our proposed approach. We compared the proposed recommendation system against four of the most widely used methods, namely Popularity, PMF, SVD++ and DRR [5]. The results of experiments demonstrate that the proposed recommendation system outperforms the competitors in terms of the quality of the solutions obtained. This confirms that our approach is very effective and can replace previously used strategies.

5. Conclusion

In this paper, a recommendation framework based on deep reinforcement learning has been proposed. So far, several recommender systems have been developed each of which uses a different strategy. In contrast to the previously introduced recommender systems, our proposed method views recommendations as a series of decisions and utilizes an Actor-Critic learning approach that considers both short-term and long-term rewards. A state representation module is also included in the proposed method, along with instantiation structures that can explicitly capture the interactions between users and items. The proposed method has been shown to outperform four well-known and widely-used competitors in extensive experiments conducted on three real-world datasets that acknowledges its effectiveness and usefulness.

[Downloaded from jiaeee.com on 2026-02-15]

[DOI: 10.61186/jiaeee.22.1.121]

نشریه مهندسی برق و الکترونیک ایران

نشریه مهندسی برق و الکترونیک ایران، سال بیست و دوم، شماره اول، بهار ۱۴۰۴، صفحات ۱۵۱-۱۶۰

DOI: 10.61186/jiaeee.22.1.153

مدل‌سازی بهینه تعاملات بین کاربران و آیتم‌ها در سیستم‌های توصیه‌گر با استفاده از یک روش یادگیری تقویتی عمیق بهبود یافته

صابر علی نژاد[۱] ، بهروز کوهستانی[۲] ، محمدرضا فیضی درخشی[۳]

۱- دانش آموخته کارشناسی ارشد- دانشکده مهندسی برق و کامپیوتر- دانشگاه تبریز- تبریز- ایران

alinezhadsaber@gmail.com

۲- دانشیار- دانشکده مهندسی برق و کامپیوتر- دانشگاه تبریز- تبریز- ایران

b.koohestani@tabrizu.ac.ir

۳- استاد- دانشکده مهندسی برق و کامپیوتر- دانشگاه تبریز- تبریز- ایران

mfeizi@tabrizu.ac.ir

چکیده: سیستم‌های توصیه‌گر یکی از مباحث بسیار مهم هم در زمینه آکادمیک و هم در زمینه صنعت است. علت اهمیت سیستم‌های توصیه‌گر ناشی از این حقیقت است که با افزایش حجم اطلاعات و گسترش داده‌ها، دسترسی کاربران به خدمات و سرویس‌های مورد نیاز خودشان در میان انبوه اطلاعات، بدون استفاده از سیستم‌های توصیه‌گر به یک امر سر در گم کننده و گاها غیر ممکن تبدیل می‌شود. تاکنون روش‌های مختلفی از جمله فیلترینگ مشارکتی، فاکتورگیری ماتریسی، رگرسیون لجستیک و شبکه‌های عصبی در این زمینه ارائه شده‌اند که اکثر این روش‌ها دارای محدودیت‌های خاصی هستند. اولین محدودیت این سیستم‌ها ثابت بودن و عدم توجه به تعاملات کاربر با سیستم در گذر زمان و دومین محدودیت در این سیستم‌ها تمرکز کردن بر روی پاداش‌های آنی و عدم توجه به پاداش‌های بلند مدت است. در این تحقیق، مدل‌سازی تعاملات بین کاربران و آیتم‌ها با استفاده از یک الگوریتم یادگیری تقویتی عمیق بهبود یافته صورت می‌گیرد تا سیستم توصیه‌گر تصمیم‌های خود را بصورت یک فرآیند پویا با گذر زمان بهبود داده و علاوه بر امتیاز آنی حاصل از تصمیم‌های اخذ شده، تاثیرات آن تصمیم در بدست آوردن پاداش‌های بلند مدت را نیز در نظر بگیرد. نتایج حاصل از آزمایش‌ها نشان می‌دهد که الگوریتم پیشنهادی عملکرد بهتری نسبت به سایر روش‌ها داشته است.

کلمات کلیدی: سیستم های توصیه‌گر، یادگیری تقویتی عمیق، هوش مصنوعی، تعامل‌های کاربر و آیتم‌ها

نوع مقاله: پژوهشی

دریافت: ۱۴۰۲/۰۵/۲۹

بازنگری: ۱۴۰۲/۱۲/۲۹

پذیرش: ۱۴۰۳/۰۵/۱۱

نام نویسنده‌ی مسئول: دکتر بهروز کوهستانی

نشانی نویسنده‌ی مسئول: ایران - تبریز - دانشگاه تبریز – دانشکده مهندسی برق و کامپیوتر

[Downloaded from jiaeee.com on 2026-02-15]

[DOI: 10.61186/jiaeee.22.1.121]

حق انتشار این مستند، متعلق به نویسندگان آن است. ۱۴۰۴ ©. ناشر این مقاله، انجمن مهندسین برق و الکترونیک ایران است.

این مقاله تحت گواهی زیر منتشرشده و هر نوع استفاده غیرتجاری از آن مشروط بر استناد صحیح به مقاله و با رعایت شرایط مندرج در آدرس زیر مجاز است.

Creative Commons Attribution-NonCommercial 4.0 International license

(https://creativecommons.org/licenses/by-nc/4.0/)

۱- مقدمه

با افزایش سرویس‌های برخط از قبیل فروشگاه‌های اینترنتی، وبسایت‌های خبری، شبکه‌های اجتماعی و هزاران خدمات اینترنتی دیگر، دسترسی به منابع مختلف از جمله محصولات، کتاب‌ها، اخبار و سایر موارد خیلی راحت و آسان شده است. در کنار این راحتی بوجود آمده، با توجه به تعداد زیاد ارائه دهنده خدمات و دسترسی راحت کاربران به تمامی منابع، انتخاب موارد منطبق بر علایق و شرایط به یک چالش مهم تبدیل شده است به طوری که کاربران با صرف مدت زمان زیادی جهت دستیابی به آیتم مورد نظر خود، نه تنها موفق به پیدا کردن آن نمی‌شوند بلکه با افزایش سردرگمی سایت مورد نظر را ترک کرده و یک دید منفی نسبت به شیوه ارائه محتوا در سایت مورد نظر پیدا می‌کنند. حال، کسب کار مورد نظر با اینکه حجم عظیمی از اطلاعات جامعی را در اختیار کاربران قرار داده، موفق به فروش و یا افزایش بازدهی وب سایت خود نمی‌شود. سیستم‌های توصیه‌گر یک راه حل گسترده برای حل این مشکل می‌باشند که در این سیستم‌ها تلاش بر این است تا با حدس زدن شیوه تفکر کاربر، مناسب‌ترین و نزدیک‌ترین کالا به سلیقه او را شناسایی کرده و به او پیشنهاد کنیم. تاکنون سیستم‌های توصیه‌گر متفاوتی با روش‌های گوناگونی پیاده‌سازی شده و در زمینه‌های مختلف به کار گرفته شده‌اند [۱]. از جمله این روش‌ها می‌توان به فیلترینگ مبتنی بر محتوا[۱] [۲-۴]، روش‌های مبتنی بر فاکتورگیری ماتریسی[۲] [۵-۹] و مدل‌های یادگیری عمیق[۳] [۱۰-۱۲] اشاره کرد. اساس کار فیلترینگ مبتنی بر محتوا بر این فرضیه استوار است که اشخاصی که در گذشته سلیقه مشابهی داشته‌اند در آینده هم سلیقه مشابهی خواهند داشت و آیتم‌ها و سرویس‌های یکسانی را می‌پسندند. یکی از مشکلات اساسی در فیلترینگ مبتنی بر محتوا ، بار پردازشی زیاد برای توصیه است زیرا در محیط اجرای آن، تعداد کاربران و آیتم‌ها بسیار زیاد است [۱۳]. همچنین، پراکندگی اطلاعات در این سیستم‌ها نیز یک مشکل دیگر به حساب می‌آید زیرا در مقابل حجم زیادی از آیتم‌ها کاربر فقط با تعداد اندکی از آن ها تعامل دارد. در واقع فیلترینگ مبتنی بر محتوا عملکرد ضعیفی را در مقابل پراکندگی داده‌ها از خود نشان می‌دهد. همانطور که بیان شد، استفاده از این سیستم‌ها باعث کاهش سردرگمی کاربران و همچنین دستیابی به آیتم مورد نظر در مدت زمان کمتر شده است که این امر نه تنها سبب رضایت بیشتر کاربران بلکه باعث افزایش میزان فروش محصولات در فروشگاه‌های اینترنتی می‌شود. این سیستم‌ها، داده‌های اضافی یا ناخواسته را از پایگاه داده حذف می‌کنند و نویز را در سطح معنایی کاهش می‌دهند. معمولا پیشنهادات به انواع مختلف فرآیندهای تصمیم‌گیری اشاره می‌کنند، از جمله اینکه کدام محصول را بخرم، به کدام موسیقی گوش کنم، و یا کدام خبر را بخوانم. با این حال، با اعمال این سیستم‌ها بر روی کسب و کارهای مختلف و با گذر زمان و بهبود تجربه کاربری مشکلاتی در استفاده از این سیستم‌ها گزارش شده است. یکی از مشکلات عمده الگوریتم‌های مختلف سیستم‌های توصیه‌گر[۴] نگاه کردن به فرآیند توصیه آیتم مورد نظر به کاربر به عنوان یک پارامتر ثابت است، در صورتی که ترجیح‌های کاربر با گذر زمان و با توجه به رویدادها و نیازهای روزانه تغییر پیدا می‌کند [۱۳]. دومین مشکل توصیه گرها سعی در بدست آوردن بیشترین امتیاز در لحظه حال و عدم توجه به مجموع امتیاز بدست آمده در توصیه‌های پیاپی است، در صورتی که بدست آوردن امتیاز در بلند مدت از اهمیت زیادی برخوردار است [۱۳]. به طور مثال، در یک سایت خبری برای یک کاربر برنامه نویس ممکن است یک خبر در مورد زبان برنامه نویسی و یک خبر در مورد خودرو دارای احتمال مشاهده یکسان باشند ولی نکته حیاتی در اینجا این است که کاربر با مشاهده کدام اخبار تمایل دارد تا خبرهای بیشتری در این زمینه را مطالعه کند تا در نهایت ماندگاری کاربر در سامانه به بیشرین زمان و یا به بالاترین امتیاز ممکن برسد. اخیرا روش‌های مبتنی بر یادگیری تقویتی عمیق[۵] [۱۴] پتانسیل خوبی را در حل مسائل مختلف [۱۵-۱۷] از جمله در زمینه سیستم‌های توصیه‌گر [۱۸] از خود نشان داده‌اند. در این تحقیق، یک روش یادگیری تقویتی عمیق بهبود یافته برای مدل‌سازی بهینه تعاملات بین کاربران و آیتم‌ها ارائه می‌شود. الگوریتم پیشنهادی می‌تواند تصمیم‌های خود را در یک فرآیند پویا با گذر زمان بهبود داده و علاوه بر امتیاز آنی، تاثیر تصمیم‌های خود در بدست آوردن پاداش‌های بلندمدت را نیز در نظر بگیرد.

در ادامه، این مقاله شامل بخش‌های زیر است:

در بخش دوم، سیستم‌های توصیه گر به همراه انواع مدل‌های آن مورد بررسی قرار می‌گیرند. در بخش سوم، الگوریتم پیشنهادی به همراه اجزای تشکیل دهنده آن ارائه می‌شود. در بخش چهارم، الگوریتم پیشنهادی مورد ارزیابی قرار گرفته و نتایج حاصل از عملکرد آن با نتایج چهار الگوریتم پرکاربرد مقایسه می‌شود. در بخش پنجم، دلایل عملکرد بهتر الگوریتم پیشنهادی ذکر شده و در نهایت در بخش ششم به نتیجه‌گیری از این تحقیق پرداخته می‌شود.

۲- سیستم‌های توصیه‌گر

به طور کلی، سیستم‌های توصیه‌گر را می‌توان به دو کلاس اصلی شامل سیستم‌های توصیه‌گر غیر مبتنی بر یادگیری تقویتی و سیستم‌های توصیه‌گر مبتنی بر یادگیری تقویتی تقسیم کرد. هر یک از این دو کلاس نیز به زیر کلاس‌هایی تقسیم می‌شوند. جدول ۱، انواع مختلف سیستم‌های توصیه‌گر را به همراه طبقه‌بندی و محدودیت‌های هر کدام نشان می‌دهد. از رده سیستم‌های توصیه‌گر مبتنی بر یادگیری تقویتی، ابتدا سیستم‌های توصیه‌گر مبتنی بر مدل[۶] ارائه شدند که به دلیل پیچیدگی زمانی قابلیت پیاده‌سازی بر روی سیستم‌هایی که تعداد آیتم‌ها در آن‌ها زیاد است را نداشتند. سپس، سیستم‌های توصیه گر بدون مدل[۷] ارائه شدند که عمدتا به دو نوع مبتنی بر ارزش[۸] و مبتنی بر سیاست[۹] تقسیم می‌شوند. در نوع مبتنی بر ارزش، با گرفتن وضعیت فعلی، مقدار کیو (Q-value) برای تمامی عملیات ممکن محاسبه شده و سپس

جدول (۱): انواع مختلف سیستم‌های توصیه‌گر

[Downloaded from jiaeee.com on 2026-02-15]

[DOI: 10.61186/jiaeee.22.1.121]

محدودیت	مرجع	روش		کلاس سیستم توصیه‌گر
عمدتا از دو مشکل اساسی رنج می‌برند: ۱- روند توصیه را یک فرآیند ثابت در نظر می‌گیرند. ۲- تمرکز عمده آنها بر به حداکثر رساندن پاداش‌های آنی است.	[۲-۴]	فیلترینگ مبتنی بر محتوا[1]		غیر مبتنی بر یادگیری تقویتی
	[۵-۹]	روش های مبتنی بر فاکتورگیری ماتریسی[2]		
	[۱۰-۱۲]	مدل‌های یادگیری عمیق[3]		
مشکل اصلی سیستم‌های توصیه‌گر مبتنی بر مدل پیچیدگی زمانی زیاد آنها است.	[۱۹,۲۰]	مبتنی بر مدل[6]	مبتنی بر ارزش[8]	مبتنی بر یادگیری تقویتی
	[۲۱,۲۲]		مبتنی بر سیاست[9]	
	[۲۳,۲۴]		ترکیبی[10]	
نیاز به ارزیابی Q-value همه عملیات تحت یک حالت خاص دارند که این علت اصلی ناکارآمدی آنها است.	[۲۵,۲۶]	بدون مدل[7]	مبتنی بر ارزش	
یکی از محدودیت‌های رایج این رویکردها این است که آنها نمایش/ توصیف حالت را با دقت یاد نمی‌گیرند.	[۲۷,۲۸]		مبتنی بر سیاست	
	[۲۹,۳۰]		ترکیبی	

عمل با بیشترین مقدار کیو برای اجرا انتخاب می‌شود. در نوع مبتنی بر سیاست از یک بردار برای نگاشت یک حالت به بهترین عمل ممکن استفاده می‌شود. در واقع اساس کار یادگیری تقویتی بر پایه فرآیند تصمیم‌گیری مارکوف [۳۱] است که به صورت زیر تعریف می‌شود:

(S, A, P, R, y)

- S: فضای حالت
- A: فضای عمل
- P: تابع احتمال انتقال حالت
- R: تابع پاداش
- Y: ضریب کاهش[11]

هدف عامل در فرآیند تصمیم مارکوف، پیدا کردن سیاست بهینه‌ای است که بیشترین امتیاز انباشته شده در هر حالت را اخذ کند. ما فرآیند توصیه را بصورت توالی از مسائل تصمیم‌گیری در نظر می‌گیریم که در آن توصیه‌گر با کاربر ارتباط برقرار می‌کند تا لیستی از آیتم‌ها را در بازه‌های زمانی به کابر پیشنهاد کند، به طوری که بیشترین پاداش انباشته شده را بدست بیاورد. در حالت کلی سیستم توصیه‌گر بصورت زیر بر اساس فرآیند تصمیم‌گیری مارکوف مدل می‌شود:

- States: حالت S نشان دهنده تاریخچه ارتباط مثبت کاربر با توصیه‌گر است.
- Actions: عمل A یک بردار از ضرب داخلی عمل و آیتم‌های تعبیه شده است که بیشترین آن برای توصیه انتخاب می‌شود.
- Transitions: تابع انتقال حالت، زمانی عمل می‌کند که در واقع بازخوردی از کاربر دریافت می‌شود و بر اساس آن حالت S به روز رسانی می‌شود.
- Reward: پاداشی است که توصیه‌گر با انجام عمل A در حالت S بدست می‌آورد.
- Discount rate: فاکتوری برای اندازه‌گیری پاداش بلند مدت است که در آن، صفر به معنای اولویت دادن به پاداش آنی و در نظر نگرفتن پاداش بلند مدت و یک به معنای اهمیت یکسان پاداش آنی و بلند مدت برای توصیه‌گر است.

۳- الگوریتم پیشنهادی

الگوریتم ارائه شده در این مقاله یک الگوریتم بهبود یافته براساس یادگیری تقویتی عمیق است که بخش‌های مختلف آن در زیر توضیح داده شده است.

۳-۱- آماده‌سازی داده‌ها

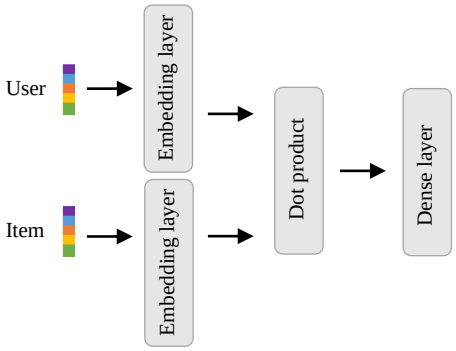
قبل از شروع فرآیند آموزش با توجه با داده‌های موجود در دیتاست، یک جاسازی[12] از داده‌ها با توجه به تاریخچه تعاملات کاربر با آیتم‌ها بدست می‌آید و در فرایند آموزش مورد استفاده قرار می‌گیرد. در الگوریتم پیشنهادی، داده‌ها به سه قسمت تقسیم بندی شده‌اند: قسمت اول برای آموزش، قسمت دوم برای تایید و قسمت سوم داده‌ها برای تست الگوریتم پیشنهادی مورد استفاده قرار می‌گیرد. نحوه تقسیم‌بندی داده‌ها به این شکل است که ابتدا داده‌های مربوط به کاربران براساس تاریخ ارتباط با آیتم‌ها مرتب شده، سپس تقسیم بندی داده صورت می‌گیرد. دلیل این کار تغییر اولویت کاربران در برخورد با آیتم‌ها با گذر زمان است. همچنین، برای پردازش سن کاربران از روش کیف کلمات استفاده شده که بازه‌های سنی مختلف را در گروه‌های متفاوت قرار می‌دهد. ویژگی جنسیت و شغل کاربران با استفاده از بردارسازی شمارشی به اندیس عددی نگاشت شده است و در نهایت عناوین که در اینجا نام فیلم و متن مربوط به جوک‌ها است و همچنین ژانر مربوط به فیلم‌ها (یک فیلم ممکن است چندین ژانر داشته باشد) با استفاده از روش‌های تعبیه کلمات به بردارهای عددی تبدیل شده‌اند.

[Downloaded from jiaeee.com on 2026-02-15]

[DOI: 10.61186/jiaeee.22.1.121]

۳-۲- ایجاد مدل ارتباط کاربر با آیتم

در این مرحله، داده‌های پیش پردازش شده در قسمت قبلی جهت آموزش یک مدل شبکه عصبی مورد استفاده قرار می‌گیرند تا یک جاسازی از داده‌های موجود را به دست بیاوریم.

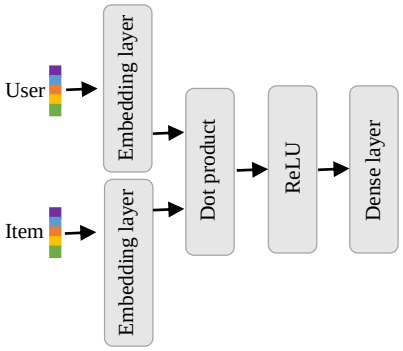


شکل (۱): مدل ارتباط کاربر با آیتم

با توجه به شکل (۱)، مدل ارتباط کاربر با آیتم پس از دریافت ویژگی‌های مربوط به کاربر و آیتم، بردار جاسازی آنها را بدست آورده و پس از ضرب داخلی آنها، یک بردار واحد بدست می‌آورد. داده‌های این بردار نهایتا وارد یک لایه متراکم[۱۳] می‌شوند تا با توجه به امتیاز کاربران به آیتم‌ها، شبکه مورد نظر مورد آموزش قرار گیرد. بعد از آنکه شبکه عصبی ایجاد شده مورد آموزش قرار گرفت، بردارهای مربوط به لایه جاسازی شبکه مورد نظر به عنوان پارامتر ورودی در الگوریتم پیشنهادی مورد استفاده قرار می‌گیرند. به عبارت دیگر، به جای استفاده مستقیم از بردارهای ویژگی کاربران و آیتم‌ها از بردارهای جاسازی آن‌ها استفاده می‌شود.

۳-۳- ایجاد مدل پیش‌بینی امتیاز کاربر به آیتم

همانطور که قبلا هم اشاره شد، امتیازهای موجود در دیتاست به ازای تعامل بین کاربر و آیتم ثبت گردیده است. بنابراین، در میان انبوهی از آیتم ها، کاربران فقط تعداد اندکی از آنها را مشاهده کرده و برای آنها امتیاز ثبت کرده‌اند. پس نیاز داریم تا یک مدل پیش بینی کننده امتیاز نیز پیاده‌سازی کنیم تا در پروسه یادگیری الگوریتم پیشنهادی، در صورتی که به ازای آیتم توصیه شده به کاربر امتیازی در دیتاست موجود نباشد از مدل پیش بینی کننده امتیاز استفاده کنیم. با توجه به شکل (۲)، مدل پیش بینی امتیاز پس از دریافت ویژگی‌های مربوط به کاربر و آیتم، بردار جاسازی آنها را بدست آورده و پس از ضرب داخلی آنها یک بردار واحد بدست می‌آورد که دادهای این بردار با عبور از سه لایه متراکم با تابع فعال سازی ReLU نهایتا وارد یک لایه متراکم بدون تابع فعال‌سازی می‌شوند تا با توجه به امتیاز کاربران به آیتم‌ها شبکه مورد نظر مورد آموزش قرار گیرد، این شبکه امتیاز کاربر به آیتم‌هایی که قبلا مشاهده نکرده است را پیش‌بینی می‌کند.



شکل (۲): مدل پیش بینی امتیاز

۳-۴- ایجاد شبکه actor

شبکه actor که شبکه سیاست هم نامیده می‌شود برای تولید action بر پایه حالت S به کار می‌رود. ورودی این شبکه شامل n تا از آخرین ارتباط‌های مثبت کاربر با آیتم‌های تعبیه شده است. سپس، آیتم‌های تعبیه شده وارد بخش نمایش حالت می‌شوند تا یک حالت برای کاربر بدست آید. برای نمونه، در زمان t وضعیت می‌تواند به صورت فرمول (۱) تعریف شود:

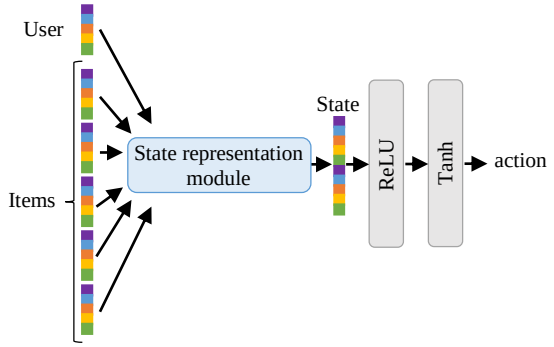
$$S_t = f(H_t) \qquad (۱)$$

در واقع f(0) نشان دهنده state است و $H_t=\{i_1,\ldots,i_n\}$ شامل جاسازی تاریخچه تعاملات مثبت کاربر با آیتم‌ها است. در صورتی که کاربر به آیتم تولید شده توسط توصیه‌گر پاسخ مثبت بدهد، state کاربر بصورت فرمول (۲) به روزرسانی می‌شود.

$$S_{t+1} = f(H_{t+1}), H_{t+1} = \{i_{2,\ldots,}i_t\} \qquad (۲)$$

در این شبکه، state با استفاده از یک لایه ReLU و یک لایه Tanh به action مورد نظر تبدیل می‌شود که شامل آرایه‌ای از امتیاز آیتم‌ها است و در نهایت خروجی شبکه شامل یک بردار است که action مورد نیاز را مشخص می‌کند و بر اساس این action آیتمی که بیشترین امتیاز را داشته باشد به کاربر توصیه می‌شود. یکی از مشکلاتی که در اینجا وجود دارد عدم اکتشاف در محیط اقدامات است که برای جلوگیری از آن، در فضای action بدست آمده با یک احتمال خیلی کم اختلال ایجاد می‌کنیم. با توجه به شکل (۳) برای بدست آوردن بردار action، وضعیت حاصل از ماژول نمایش حالت که در ادامه توضیح داده خواهد شد، از سه لایه متراکم عبور می‌کند. تابع فعال سازی مربوط به دو لایه اول ReLU بوده و تابع فعال سازی مربوط به لایه نهایی Tanh است.

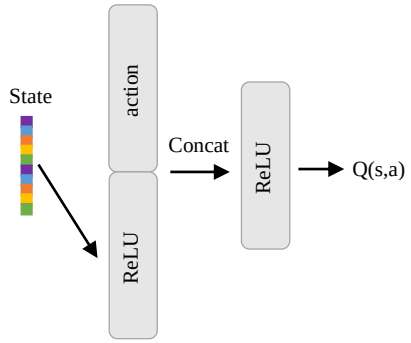
[Downloaded from jiaeee.com on 2026-02-15]
[DOI: 10.61186/jiaeee.22.1.121]



شکل (۳): شبکه actor

۳-۵- ایجاد شبکه critic

این قسمت شامل یک شبکه عمیق Q است که برای تخمین مقدار واقعی تابع state-action مورد استفاده قرار می‌گیرد و به آن تابع Q-value می‌گویند. می‌توان گفت، تابع Q-value عمل تولید شده به وسیله شبکه سیاست را مورد ارزیابی قرار می‌دهد و بر اساس مقدار این تابع پارامترهای شبکه سیاست به روزرسانی می‌شوند تا کارایی آن بهبود پیدا کند. علاوه بر این، خود شبکه critic نیز بر اساس رویکرد یادگیری temporal-difference به روز رسانی می‌شود. یکی از مشکلات اساسی در این مرحله همبستگی نمونه‌های آموزشی است که از همگرایی مدل جلوگیری می‌کند. برای رفع این مشکل از روش تکرار تجربه استفاده می‌شود، به این صورت که تجربیات مدل در یک بافر ذخیره شده و مدل هر بار که نیاز باشد به روزرسانی شود، تعدادی از نمونه ها از بافر استخراج شده و برای به روز رسانی مدل مورد استفاده قرار می‌گیرند. همچنین، به دلیل اینکه شبکه اهداف غیر ثابتی را دنبال می‌کند از روش شبکه هدف مجزا نیز در اینجا استفاده شده است. با توجه به شکل (۴)، شبکه critic با در کنار هم قرار دادن بردار حاصل از ماژول نمایش حالت و بردار بدست آمده از شبکه actor، یک بردار واحد بدست می‌آورد که این بردار پس از عبور از دو لایه متراکم با تابع فعال سازی ReLU مقدار Q-value را تخمین می‌زند.



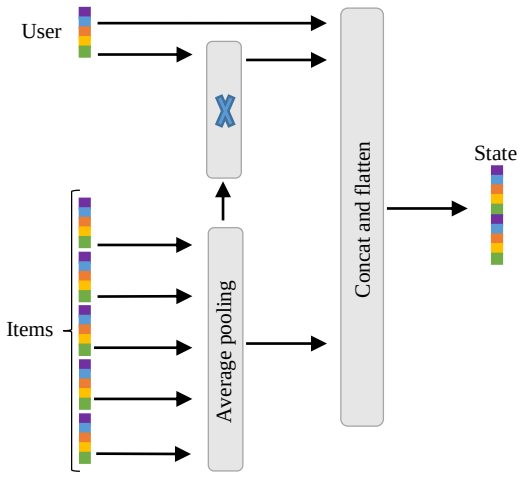
شکل (۴): شبکه critic

۳-۶- ماژول تکرار تجربه

این ماژول، شامل مجموعه‌ای از آرایه‌ها است که به مرور طی فرایند یادگیری در الگوریتم پیشنهادی، وضعیت جاری، عملی که انجام شده، پاداشی که بدست آمده، وضعیت بعدی و همچنین یک متغیر منطقی که مشخص کننده حالت مطلوب است را در خود ذخیره می‌کند. با پر شدن بافر، تجربه های جدید الگوریتم پیشنهادی جایگزین تجربه های قدیمی می‌شوند، در نهایت هنگام به روزرسانی الگوریتم پیشنهادی این داده‌ها به صورت تصادفی و به اندازه batch size از بافر استخراج شده و جهت به روزرسانی پارامترها مورد استفاده قرار می‌گیرند.

۳-۷- ماژول نمایش حالت

ماژول نمایش حالت نقش اساسی را در شبکه actor و critic بازی می‌کند، پس باید ساختار مناسبی برای مدل سازی حالت داشته باشد. این ماژول با توجه به تاریخچه ارتباط کاربر با آیتم‌ها، بردار مربوط به آیتم‌ها را دریافت کرده و با اعمال یک لایه average pooling، میانگینی از این بردارها را بدست می‌آورد و آن را در بردار مربوط به مشخصات کاربر ضرب کرده و نتایج را بصورت هموار در کنار هم قرار می‌دهد.



شکل (۵): ماژول نمایش حالت

۳-۸- شبه کد الگوریتم پیشنهادی

فرایند یادگیری در الگوریتم پیشنهادی به این صورت است که در ابتدا، پارامترهای اولیه از جمله نرخ یادگیری شبکه actor و critic ، نرخ کاهش و اندازه پنجره وضعیت مشخص می‌شوند. سپس، پارامترهای مربوط به شبکه actor و critic عمومی با مقادیر تصادفی مقداردهی می‌شوند و در داخل هر ورکر، در هر مرحله از حلقه تکرار، یک کاربر و n مورد از آیتم‌هایی که کاربر در لحظه t با آنها تعامل مثبت داشته است انتخاب می‌شوند. سپس، موارد انتخاب شده وارد ماژول نمایش حالت می‌شود تا وضعیت جاری کاربر بدست آید. پس از بدست آوردن وضعیت جاری، آن را وارد شبکه actor می‌کنیم تا action مورد نیاز را بدست بیاوریم. بعد از آن، بردار action مربوطه را با بردار سایر آیتم‌هایی که

[Downloaded from jiaeee.com on 2026-02-15]
[DOI: 10.61186/jiaeee.22.1.121]

Algorithm 1: Training algorithm

Input: Actor learning rate lr_a, Critic learning rate lr_c, Discount factor γ, batch size N, state window n.

Initialize the global actor − critic network with random weights θ and ω.
Initialize replay buffer B.
foreach worker in workers **do**:
  Randomaly initialize the actor π_θ and the critic Q_ω with paramiters θ and ω.
  for session = 1, N **do**
    Observe current state $s_t = f(H_t)$ where $H_t = \{i_1,...,i_n\}$.
    Find action $a_t = \pi_\theta(s_t)$ according to the current policy with ε greedy exploration.
    Recommend item i_t according to action a_t.
    Calculate reward $r_t = R(s_t,a_t)$ based on the feedback of the user.
    Observe new state $s_{t+1} = f(H_{t+1})$ where $H_{t+1} = \{i_2,...,i_n,i_t\}$ if r_t is positive else $H_{t+1} = H_t$
    Store transition (s_t,a_t,r_t,s_{t+1}) in B.
    Sample a minibatch of N transitions in B.
    Set $y_i = r_i + \gamma Q_{\acute{\omega}}(s_{i+1},\pi_{\acute{\theta}}(s_{i+1}))$
    Update the critic network by minimizing the loss.
    Update the actor network using the sampled policy gradient.
    Update the global actor − critic network.
  End for
End foreach
Return global actor-critic network

شکل (۶): شبه کد یادگیری الگوریتم پیشنهادی

فعلا به کاربر پیشنهاد داده نشده‌اند ضرب داخلی می‌کنیم تا به ازای هر آیتم مقداری را بدست بیاوریم. سپس، آیتم‌هایی که بیشترین مقادیر را دارند برای پیشنهاد دادن به کاربر انتخاب می‌شوند. حال از بین این آیتم‌ها اگر در تاریخچه موجود در دیتاست، امتیازی برای آیتم مورد نظر ثبت شده باشد، مقدار امتیاز را استخراج می‌کنیم، در غیر اینصورت، مقدار امتیاز آیتم پیشنهاد شده با استفاده از مدل پیش‌بینی کننده محاسبه و بدست می‌آید. حال که تمام امتیاز ها را بدست آوردیم، مجموع امتیاز بدست آمده را به عنوان پاداش عملی که انجام داده‌ایم به همراه بردار حالت، عمل، حالت بعدی و وضعیت عملی که انجام داده‌ایم را جهت استفاده از تجربه‌ها، برای به روزرسانی الگوریتم پیشنهادی در بافر ذخیره می‌کنیم. در صورتی که آیتم‌های پیشنهاد شده به کاربر امتیاز مطلوبی نداشته باشند (امتیاز کاربر به آیتم‌های مورد نظر منفی باشد) با حفظ حالت فعلی کاربر، آیتم‌های جدیدی را توصیه می‌کنیم تا به یک وضعیت مطلوب برسیم و در نهایت با رسیدن به وضعیت مطلوب این روند را برای سایر کاربران نیز تکرار می‌کنیم. پس در نتیجه، در هر مرحله از الگوریتم پیشنهادی، با توجه به امتیاز آیتم‌هایی که به کاربر توصیه می‌شود، وضعیت عمل انجام شده بدست می‌آید. این وضعیت که دارای دو حالت مطلوب و نامطلوب است، در یک متغیر منطقی به نام "انجام شده" [۱۴] در بافر ذخیره می‌شود. در شکل (۷)، شبه کد مراحل به روز رسانی الگوریتم پیشنهادی توضیح داده شده است. لازم به ذکر است که به طور مشخص، نوآوری اصلی الگوریتم پیشنهادی در همین بخش به روز رسانی است.

Algorithm 2: Update steps for the proposed algorithm

Input: batch of N transitions in Buffer, Discount factor γ, global learning rate lr_g.

$Next_actions_{global} \leftarrow global_actor(states_{next})$
$Q_{value} \leftarrow critic_network(states, Next_actions_{global})$
$GQ_{value} \leftarrow global_critic_network(states, Next_actions_{global})$
$MQ_{value} \leftarrow \min(Q_{value}, GQ_{value})$
$TD \leftarrow calculate_temporal_difference(rewards, MQ_{value}, dons, \gamma)$
Update critic_network by TD.
$Gradient \leftarrow sample_policy_gradient(actios, states)$
Update actor_network ← (states,gradient)
Update global_network weights by actor,critic and lr_g.
Return global actor-critic network

شکل (۷): شبه کد مراحل به روز رسانی الگوریتم پیشنهادی

مراحل به روز رسانی الگوریتم پیشنهادی عبارتند از:

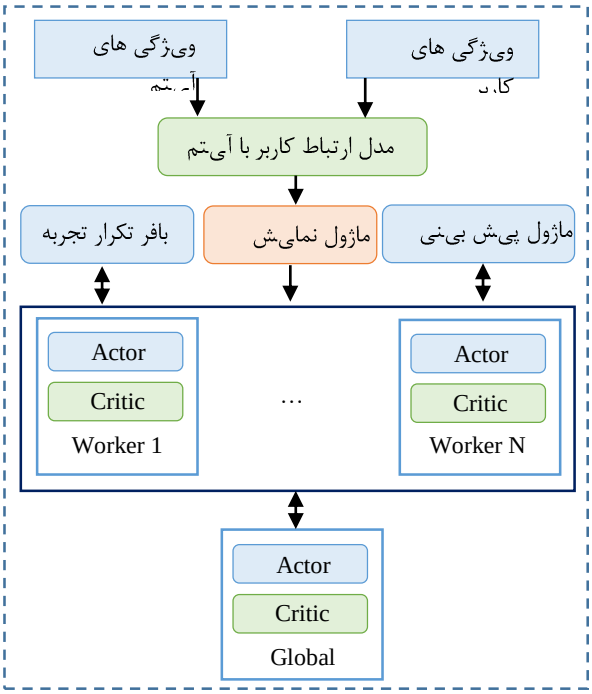
- ورود حالات بعدی استخراج شده از بافر به شبکه actor عمومی و گرفتن action بعدی
- گرفتن q value از شبکه critic با توجه با وضعیت‌های استخراج شده از بافر و action بعدی
- گرفتن q value از شبکه critic عمومی با توجه به state و action بعدی
- بدست آوردن حداقل مقدار کیو از کیوهای بدست آمده در دو مرحله قبلی
- محاسبه تفاوت زمانی از روی حداقل مقدار کیو
- آموزش شبکه critic با توجه به تفاوت زمانی بدست آمده
- محاسبه گرادیان و به روز رسانی شبکه actor
- به روزرسانی وزن شبکه actor و critic عمومی براساس وزن‌های جدید

۳-۹- فلوچارت الگوریتم پیشنهادی

در شکل (۸)، فلوچارت الگوریتم پیشنهادی به عنوان یک جمع بندی از بخش‌های مختلف آورده شده است. مطابق فلوچارت، ابتدا ویژگی‌های مربوط به تعامل بین کاربران وآیتم‌ها توسط مدل ارتباط کاربر و آیتم پردازش شده و لایه‌های جاسازی بوجود می‌آیند. این لایه‌ها توسط ماژول نمایش حالت به یک بردار ویژگی تبدیل می‌شوند تا توسط شبکه های actor و critic موجود در thread های مختلف مورد استفاده قرار گیرند. همچنین، شبکه‌های موجود در thread های مختلف در حین فرایند آموزش تجربیات خود را در بافر ذخیره کرده و از مدل پیش

[Downloaded from jiaeee.com on 2026-02-15]
[DOI: 10.61186/jiaeee.22.1.121]

بینی امتیاز برای پیش بینی امتیاز کاربران به آیتم‌هایی که در تاریخچه کاربر وجود ندارند استفاده می‌کنند و در نهایت با استفاده از تجربیات خود شبکه عمومی را به روزرسانی می‌کنند.



شکل (۸): فلوچارت الگوریتم پیشنهادی

۴- ارزیابی

در این بخش به سنجش عملکرد و کارایی الگوریتم پیشنهادی می‌پردازیم. بنابراین، تمامی پیش نیازها و شرایط لازم جهت انجام آزمایش‌های مورد نیاز را شرح داده و عملکرد الگوریتم پیشنهادی بر روی دیتاست‌های مختلف را از نظر میزان درستی و کیفیت امتیازها مورد بررسی قرار می‌دهیم. جهت ارزیابی الگوریتم پیشنهادی از ۳ دیتاست Movielens100k[15]، Movielens1m[16] و Jester[17] استفاده شده است که جزئیات آنها در جدول (۲) ذکر شده است.

جدول (۲): مجموعه داده‌ها

ردیف	نام	تعداد کاربر	تعداد آیتم ها	تعداد امتیاز
۱	Movielens100k	۹۴۳	۱۶۸۲	۱۰۰۰۰۰
۲	Movielens1m	۶۰۴۰	۳۹۵۲	۱۰۰۰۲۰۹
۳	Jester	۶۳۹۷۸	۱۵۰	۱۷۶۱۴۳۹

۴-۱- الگوریتم ارزیابی

در قسمت ارزیابی، به ازای هر کاربر وضعیت جاری را بدست می‌آوریم و با توجه به آن، از میان آیتم‌های موجود در قسمت دیتاست تست، با استفاده از الگوریتم پیشنهادی، آیتم‌های مورد نظر را به کاربر پیشنهاد می‌کنیم. در نهایت با توجه به امتیاز به دست آمده، الگوریتم پیشنهادی را مورد ارزیابی قرار می‌دهیم.

Algorithm 3: Evaluation algorithm

Input: state window size n.
foreach user in users **do**:
  Observe current state $s_t = f(H_t)$ where $H_t = \{i_1,\dots,i_n\}$.
  Execute action $a_t = \pi\theta(s_t)$ according to the current policy.
  Recommend item i_t according to action a_t.
  Calculate reward $r_t = R(s_t,a_t)$ based on the feedback of the user.
  Calculate precision and NDCG according to recommended items and rewards.
End foreach
Return precision and NDCG

شکل (۹): الگوریتم ارزیابی

۴-۲- تنظیمات پارامتریک

به ازای هر دیتاست، ۷۰ درصد از تعاملات کاربر برای آموزش، ۱۰ درصد آن برای تایید و ۲۰ در صد باقی مانده برای تست الگوریتم‌ها استفاده شده است. بازه امتیازدهی به آیتم‌ها در دیتاست movielens از ۱ تا ۵ است که با توجه به کار لیو و همکاران [۲۸]، امتیاز ۴ و ۵ به عنوان امتیاز مثبت در نظر گرفته شده است. همچنین، بازه امتیاز کاربران به آیتم‌ها در دیتاست jester از ۱۰- تا ۱۰+ است که با توجه به کار لیو و همکاران [۲۸]، امتیاز ۱ تا ۱۰ به عنوان امتیاز مثبت در نظر گرفته شده است. در مراحل آموزش الگوریتم ارائه شده از پیشنهاد آیتم‌های تکراری به کاربر جلوگیری شده است. همچنین، بازه امتیاز کاربران به آیتم‌ها به محدوده ۱- تا ۱+ نرمال سازی شده است، به طوری که در لحظه t زمانی که توصیه‌گر آیتمی را به کاربر توصیه می‌کند، امتیاز مربوطه از دیتاست استخراج شده و به کاربر نمایش داده می‌شود و در صورتی که در دیتاست نباشد از مدل پیش‌بینی کننده برای امتیاز دهی استفاده می‌شود. همانطور که قبلا هم اشاره شد، بازه امتیاز دهی کاربران به آیتم‌ها در دیتاست‌های movielens و jester با همدیگر یکسان نیست. به این دلیل، از توابع متفاوتی برای نرمال‌سازی امتیاز کاربران استفاده می‌شود. همچنین پارامترهای مورد استفاده در آزمایش‌ها نیز در جدول (۳) قابل مشاهده است.

تابع مربوط به محاسبه امتیاز در دیتاست های movielens بصورت فرمول (۳) است.

$$R(s,a) = (\frac{1}{2})(rate_{i,j} - 3) \qquad (۳)$$

تابع مربوط به محاسبه امتیاز در دیتاست jester بصورت فرمول (۴) ذکر شده است.

$$R(s,a) = rate_{i,j}/10 \qquad (۴)$$

[Downloaded from jiaeee.com on 2026-02-15]
[DOI: 10.61186/jiaeee.22.1.121]

در فرمول (۳) و (۴) با توجه به وضعیت s که کاربر در آن قرار دارد، با انجام عمل a آیتم‌هایی به کاربر توصیه می‌شوند که امتیاز این آیتم‌ها ($rate_{i,j}$) با توجه با تاریخچه تعاملات موجود در مجموعه داده‌ها و مدل پیش بینی کننده امتیاز بدست آمده و به بازه ۱- تا ۱+ نرمال‌سازی شده‌اند.

جدول (۳): پارامترهای استفاده شده در آزمایش‌ها

نام پارامتر	مقدار
تعداد آیتم‌های اخیری که کاربر با آنها تعامل مثبت داشته است	N=5 [30]
ضریب کاهش	0.9 [30]
نرخ یادگیری الگوریتم پیشنهادی	0.001
اندازه بافر تکرار تجربه	1000000
نرخ یادگیری مدل ارتباط کاربر با آیتم	0.001
نرخ یادگیری مدل پیش بینی کننده امتیاز	0.001

وضعیت کاربر در واقع با توجه به تاریخچه تعامل کاربر با آیتم‌های موجود در مجموعه داده‌ها بدست می‌آید. در اینجا برای مشخص کردن علاقه‌مندی‌های کاربر از تعاملات مثبت استفاده شده است. ضریب کاهش، فاکتوری برای اندازه گیری پاداش بلند مدت است که با صفر شدن آن توصیه‌گر فقط به پاداش‌های آنی اهمیت می‌دهد و با یک شدن آن اهمیت پاداش‌های بلند مدت و کوتاه مدت با هم برابر می‌شود. اندازه بافر تکرار تجربه با توجه به میزان حافظه اصلی موجود در سیستم بدست آمده است.

۴-۳- نتایج ارزیابی و مقایسه

برای ارزیابی کارایی الگوریتم پیشنهادی از دو معیار Precision@k و NDCG@k استفاده شده است. معیار precision به این سوال پاسخ می‌دهد که چه نسبتی از آیتم‌هایی که به کاربر توصیه شده‌اند واقعا درست هستند. معیار NDCG[18] نیز برای اندازه گیری کیفیت رتبه بندی است. این معیار در واقع سود تجمعی درجه‌بندی شده آیتم‌های پیشنهاد شده به کاربر را محاسبه می‌کند. در این قسمت میانگین نتایج حاصل از اجرای الگوریتم پیشنهادی با چند الگوریتم پایه و پرکاربرد و همچنین الگوریتم ارائه شده توسط لیو و همکاران مقایسه شده است که این روش‌ها به صورت زیر شرح داده می‌شوند:

- الگوریتم Popularity [۸] : در این روش در هر بازه زمانی از میان آیتم‌هایی که به کابر توصیه نشده‌اند، آیتمی با بالاترین میانگین امتیاز و یا آیتمی که بیشترین تعداد امتیاز مثبت را دارد به کاربر توصیه می‌شود.
- الگوریتم PMF [۸] : فاکتورسازی ماتریس احتمالی که یکی از روش‌های فاکتورگیری ماتریسی بوده و در فیلترینگ مشارکتی از آن استفاده می‌شود.
- الگوریتم SVD++ [۹] : ترکیب روش‌های مبتنی بر فیلترینگ مشارکتی که از تشابه کاربران و آیتم‌ها برای توصیه استفاده می‌کند و از روش‌های فاکتورگیری ماتریسی از جمله تجزیه مقادیر منفرد بهره می‌برد.
- الگوریتم DRR [۲۸] : الگوریتم ارائه شده توسط لیو و همکاران که بر پایه یادگیری تقویتی عمیق پیاده سازی شده است.

جدول (۴): مقایسه عملکرد الگوریتم پیشنهادی با چهار الگوریتم تحت بررسی بر روی دیتاست movielens 100K

مدل	Precision@5	Precision@10	NDCG@5	NDCG@10
Popularity	0.6933	0.6012	0.9104	0.9008
PMF	0.6988	0.6194	0.9095	0.8968
SVD++	0.7034	0.6255	0.9125	0.8991
DRR	0.7887	0.6935	0.9255	0.9046
الگوریتم پیشنهادی	**0.7979**	**0.7715**	**0.9494**	**0.9362**

جدول (۵): مقایسه عملکرد الگوریتم پیشنهادی با چهار الگوریتم تحت بررسی بر روی دیتاست movielens 1M

مدل	Precision@5	Precision@10	NDCG@5	NDCG@10
Popularity	0.7141	0.6181	0.9806	0.9738
PMF	0.7072	0.6193	0.9801	0.8746
SVD++	0.7142	0.6258	0.9009	0.8776
DRR	0.7693	0.6594	0.9112	0.8980
الگوریتم پیشنهادی	**0.7889**	**0.7834**	**0.9508**	**0.9404**

جدول (۶): مقایسه عملکرد الگوریتم پیشنهادی با چهار الگوریتم تحت بررسی بر روی دیتاست Jester

مدل	Precision@5	Precision@10	NDCG@5	NDCG@10
Popularity	0.6167	0.6012	0.8932	0.8703
PMF	0.6171	0.6015	0.8740	0.8676
SVD++	0.6184	0.6027	0.8819	0.8614
DRR	0.6278	0.6076	0.9124	0.9079
الگوریتم پیشنهادی	**0.8593**	**0.8611**	**0.9563**	**0.9484**

با مشاهده نتایج بدست آمده به این نتیجه می‌رسیم که الگوریتم ارائه شده در این تحقیق نسبت به الگوریتم ارائه شده توسط لیو و همکاران و روش‌های پایه و پرکاربرد از دقت بیشتری برخوردار است. لازم به ذکر است که نتایج مرتبط با الگوریتم های Popularity ، PMF ، SVD++ و DRR از مرجع [۲۸] استخراج شده اند.

[Downloaded from jiaeee.com on 2026-02-15]

[DOI: 10.61186/jiaeee.22.1.121]

۵- بحث و بررسی

همانطور که قبلا هم بیان شد و با توجه به نتایج بدست آمده، در حالت کلی الگوریتم پیشنهادی در هر سه دیتاست نسبت به الگوریتم ارائه شده توسط لیو و همکاران و سایر الگوریتم‌های پایه و مرسوم از دقت بالاتری برخوردار است که دلایل آن در زیر بیان شده است.

۵-۱- آماده‌سازی بهتر داده‌ها و استخراج ویژگی‌های بهتر

یکی از مراحل مهم در یادگیری ماشین آماده‌سازی و پیش پردازش داده‌ها قبل از شروع فرآیند یادگیری است. چرا که استخراج و انتخاب ویژگی‌های مناسب به طور چشمگیری می‌تواند باعث بهبود کارایی الگوریتم یادگیری شود. در اینجا به دلیل وجود ویژگی‌های از نوع متن در دیتاست های استفاده شده، ابتدا با استفاده از رویکردهای مختلف بردارسازی متن، داده‌های مورد نیاز به بردار عددی آنها تبدیل شده است. سپس، با طراحی یک شبکه عصبی، با توجه به امتیازاتی که کاربران به آیتم‌ها داده اند، شبکه مورد نظر را آموزش داده‌ایم. همانطور که قبلا نیز شرح داده شد، شبکه مذکور شامل دو لایه جاسازی برای ویژگی‌های کاربر و آیتم است که با آموزش شبکه، وزن‌های موجود در این لایه ها با توجه به امتیاز کاربر تغییر پیدا کرده و سپس از بردارهای موجود در لایه‌های جاسازی برای یادگیری الگوریتم پیشنهادی استفاده شده است.

۵-۲- به روزرسانی کارآمدتر شبکه

الگوریتم پیشنهادی شامل یک شبکه عمومی و چندین شبکه actor و critic مجزا است که هر کدام در thread های مختلف اجرا شده و تجربه خود را از تعامل بین کاربر و آیتم بدست می‌آورند و پس از بهبود، وزن خود را در شبکه عمومی کپی می‌کنند. در این مرحله مجموعه‌ای از گام‌های انتقال از بافر استخراج می‌شوند تا مراحل به روزرسانی الگوریتم شروع شود و در نهایت بتوانیم آیتم‌های مطلوبی را به کابر توصیه کنیم.

۵-۳- استفاده از شبکه‌های عصبی و یادگیری تقویتی عمیق

در الگوریتم‌های پایه و مرسوم ارائه شده از روش‌های ریاضی از جمله میانگین‌گیری از مجموع امتیاز آیتم‌ها، بدست آوردن بیشترین تعداد امتیاز مثبت، فاکتورگیری ماتریسی و فیلترینگ مشارکتی استفاده شده است. در صورتی که، در الگوریتم پیشنهادی با استفاده از شبکه‌های عصبی، ویژگی‌های کاربران و آیتم ها به بردارهای جاسازی تبدیل شده و الگوریتم پیشنهادی با استفاده از روش‌های یادگیری تقویتی عمیق، ارتباط میان کاربران و آیتم‌ها را با توجه به این بردارها شناسایی کرده و با گذر زمان با توجه به وضعیت کاربر، بهترین آیتم‌ها را به کاربران توصیه می‌کند.

۶- نتیجه‌گیری

در این مقاله لزوم استفاده از سیستم‌های توصیه‌گر در مواجهه با حجم عظیم کالاها و خدماتی که هم اکنون در بستر اینترنت در اختیار کاربران قرار می‌گیرند، توضیح داده شده و یک رویکرد کارآمد نیز در این ارتباط ارائه شده است. تاکنون روش‌های مختلفی از جمله فیلترینگ مشارکتی، فاکتورگیری ماتریسی، رگرسیون لجستیک و شبکه‌های عصبی در زمینه سیستم‌های توصیه‌گر ارائه شده‌اند. الگوریتم‌های یاد شده دارای محدودیت‌های خاصی هستند که از جمله آنها می‌توان به نگاه کردن به سیستم توصیه‌گر به عنوان یک پارامتر ثابت و عدم توجه به تعاملات کاربر با آیتم‌ها در گذر زمان و تمرکز بر روی پاداش‌های آنی و عدم توجه به پاداش‌های بلند مدت اشاره کرد. برای برطرف کردن این محدودیت‌ها، در این تحقیق یک الگوریتم مبتنی بر یادگیری تقویتی عمیق ارائه شده است تا تصمیم‌های خود را به صورت یک فرایند پویا با گذر زمان بهبود داده وعلاوه بر امتیاز آنی به تاثیر آن در پاداش های بلند مدت نیز اهمیت دهد. الگوریتم ارائه شده در این مقاله یک الگوریتم بهبود یافته بر اساس یادگیری تقویتی عمیق است که با استفاده از روش actor-critic پیاده سازی شده است. در این الگوریتم بخش actor عمل مورد نیاز را تعیین می‌کند و این عمل توسط قسمت critic مورد ارزیابی قرار می‌گیرد تا بر اساس ارزیابی انجام شده شبکه بتواند تخمین خود را بهبود دهد. از جمله مشکلاتی که در بهبود مدل وجود دارد می‌توان به همبستگی نمونه‌های آموزشی و اهداف غیر ثابت در شبکه اشاره کرد که با روش‌های بافرینگ و شبکه هدف مجزا این مشکلات رفع شده است. نتایج حاصل از آزمایش‌ها بر اساس دو معیار Precision و NDCG نشان می‌دهد که الگوریتم پیشنهادی از الگوریتم ارائه شده توسط لیو و همکاران و سه الگوریتم پایه‌ای و پرکاربرد دیگر بهتر عمل کرده است.

مراجع

[1] R. J. Kuo and S.-S. Li, "Applying particle swarm optimization algorithm-based collaborative filtering recommender system considering rating and review", Applied Soft Computing, vol. 135, p. 110038, 2023/03/01/ 2023.

[2] Y. Koren, S. Rendle, and R. Bell, "Advances in collaborative filtering", Recommender systems handbook, pp. 91-142, 2021.

[3] R. J. Mooney and L. Roy, "Content-based book recommending using learning for text categorization", presented at the Proceedings of the fifth ACM conference on Digital libraries, San Antonio, Texas, USA, 2000.

[4] D. Wang, Y. Liang, D. Xu, X. Feng, and R. Guan, "A content-based recommender system for computer science publications", Knowledge-Based Systems, vol. 157, pp. 1-9, 2018/10/01/ 2018.

[5] S. Rendle, W. Krichene, L. Zhang, and J. Anderson, "Neural collaborative filtering vs. matrix factorization revisited", in Proceedings of the 14th ACM Conference on Recommender Systems, 2020, pp. 240-248.

[Downloaded from jiaeee.com on 2026-02-15]

[DOI: 10.61186/jiaeee.22.1.121]

[6] Y. Koren, R. M. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems", IEEE Computer, vol. 42, no. 8, pp. 30–37, 2009.
[7] S. Rendle, W. Krichene, L. Zhang, and J. Anderson, "Neural collaborative filtering vs. matrix factorization revisited", in Proceedings of the 14th ACM Conference on Recommender Systems, 2020, pp. 240-248.
[8] A. Mnih and R. R. Salakhutdinov, "Probabilistic matrix factorization", Advances in neural information processing systems, vol. 20, 2007.
[9] Y. Koren, "Factorization meets the neighborhood: a multifaceted collaborative filtering model", in Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, 2008, pp. 426–434.
[10] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning", Nature, vol. 521, no. 7553, pp. 436-444, 2015/05/01 2015, doi: 10.1038/nature14539.
[11] W. Zhang, T. Du, and J. Wang, "Deep learning over multi-field categorical data - - A case study on user response prediction", in ECIR 2016, Padua, Italy, March 20-23, 2016. Proceedings, 2016, pp. 45–57.
[12] H. Cheng, L. Koc, J. Harmsen, T. Shaked, T. Chandra, H. Aradhye, G. Anderson, G. Corrado, W. Chai, M. Ispir, R. Anil, Z. Haque, L. Hong, V. Jain, X. Liu, and H. Shah, "Wide & deep learning for recommender systems", CoRR, vol. abs/1606.07792, 2016.
[13] K. Falk, Practical recommender systems. Simon and Schuster, 2019.
[14] V. François-Lavet, P. Henderson, R. Islam, M. G. Bellemare, and J. Pineau, "An Introduction to Deep Reinforcement Learning", Foundations and Trends® in Machine Learning, vol. 11, no. 3-4, pp. 219-354, 2018, doi: 10.1561/2200000071.
[15] T. Hickling, A. Zenati, N. Aouf, and P. Spencer, "Explainability in deep reinforcement learning: A review into current methods and applications", ACM Comput. Surv., vol. 56, no. 5, dec 2023.
[16] Molaei M, Amirkhani A. Policy-based Auto-Driving in Highway based on Distributional Reinforcement Learning Methods. Journal of Iranian Association of Electrical and Electronics Engineers 2022; 19 (2) :209-222
[17] Beigi A, Akbarian A. Profit increasing in smart grid market via actor-critic reinforcement learning. Journal of Iranian Association of Electrical and Electronics Engineers 2022; 19 (1) :245-258
[18] X. Chen, L. Yao, J. McAuley, G. Zhou, and X. Wang, "Deep reinforcement learning in recommender systems: A survey and new perspectives", Knowledge-Based Systems, vol. 264, p. 110335, 2023.
[19] Y. Zhang, C. Zhang, and X. Liu, "Dynamic scholarly collaborator recommendation via competitive multi-agent reinforcement learning", in Proceedings of the Eleventh ACM Conference on Recommender Systems, ser. RecSys '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 331–335.
[20] X. Chen, S. Li, H. Li, S. Jiang, Y. Qi, and L. Song, "Generative adversarial user model for reinforcement learning based recommendation system", in International Conference on Machine Learning. PMLR, 2019, pp. 1052–1061.
[21] X. Bai, J. Guan, and H. Wang, "A model-based reinforcement learning with adversarial training for online recommendation", Advances in Neural Information Processing Systems, vol. 32, 2019.
[22] D. Hong, Y. Li, and Q. Dong, "Nonintrusive-sensing and reinforcementlearning based adaptive personalized music recommendation", in Proceedings of the 43rd International ACM SIGIR conference on research and development in information retrieval, 2020, pp. 1721–1724.
[23] X. Zhao, L. Xia, L. Zou, H. Liu, D. Yin, and J. Tang, "Wholechain recommendations", in Proceedings of the 29th ACM international conference on information & knowledge management, 2020, pp. 1883–1891.
[24] A. Montazeralghaem and J. Allan, "Extracting relevant information from user's utterances in conversational search and recommendation", in Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2022, pp. 1275–1283.
[25] G. Zheng, F. Zhang, Z. Zheng, Y. Xiang, N. J. Yuan, X. Xie, and Z. Li, "Drn: A deep reinforcement learning framework for news recommendation", in Proceedings of the 2018 world wide web conference, 2018, pp. 167–176.
[26] X. Zhao, C. Gu, H. Zhang, X. Yang, X. Liu, J. Tang, and H. Liu, "Dear: Deep reinforcement learning for online advertising impression in recommender systems", in Proceedings of the AAAI conference on artificial intelligence, vol. 35, no. 1, 2021, pp. 750–758.
[27] F. Pan, Q. Cai, P. Tang, F. Zhuang, and Q. He, "Policy gradients for contextual recommendations", in The World Wide Web Conference, 2019, pp. 1421–1431.
[28] F. Liu, R. Tang, X. Li, W. Zhang, Y. Ye, H. Chen, H. Guo, and Y. Zhang, "Deep reinforcement learning based recommendation with explicit useritem interactions modeling", arXiv preprint arXiv:1810.12027, 2018.
[29] Q. Cai, A. Filos-Ratsikas, P. Tang, and Y. Zhang, "Reinforcement mechanism design for e-commerce", in Proceedings of the 2018 World Wide Web Conference, 2018, pp. 1339–1348.
[30] K. Zhao, X. Wang, Y. Zhang, L. Zhao, Z. Liu, C. Xing, and X. Xie, "Leveraging demonstrations for reinforcement recommendation reasoning over knowledge graphs", in Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval, 2020, pp. 239–248.
[31] M. van Otterlo and M. Wiering, "Reinforcement Learning and Markov Decision Processes", in Reinforcement Learning: State-of-the-Art, M. Wiering and M. van Otterlo Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 3-42.

زیرنویس‌ها

[1] Content based filtering
[2] Matrix factorization
[3] Deep learning
[4] Recommender systems
[5] Deep reinforcement learning
[6] Model-based
[7] Model-free
[8] Value-based
[9] Policy-based
[10] Hybrid
[11] Discount rate
[12] Embedding
[13] Dense
[14] Done
[15] https://grouplens.org/datasets/movielens/100k/
[16] https://grouplens.org/datasets/movielens/1m/
[17] http://eigentaste.berkeley.edu/dataset/
[18] Normalized discounted cumulative gain

[Downloaded from jiaeee.com on 2026-02-15]
[DOI: 10.61186/jiaeee.22.1.121]