Expert Systems With Applications 140 (2020) 112887

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Mitigating long tail effect in recommendations using few shot learning technique

Rama Syamala Sreepada, Bidyut Kr. Patra*

Department of Computer Science and Engineering, National Institute of Technology Rourkela, Odisha, India

A R T I C L E I N F O

Article history:
Received 18 July 2018
Revised 8 July 2019
Accepted 17 August 2019
Available online 19 August 2019

Keywords:
Long tail items
Hierarchical clustering
Few shot learning
Siamese networks

A B S T R A C T

Recommender system has been established as an effective tool for users in providing personalized suggestions in many domains, especially in e-commerce. In these domains, recommendations are provided based on the feedback (ratings) given by the users. However, recommendations provided by the traditional approaches are biased towards the popular items (items that receive more number of ratings). As a result, unpopular items are left out and these items remain un-recommended and unsold. These unpopular items form a “long tail” in the product space, resulting in a huge loss to the e-commerce industry. However, diminishing this long tail effect is a highly challenging and non-trivial task due to limited available rating information. Recommending long tail items helps in improving the item liquidation and recommendation diversity as well.

In this paper, we propose a novel framework to mitigate the long tail effect and overcome the limited ratings problem using few shot learning techniques. Siamese network, a type of few shot learning technique is found to be performing well in many domains with a limited number of instances in the recent past. In the proposed framework, vital statistics of each user are computed and this information is provided to deep siamese network. The trained siamese network is used to identify the long tail items that are similar to the liked items of each user. Finally, the identified long tail items are recommended to the appropriate users. We introduce three novel performance metrics to evaluate the long tail item recommendations. The proposed framework is evaluated on two real world datasets (MovieLens 1M and Netflix) and the results demonstrate that the proposed framework outperforms the traditional approaches and existing long-tail recommendation techniques.

1. Introduction

Recommender System (RS), which falls under the area of information retrieval, has been very effective in helping people to overcome information overload problem and they have become an integral part of e-business domain over the last few decades. RS provides relevant and personalized item recommendations to the users, thereby improving user satisfaction and item sales of e-commerce companies. It has a significant footprint in diversified applications including music, movies, apparel, tourism destinations, hotels, online courses, books, friends, electronic gadgets, recipes, doctors, *etc.* Most of the RS in e-commerce domain work as follow. RS predicts the rating of a target item and it recommends the item to the active user if the predicted rating exceeds a certain threshold. The algorithms that perform the task of predicting the ratings and providing personalized recommendations are broadly classified into two types: *content based filtering* and *collaborative filtering* (CF). Content based filtering techniques provide recommendations by analyzing the active user's content and features, profile of the item and profiles of items s/he preferred in past (Lang, 1995; Pazzani & Billsus, 1997). However, content based filtering approaches being solely reliant on item and user profiles, these approaches lack diversity in provided recommendations (Desrosiers & Karypis, 2011). On the other hand, CF based approaches are successfully deployed in many e-commerce domains (*e.g.* Amazon.com Linden, Smith, & York, 2003). CF approaches are categorized into *neighborhood based CF* and *model based CF*. Neighborhood based CF approaches compute the users' (items') neighbors based on the ratings provided (received) in the past. These techniques work on a simple intuition that an item might be preferred by an active user if the item is liked by a set of similar users (neighbors) or she has liked similar items (Resnick, Iacovou, Suchak, Bergstrom, & Riedl, 1994), (Sarwar, Karypis, Konstan, & Riedl, 2001). Though, neighborhood based CF approaches

* Corresponding author.
E-mail addresses: 515cs1002@nitrkl.ac.in (R.S. Sreepada), patrabk@nitrkl.ac.in (B.Kr. Patra).

https://doi.org/10.1016/j.eswa.2019.112887
0957-4174/

are simple, intuitive and efficient, model based approaches are proven to be more accurate (Koren, Bell, & Volinsky, 2009). Model based techniques build a model using machine learning, evolutionary computation or artificial intelligence techniques by navigating through the rating space (Koren et al., 2009; Luo, Xia, & Zhu, 2012; Yager, 2003). However, due to unavailability of ratings of new users (or items), the traditional CF approaches do not perform well in user or item cold-start scenarios. To overcome the shortcomings of content based and CF based approaches, hybrid approaches are also proposed (Kim, Ji, Ha, & Jo, 2010; Leung, Chan, & Chung, 2008; Strub, Gaudel, & Mary, 2016).

The above mentioned methods are biased towards recommending highly rated and most popular items (Park & Tuzhilin, 2008). However, significant number of the items receive less number of ratings. In this context, the Pareto principle suggested (80 - 20 rule) by economist Joseph M. Juran can be reminisced. As the rule says, 80% of the items obtain only 20% of the total ratings and 20% of the items obtain 80% of the ratings. Due to significant number of ratings on the 20% of the items, these items end up being highly recommended items. The left out items might receive a few good ratings, but are generally dominated by the highly popular items. Therefore, the left out items form a *long tail* in the product space. Predicting the rating of long tail items is a highly challenging task, but the vendors incur a huge loss if the items are not consumed by the users. Also, the users might be missing out on interesting items. Therefore, it is very necessary to recommend relevant long tail items to the appropriate users.

In this paper, we propose a framework to mitigate the long tail effect in recommender systems. We effectively adopt few shot learning technique called *siamese networks* (Bromley, Guyon, LeCun, Säckinger, & Shah, 1994) to recommend long tail items to the appropriate users. In the proposed framework, the task of recommendations is performed in three phases. In the first phase, we utilize One Class Collaborative Filtering (OCCF) approach to obtain the pre-use preferences of each user over all the items (Pan et al., 2008). In the second phase, we group users into a number of clusters and discover three categories of items from each cluster. These item statistics and pre-use preferences obtained from OCCF are used to train the siamese network. Finally, trained siamese network is used to identify the relevant users to whom long tail and head items are recommended. The main contributions of this paper are summarized below.

- The task of recommending items to the users is modeled as a classification problem and this is addressed using a few shot learning technique in this paper.
- The ‘pre-use’ preferences are predicted intelligently using OCCF approach. These preferences are provided as part of training input to the siamese network.
- A “hierarchy based clustering approach” is utilized to group the similar users. A similarity approach is proposed to obtain three categories of items from each user cluster: ‘liked items’, ‘long tail items’ and ‘disliked items’. This information is fed to siamese network.
- To overcome the limited ratings problem, siamese networks are employed to learn the interests of each user cluster. The learnt models are further used to identify appropriate users to recommend each long tail item.
- To evaluate the performance of recommendations on long tail items, three novel performance evaluation metrics are introduced in this article.
- Two real-world datasets (MovieLens 1M and Netflix) are used to validate the proposed approach on head items and long tail items. Experimental results show the effectiveness of proposed approach over existing long tail item recommender systems and traditional collaborative filtering approaches.

Rest of the paper is organized as follows. In Section 2, we provide the related work and relevant background of the proposed framework. Proposed framework is presented in detail in Section 3. Section 4 gives experimental setup and evaluation results. Finally, we conclude our work in Section 5.

2. Related work and background

This section has two subsections. In the first subsection, we discuss the works which address the long tail problem in recommendations. In the second subsection, we describe two important approaches namely OCCF and siamese network, which are adapted in our paper.

2.1. Related work

The term ‘long tail’ in the context of e-commerce domain is first coined by Anderson (2006). Anderson describes that an item belongs to long tail if it receives a few ratings or if it is consumed by less number of users. Anderson observed that long tail items are niches in most of the cases. However, the niches are not useful until they are visible in terms of recommendations to the users. Therefore, recommender systems may play an important role in identifying niche items in the long tail and recommending them to the appropriate users. In the last few decades, a few researchers attempted to diminish the effect of long tail in recommender systems.

Park and Tuzhilin (2008) proposed an approach to split the items into head and tail. For each item, a few features such as average rating, popularity rating, liked rating, total number of ratings received *etc.* are utilized to cluster the tail items using expectation maximization technique. The ratings prevalent in the obtained clusters are used to obtain the relevant users to whom the items can be recommended. However, this approach is completely reliant on the derived feature set, which is not exhaustive to perform clustering on the items.

Steck (2011) examined the trade-off between item popularity and recommendation accuracy. This approach assumes a bias towards the popular items and the recall measure is adjusted to reduce the bias. An evaluation metric called Popularity-stratified recall, which emphasizes more on the successful prediction and recommendation of long tail items is proposed. This measure gives weightage to unpopular items and penalizes the recall of popular items. Yin, Cui, Li, Yao, and Chen (2012) represented the user-item rating information in the form of a weighted graph. This weighted graph is used to apply the Hitting time algorithm to recommend the long tail items. Further, absorbing time and entropy-LDA based approaches are proposed, which are extension to hitting time algorithm.

Shi (2013) proposed an unbiased approach which recommends head and tail items. This approach balances accuracy, diversity and long tail items while providing recommendations. A graph based ‘cost flow’ concept is proposed which computes and recommends lower cost items to the users. Subsequently, an orthogonal-sparse-orthogonal non-negative matrix tri-factorization technique is proposed to recommend long tail items. Valcarce et al. utilized relevance based modeling to recommend long tail items to the users in Valcarce, Parapar, and Barreiro (2016). Out of the items present, least rated, lowest rated and least recommended products are identified as items to be liquidated. Relevance model on each of these items for each user is built through a probabilistic approach. In Wang, Gong, Li, and Yang (2016), Wang et al. proposed a genetic algorithm based multi-objective optimization approach to recommend accurate and novel items. Two fitness functions are defined. The first fitness function computes the summation of predicted rating values. The second fitness function computes the sum