Image Processing, Electronics and Computers
L. Trajkovic et al. (Eds.)

doi:10.3233/ATDE240495

Research on Efficient Information Retrieval Method Based on Recurrent Neural Network

Yanlin ZENG[1]
School of Petroleum and Natural Gas Engineering, Chongqing University of Science and Technology, Chongqing. 401331 China

Abstract. In the era of information explosion, efficiently retrieving relevant information from vast datasets has become a paramount challenge, with deep learning offering new avenues for enhancing information retrieval (IR) systems. This study introduces an innovative Recurrent Neural Network (RNN)-based approach to significantly improve the efficiency and accuracy of IR tasks, surpassing the limitations of traditional methods. By designing a specialized RNN architecture and comparing it against traditional and existing deep learning-based IR models across diverse datasets and various IR tasks—including document retrieval, question answering, and semantic search—our methodology demonstrates remarkable improvements in retrieval efficiency and accuracy. Specifically, the proposed model achieved substantial increases in precision, recall, and F1 score, highlighting the potential of RNNs in mastering the sequential and contextual nuances inherent in IR processes. This advancement offers a robust and efficient solution to the challenges of traditional IR systems, with future work aiming to integrate this model into broader IR applications and explore further enhancements using advanced neural network techniques.

Keywords. Personalized recommendation algorithms, migration learning, multi-source multi-tasking, deep learning; ontologies

1. Introduction

With the rapid development of Internet technology, data on the network has been growing explosively. On the one hand, the huge amount of data makes it easier for people to get rich information, on the other hand, people have to spend a lot of energy and time to search for information useful to them, and the problem of information overload is becoming more and more serious. In the face of massive data resources, traditional search engines have been unable to meet the needs of users, personalized recommendation system has become the new favorite of the times. Personalized recommendation system captures users' interests by analyzing their data and recommends information or products of interest to them.

The emergence of recommendation systems has changed the way users obtain information, i.e., from simple target and clear data search to a context-rich information acquisition method that is more in line with people's usage habits. At present, the landscape of recommendation systems is segmented into several main approaches: those

[1] Corresponding Author: Yanlin Zeng, m13452998136@163.com.

reliant on collaborative filtering [1], recommendations based on content analysis [2], systems that utilize knowledge-based mechanisms [3], and the integrated or hybrid models. The essence of collaborative filtering lies in the assumption that if users have exhibited similar interests previously, they are likely to continue doing so, effectively participating in an unseen form of collaboration [4]. The Latent Dirichlet Allocation (LDA) topic model is employed for the analysis of textual content, assigning texts to a more compact topic space [5]. The primary benefit of content-based systems is their independence from a large user database, a contrast to the collaborative filtering approach. Yet, the acquisition of relevant feature data presents a significant hurdle for content-based recommendations, alongside the challenge of introducing novel suggestions. Xu introduces a methodology that draws upon the Elo Rating System for comparing ratings, aimed at bridging the divide between newly introduced (cold start) users and those already active, through an analysis of the variance in ratings given to items by these two groups [6]. H. Lin is concerned with recommender systems for crowdsourced task recommendation [7]: on crowdsourcing platforms, users generally do not comment explicitly on tasks. H. Lin suggests a crowdsource task recommendation strategy based on task types and users' task completion and selection frequency. As the field of recommendation algorithms progresses, novel methodologies, including those predicated on social networks [8] and contextual data [9], have been developed. Recommendations utilizing social networks capitalize on the intricacies of social connections, trust, and network graphs to deliver personalized suggestions. On the other hand, recommendations informed by contextual information draw upon variables such as time, location, nearby individuals, emotional states, and activities for customization. The deepening exploration into deep learning has piqued interest in applying its potential to recommender systems [10]. Deep learning outperforms traditional algorithms with its superior capacity for autonomously identifying abstract characteristics, and its application is bolstered by developments in distributed computing technologies, which have significantly advanced big data processing capabilities. The integration of deep learning with recommender systems, aiming to precisely model user interests from seemingly chaotic data, has emerged as a focal point of innovation in the recommender system domain.[11]

To sum up, it is imperative to incorporate machine learning into personalized recommendation algorithms in order to achieve effectiveness and efficiency. The study of personalized recommendation algorithms based on machine learning has important theoretical value and practical application significance. [12]In this paper, to address the problems of cold start, data sparsity, and low accuracy of existing recommendation systems, machine learning methods are introduced, cross-domain recommendation with migration learning, interest recommendation with multiple sources and multiple tasks, hybrid recommendation with BP-RNN, and ontology-based travel recommendation are studied, and the corresponding algorithm research is designed for realizing personalized recommendation.

2. Method

In this paper, a short-term recommendation item is designed to recommend items that the user is likely to purchase in the next moment. If the user ui watched DieHard1, DieHard2 and DieHard3, thus presumably, the next time the user is likely to watch DieHard4, only the short-term behavior of the user is focused here, i.e., the time range

closer to the current time. By analyzing the user's recent behavior, the next most likely item to be consumed by the user is calculated and predicted.

2.1. Backward Propagation Algorithm

In the operation of a neural network, the process whereby input \(x\) is transformed into output \(y\) involves the sequential transmission of data through the network's structure. Initially, the input \(x\) introduces fundamental data that is forwarded through successive hidden layers, culminating in the output \(y\). This sequential data processing is known as forward propagation. Throughout the training phase of a neural network, upon reaching the network's final layer, a discrepancy is often observed between the neural network's output and the actual desired output. To address this, the backward propagation (bp) algorithm is employed, enabling the reversal of error information flow from the output layer back through the network to the input layer via the hidden layers. This reverse flow facilitates the adjustment of the network's internal parameters based on the error, with the aim of refining the network's accuracy through repeated iterations until the model achieves a state of convergence. [13] It's a common misconception that back propagation is the sole mechanism for learning in multilayer neural networks, whereas it specifically refers to the method of error correction and parameter adjustment. However, backpropagation refers only to the method used for gradient computation, while stochastic gradient descent, for example, uses the gradients computed by the backpropagation algorithm to perform learning. The following explains in detail how neural networks are trained to learn using backpropagation.

Figure 1 shows a is a three-layer neural network, which includes an input layer, a hidden layer and an output layer. First, some of the symbols are defined as follows:

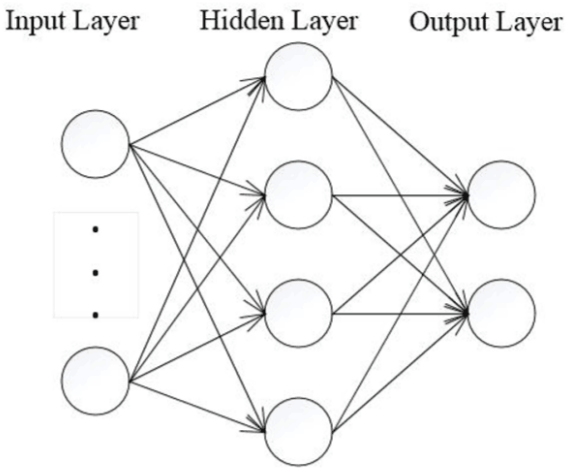

Figure 1. Three layers neural networks

W_{jk}^{l} represents the weight connecting the i^{th} neuron in the k^{th} layer to the j^{th} neuron in the l^{th} layer.;

b_{j}^{l} indicates the bias term for the j^{th} neuron within the l^{th} layer.;

z_j^l signifies the input value for the i[th] neuron in the k[th] layer, determined through calculation as

$$z_j^l = \sum kW_{jk}^l a_j^{l-1} + b_j^l \tag{1}$$

a_j^l represents the output from the j[th] neuron in the l[th] layer, obtained by calculation as

$$a_j^l = \sigma \sum kW_{jk}^l a_j^{l-1} + b_j^l \tag{2}$$

σ is the activation function, which is usually a sigmoid function.

The Backpropagation (BP) algorithm operates in two main phases: forward propagation and backward propagation. During forward propagation, information flows unidirectionally from the input layer, through any hidden layers, and ultimately to the output layer, resulting in the network's overall output. This means the output from one layer in the neural network becomes the input for the subsequent layer $a^l = \sigma \sum W^l a^{l-1} + b^l$. For the output layer[14], we consider yl where x is the input dataset represented as a column vector. In the context of the hidden layer, and subsequently for the output layer, the forward propagation calculations can be conducted using the formula provided earlier in the document.

$$J(Q, b; x, y) = \frac{1}{2}\left\| h_{W,b}(x) - y \right\|^2 \tag{3}$$

During the neural network's training phase, the gradient descent technique is employed to optimize the network. The parameters W (weights) and b (biases) are updated in each iteration of the gradient descent process according to the specified formula.

$$W_{ij}^{(l)} = W_{ij}^{(l)} - \alpha \frac{\partial}{\partial W_{ij}^{(l)}} J(W, b) \tag{4}$$

$$b_{ij}^{(l)} = b_{ij}^{(l)} - \alpha \frac{\partial}{\partial b_{ij}^{(l)}} J(W, b) \tag{5}$$

Where α denotes the learning rate speed. The most critical step in the update is to calculate the partial derivatives using back propagation, and back propagation propagation is a very effective method to calculate the partial derivatives.

Once this partial derivative is computed, it is easy to derive the partial derivative of the overall cost function J (W, b). The results of the calculation are shown below:

$$\frac{\partial}{\partial W_{ij}^{(l)}} \mathrm{J}\,(\mathrm{W}, \mathrm{b}) = \left[\frac{1}{m} \sum_{i=1}^{m} \frac{\partial}{\partial W_{ij}^{(l)}} \mathrm{J}\left(\mathrm{W}, \mathrm{b}; x^{(i)}, y^{(i)}\right) \right] + \lambda W_{ij}^{(l)} \tag{6}$$

$$\frac{\partial}{\partial b_{ij}^{(l)}} \mathrm{J}\,(\mathrm{W}, \mathrm{b}) = \frac{1}{m} \sum_{i=1}^{m} \frac{\partial}{\partial b_{ij}^{(l)}} \mathrm{J}\left(\mathrm{W}, \mathrm{b}; x^{(i)}, y^{(i)}\right) \tag{7}$$

1. Perform feed forward conduction calculations and use the forward propagation equation to obtain L1, L2... .LN up to the activation value of the output layer l nL.

2. For each output cell i in the l nth layer, calculate the residuals (the residuals show how much influence the node has on the residuals of the final output values)

2.2. Recurrent Neural Networks

Recurrent neural networks have shown great power in the task of processing sequential data and have become a very popular model for processing sequential data. Figure 2 shows the design of a recurrent neural network expanded into a complete network.

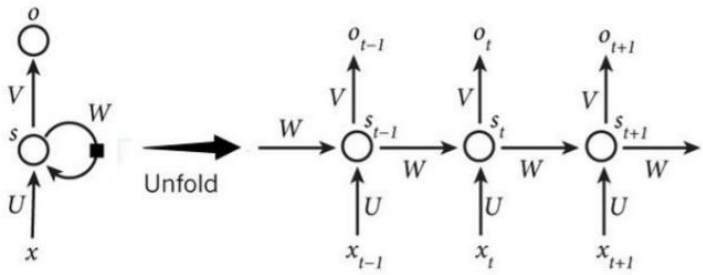

Figure 2. Unfold recurrent neural networks

A recurrent neural network is constructed by unfolding it into a complete sequence of data. If the user has 6 historical interactions with the system, then the network is expanded into a 6-layer neural network, one layer for each user-system interaction.

The forward propagation formulation of the RNN in Figure 3, where the activation function of the hidden unit is not specified in the Figure 3, assumes that a hyperbolic tangent activation function is used. In addition, the output and loss function of any form are not explicitly stated in the Figure 3. The output is assumed to be discrete, such as an RNN for predicting characters or for predicting the movie that a user will watch. A natural way to represent discrete variables is to treat the output o as an unnormalized log probability of the possible values of each discrete variable.[15] Then, a Softmax function can be applied for subsequent processing.

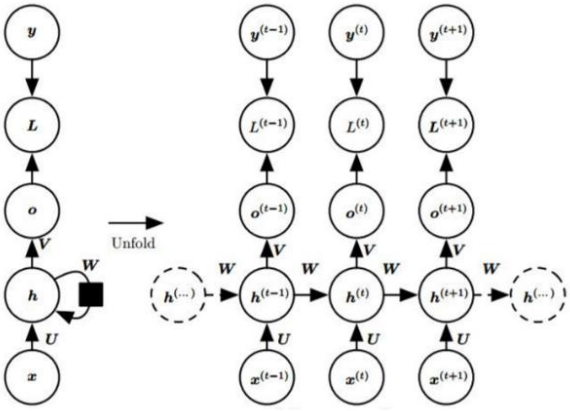

Figure 3. RNN training loss computing

3. Experiment

3.1. Experimental Platform and Environment

Hardware configuration: Intel (R) Core (R) G1820, main frequency 2.70Ghz, 6GB RAM operating system: Ubuntu12.04 development language: Python deep learning framework: TensorFlow libraries: NumPy; Matplotlib

3.2. Data Sets and Evaluation Metrics

The recommendation algorithms are based on common recommendation algorithms such as user-based collaborative filtering algorithm, item-based collaborative filtering algorithm, and matrix factorization algorithm, compared with recurrent neural network-based recommendation algorithm and improved recurrent neural network-based algorithm, and finally a complete recommendation algorithm framework based on recurrent neural network is built for experiments. MovieLens is commonly used to evaluate the performance of recommendation system algorithms and is used as the experimental data set in this chapter. Based on 1682 movies watched by 943 users, the dataset (MovieLens100K) has 100000 ranks of data from 1-5. Each user has watched at least 20 movies. And the data is arranged randomly. The table contains the user ID, the subject ID, the rating and the time. The timestamps are Unix from 1/1/1970 UTC. different methods are compared by a series of precisely designed metrics to evaluate the performance of the recommendation algorithm. Short-termPredictionSuccess (SPS): characterizes the ability of the algorithm to predict the next movie to be watched by the user. Recall: characterizes the long-term prediction ability of the recommendation algorithm.

3.3. Experimental Results and Analysis

To enhance the accuracy of short-term forecasts within a recommendation system, expanding the selection of candidate items proves to be a beneficial strategy. Imagine a scenario where a movie website has a catalog of 5 films. Recommending a single movie to a user might yield a relatively low success rate for short-term predictions. However, in a hypothetical scenario where all 5 movies are recommended simultaneously, the short-term prediction success rate would effectively reach 100%.

Table 1. Comparison of different recommendation algorithms in movie footage 100K

	SPS@1	SPS@5	SPS@10	recall@5	recall@10
MHR	0.00	0.00	0.66	0.02	0.17
MP	0.41	3.25	17.44	3.79	3.83
User-CF	0.00	1.60	2.80	1.60	2.90
Item-CF	0.00	1.70	2.90	1.60	3.00
MF-CF	0.00	2.30	3.40	1.90	3.20
RNN	9.54	29.59	40.51	4.50	7.50
RNN-BP	11.27	31.62	42.34	4.30	7.60

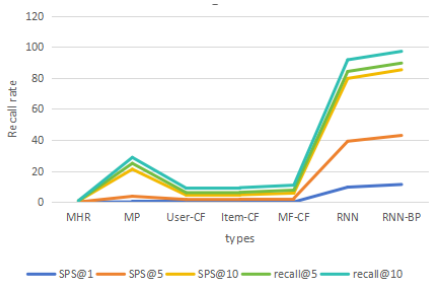

Figure 4. Comparison of different recommendation algorithms in movie footage 100K

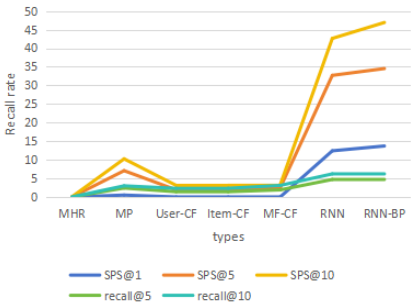

Figure 5. MovieLens1M Comparison table of different recommendation algorithms

3.4. Model Comparison

The Sequential Probability Score (SPS) and Recall metrics for various models on the MovieLens100K and MovieLens1M datasets are compared and presented in Tables 1 and 2. Additionally, these comparisons are visually represented through line graphs in Figures 4 and 5.

Table 2. Comparison Table of Various Recommendation Algorithms on MovieLens1M Dataset

	SPS@1	SPS@5	SPS@10	recall@5	recall@10
MHR	0.00	0.00	0.06	0.01	0.01
MP	0.50	7.06	10.24	2.39	2.99
User-CF	0.00	1.90	3.10	1.40	2.30
Item-CF	0.00	1.94	3.12	1.40	2.32
MF-CF	0.00	2.5	3.30	1.90	3.10
RNN	12.4	32.69	42.63	4.70	6.20
RNN-BP	13.7	34.52	46.91	4.70	6.20

It is clear that the improved BP-RNN recommendation algorithm has improved the short-term prediction accuracy SPS and recall recall evaluation, especially in the prediction accuracy. Compared with the basic RNN, the improved BP-RNN recommendation algorithm proposed in this chapter has better recommendation results, which indicates that it is feasible to combine the RNN with BP neural network when analyzing user sequence data.

4. Discussion

Collaborative filtering algorithm is an algorithm for predicting and recommending users' behaviors and preference levels in a system, which has been widely used in recent years. Since the implementation of collaborative filtering algorithm needs to obtain all behaviors and preference levels of users for prediction, and cannot reflect the recent behaviors and preferences of users, the collaborative filtering algorithm may ignore the behaviors or preferences of users that have changed over time. In short-term recommender systems, the most likely consumption of users often depends on recent

behaviors. In this paper, we address the phenomenon that user consumption in short-term recommender systems is most likely to depend on recent behaviors by building a recurrent neural network that uses gated recurrent units to solve the problem of time sequences. The recurrent neural network is used to treat the user's recent behavior as a sequence, and each hidden layer simulates each user's behavior or preference in a sequential manner. The experimental results demonstrate that combining recurrent neural networks with back propagation neural networks for recommendation can achieve higher prediction accuracy and can effectively overcome the problems inherent in traditional algorithms.

5. Conclusion

In this paper, we focus on the personalized recommendation technology based on machine learning. Compared with traditional recommendation algorithms, this paper takes advantage of migration learning, multi-source multi-task learning and deep learning, which are more representative in the field of machine learning, to perform personalized recommendation and effectively overcome the inherent problems of traditional algorithms. However, the basic assumption of the cross-domain recommendation model with migration learning proposed in this paper is that the domains are homogeneous, i.e., the characteristics of each domain are similar in nature. Future work can explore how cross-domain recommendation algorithms can be investigated on domains that are more general, i.e., non-homogeneous. In addition, the algorithm proposed in this paper is not effective enough in recommending when there are no or few overlapping users in each domain dataset, so it can be explored how to better perform cross-domain recommendation when the dataset does not contain overlapping users between domains.

References

[1] J. Cho, K. Kwon, and Y. Park, "Collaborative filtering using dual information sources," IEEE Intelligent Systems, vol. 22, no. 3, pp. 30-38, 2007.
[2] M. Pazzani and D. Billsus, "Content-based recommendation systems," The Adaptive Web, Springer, Berlin, Heidelberg, LNCS, volume 4321, pp. 325-341, 2022.
[3] R. Burke, "Knowledge-based recommender systems," in Encyclopedia of Library and Information Systems, vol. 69, pp. 175-186, 2020.
[4] D. Jannach, M. Zanker, A. Felfernig, and G. Friedrich, "Machine Learning-Based Recommender Systems: Status Quo and Future Directions," IEEE Transactions on Knowledge and Data Engineering, vol. 34, no. 11, pp. 193-202, 2022.
[5] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet Allocation," Journal of Machine Learning Research, vol. 3, pp. 993-1022, 2021.
[6] J. Xu, Y. Yao, and T. Tong, "Ice-Breaking: Mitigating Cold-Start Recommendation Problem Comparison," in Proceedings of the International Joint Conference on Artificial Intelligence, pp. 1202-1208, 2015.
[7] C. HLin, E. Kamar, and E. Horvitz, "Signals in the Silence: Models of Implicit Feedback in a Recommendation System for Crowdsourcing," Proceedings of the National Conference on Artificial Intelligence, pp. 247-255, 2019.
[8] J. He and W. W. Chu, "A Social Network-Based Recommender System (SNRS)," in Proceedings of the IEEE Conference on Computer Communications, pp. 47-74, 2022.
[9] G. Adomavicius and A. Tuzhilin, "Context-Aware Recommender Systems," in Proceedings of the IEEE Conference on Recommender Systems, pp. 2175-2178, 2018.

[10] Xu L, Duan Y, Pei J, et al. PCA-RNN-based intelligent mobile drone spectrum sensing algorithm//Proceedings of the 5th International ACM Mobicom Workshop on Drone Assisted Wireless Communications for 5G and Beyond. 31-36, 2022.
[11] M. Elkahky, Y. Song, and X. He, "A multi-view deep learning approach for cross domain user modeling in recommendation systems," in Proceedings of the 24th international conference on world wide web, pp. 278-288, 2015.
[12] S. Zhang, L. Yao, A. Sun, and Y. Tay, "Deep learning based recommender system: A survey and new perspectives," ACM computing surveys (CSUR), vol. 52, no. 1, pp. 1-38, 2019.
[13] M. Salehi, I. Nakhai Kamalabadi, and M. B. Ghaznavi Ghoushchi, "Personalized recommendation of learning material using sequential pattern mining and attribute based collaborative filtering," Education and Information Technologies, vol. 19, pp. 713-735, 2014.
[14] J. Hur and S. Roth, "Iterative residual refinement for joint optical flow and occlusion estimation," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 5754-5763, 2019.
[15] J. Pei, S. Li, Z. Yu, L. Ho, W. Liu and L. Wang, "Federated Learning Encounters 6G Wireless Communication in the Scenario of Internet of Things," in IEEE Communications Standards Magazine, vol. 7, no. 1, pp. 94-100, March (2023)