Adaptive Collaborative Filtering for Recommender System

An La(✉), Phuong Vo(✉), and Tu Vu

Big Data Department, FPT Telecom Vietnam, Ho Chi Minh City, Vietnam
{AnLNT2,PhuongVTH,TuVA}@fpt.com.vn

Abstract. On online websites or e-commerce services, the explosive growth of resource makes the problem of content exploring increasingly challenging. The recommender system is a powerful information filtering tool to support user interaction and promote products. Dealing with determining customer interests, graph-based collaborative filtering is recently the most popular technique. Its only drawback is high computing cost, leads to bad scalability and infeasibility for large size network. Moreover, most previous studies concentrate solely on the accuracy of user preference prediction, while the efficiency of recommendation methods should be considered in many characteristics with complicated relationships, depending on particular systems: popularity, diversity, coverage, congestion. Attempt to conquer these challenges, we propose Adaptive Weighted Conduction formula handling multiple metrics, then construct a scalable model with small complexity, named Adaptive Collaborative Filtering. Experiments are conducted on Movielens, a public dataset, and FPT PLAY, a dataset of our media service. We have an increase of 6% on precision and get close to the best of previous methods on diversity, coverage and congestion. This result shows that the proposed model automatically reveals and adapts to varied requirements of recommender systems, reaches high accuracy and still balances other evaluation aspects.

Keywords: E-commerce service · Recommender system · Graph-based collaborative filtering · Scalability · Evaluation metrics

1 Introduction

Half of the world population is now online [11] and e-commerce retail continues to grow at accelerated rate [1]. Accordingly, the most essential question of both research and business is how to improve the interaction of customers and bring suitable products to them. Since the rapid increase in the number of products, this issue is more awkward. Users are overwhelmed by a vast amount of options and often expect a systematical process that helps them in making decisions. Supporting users to cope with content overloading is the key challenge of information filtering techniques. While other tools like search engine typically require

D. Endres et al. (Eds.): ICCS 2019, LNAI 11530, pp. 117–130, 2019.
https://doi.org/10.1007/978-3-030-23182-8_9

users to determine in advance what they need, a recommender system proactively gives various suggestions based on currently watching or historical items. It also dynamically controls product display, plays a vital role in the activities of online services.

Many recommendation algorithms have been proposed [13]. Mining what user selected in the past, collaborative filtering is the most ubiquitous technique [21]. Without the need of explicit item information, it automatically figures out implicit relations between objects (users or items) and captures up-to-date interests of customers. For visual representation and analysis, recent studies build Bipartite Network simulating user history then make recommendation on this structure, forming graph-based approach [4,6,20,22]. However, comparing all pairs of users has expensive computing cost, while user profiles usually update very fast and the entire system has to be rebuilt. Speed up can be attained by pre-constructing graph, but with some applications, large numbers of users and items make them impractical and badly scalable.

Another factor for the success of a recommender system is evaluation: how do we know suggestion is good or not? Customer interests change quickly, especially on systems with heated activity. The intervention of recommendation, affecting to vision and emotion of users, causes the alteration also. Thus, it is difficult to precisely foretell their next choices. Moreover, concentrating solely on accuracy may not make realistic effects. Being accurate is sometimes not useful to users, even hurts the system [17]. In addition, to validate whether items which are incorrect in the testing set could attract users, offline evaluating is insufficient. Meanwhile, with the ability of dynamical controlling visibility of products, a few studies solve other characteristics of e-commerce services, such as item coverage [14,20], diversity and novelty [15,16], diversity-accuracy dilemma [2,3,19]. Therefore, not purely precision, a recommendation algorithm should be considered in multiple properties, reflected by corresponding metrics. However, the correlation between them is complicated and their priorities are inconsistent, depending on a specific system. It is necessary to have a general way that allows us to understand the user-item distribution and foresee the importance of these properties.

In making the effort of overcoming these limitations, we propose Adaptive Weighted Conduction (AWC), a hybrid formula of existing studies dealing problems of evaluation, and figure out how our model can adapt with varied requirements on recommendation properties. We also remedy the bad scalability for the class of methods using graph. For multi-purpose adaptability, our proposal is called Adaptive Collaborative Filtering (ACF).

The rest of this paper is organized as follows. In Sect. 2, we discuss related works solving evaluation problems and current solutions of scalability issue. Detail of our proposal and proof complexity of ACF are described in Sect. 3. Section 4 includes description of evaluation metrics, experimental results and analysis on Movielens and FPT Play in order to illustrate the ability of AWC. Finally, Sect. 5 presents our conclusion.

2 Related Work

Evaluation is the essential challenge demonstrating the performance of recommendation methods. Besides accuracy, multiple properties in complicated relationships need to be considered. A few studies achieved in directly using item characteristics to control all metrics, although each of them has its own disadvantages. ProbS [19] mainly outputs trending products, which is a safe suggestion for most customers, has a limited visible product size and lack of diversity. HeatS [19] assumes that new items are selected by just a small number of users, prioritizes unpopular items to grow novelty and personalization capability. Hybrid of ProbS and HeatS [19] is a successful method in balancing between accuracy and diversity. DWC [20] raises and solves congestion problem in recommender systems, also reaches remarkable performance on both diversity and coverage. Nevertheless, it strictly gives more priority to less popular items as HeatS, which makes their precision worse.

Another common drawback of four methods is high complexity. In general, it is the limitation of almost graph-based collaborative filtering methods, caused by constructing Bipartite Network and matching all pairs of users [6]. Besides that, inspired by "people who buy x also buy y", item-based approach calculates similarities between items instead of users to overcome both data sparsity and scalability problem [10]. Utilizing this advantage, a few previous works build a graph representing item-item matrix [4,5,7,12,18], unfold an approach which has several strengths: fast computational time [7], ability to capitalize on graphical meaning and outperform standard item-based methods [12,18].

In this paper, we propose a hybrid of ProbS, HeatS, the Hybrid and DWC in order to exploit their strategies dealing with evaluation challenge, called Adaptive Weighted Conduction (AWC). After that, ACF is formed by presenting AWC as an item-item network to make sure high scalability. The detail of our proposal is described in the next section.

3 The Model

3.1 Previous Methods

Bipartite Network provides a clear way to visualize activities of users. It is denoted as $G(U, I, E)$, where U, I is respectively set of users and set of items, E includes connections of one vertex in U to one vertex in I. User set has size $m = |U|$ and item set has size $n = |I|$. We use Greek letters (α, β) to indicate items and italic letters (i, j) to indicate users.

Let the adjacency matrix $A_{m\times n}$ describe E. If a user i selected item α, (i, α) is an edge in E and $a_{i\alpha} = 1$, $a_{i\alpha} = 0$ otherwise. The degree of item α is the total number of users selected it, denoted as k_α. Similarly, the degree of user i is history length of user i, denoted as k_i. For each user i, ProbS, HeatS, the Hybrid and DWC process two steps below:

Step 1 - Diffusion from user side to item side: Calculate relation score for every neighbor j to user i, denoted by g_{ij}.

Step 2 - Diffusion from item side to user side: Compute preference score of user i for item α, denoted by $f_{i\alpha}$.

Following these steps, each method has a specific formula.

Hybrid of ProbS and HeatS (the Hybrid). Solving diversity-accuracy dilemma, [19] proposed a hybrid of ProbS and HeatS:

$$g_{ij} = \sum_{\alpha} \frac{a_{i\alpha} a_{j\alpha}}{k_\alpha^\lambda k_j^{1-\lambda}} \tag{1}$$

$$f_{i\alpha} = \sum_{j} \frac{a_{j\alpha} g_j}{k_\alpha^{1-\lambda} k_j^\lambda} \tag{2}$$

where λ is tunable parameter in range $[0,1]$. When $\lambda = 1$, it becomes ProbS that mostly recommends trending products. When $\lambda = 0$, the Eqs. (1) and (2) turn to be HeatS and just outcomes unpopular items.

In addition, [19] rewrote these formulas as dot product: $f_i = W a_i$, where a_i and f_i is respectively a vector denoting historical and recommended items of user i. The combination of ProbS and HeatS is archived in a transition matrix $W_{n \times n}$:

$$w_{\alpha\beta} = \frac{1}{k_\alpha^{1-\lambda} k_\beta^\lambda} \sum_{j=1}^{m} \frac{a_{j\alpha} a_{j\beta}}{k_j} \tag{3}$$

Directed Weighted Conduction (DWC). [20] keeps $f_{i\alpha}$ as HeatS and just modifies how to match two people:

$$g_{ij} = \frac{\sum_\alpha h_{j\alpha} a_{j\alpha} a_{i\alpha}}{\sum_\alpha h_{j\alpha} a_{j\alpha}} \tag{4}$$

where $h_{j\alpha} = (k_j k_\alpha)^\gamma$. By experiment on particular datasets, authors suppose not to use $\gamma > 0$.

3.2 Adaptive Weighted Conduction (AWC)

Despite different strategies, previous studies use the same mechanism of diffusion on Bipartite Graph. The node degree has important information, uncovers characteristics of objects, such as how active a user is or popularity of an item. Both ProbS, HeatS, the Hybrid and DWC exploit this value in matching tastes of people and recommending items for users. Since the relation between popularity and diversity, coverage, congestion, they also succeed in considering product display. However, confusing that novelty is small popularity, they strive to minimize popularity to enhance the quality of suggestions. We suppose that only counting number is inadequate to discriminate between new items and items which have unconcern of users. Generally, relying on a specific system, the homology between novelty and popularity is uncertain. More product information, such as rating or release date, should be examined. In this paper, this metric is called popularity. Instead of optimizing it, popularity becomes an indicator for all product display aspects due to its relation to others. Rather than investigating each metric individually or facing their dilemma relationships, our target is to acquire higher precision and balance all display metrics at the same time.

We process two diffusion steps and drive degrees of nodes as previous methods, but aim to bring precisely both popular and unpopular items to particular users.

At the first diffusion step, identifying how similar between people is a difficult mission. Addressing this challenge, DWC changes weights of common items according to their counting values rather than treats them equally as the Hybrid. We generalize the idea of DWC and say that the node degree reflects the role of an item in determining user interests, not exclusively in comparing tastes of customers. This comparable information is expressed by:

$$t_{j\alpha} = \frac{a_{j\alpha} k_{\alpha}^{\gamma}}{\sum_{\beta} a_{j\beta} k_{\beta}^{\gamma}} \tag{5}$$

When $\gamma > 0$, the weight increases as the degree goes up. That means popular items contribute more to the relation score of two users than unpopular items. On the contrary, when $\gamma < 0$, the weight increases as the degree goes down, unpopular items have a higher contribution. In practice, not widely-known items demonstrate the stronger relationship between users so that we suggest not using $\gamma > 0$, which is the same as the conclusion by experiments of DWC.

The second step of each method is harnessing counting values to go straight to the target. ProbS primarily gives trending products for accuracy. HeatS concentrates on unpopular items for novelty and diversity. Directly combining two opposite strategies, the Hybrid balances them by a tunable parameter λ, solves the diversity-accuracy dilemma. We leverage the second step of the Hybrid, keep $f_{i\alpha}$ as Eq. (2), so that our model could balance accuracy and diversity, but use the idea of DWC when matching user interest in order to increase its accuracy. The relation score between users of AWC is:

$$g_{ij} = \sum_{\alpha} \frac{a_{i\alpha} a_{j\alpha} t_{j\alpha}}{k_{\alpha}^{\lambda} k_{j}^{1-\lambda}} \tag{6}$$

Replace g_{ij} in (2) by (6), expansion of preference score in AWC:

$$f_{i\alpha} = \sum_{j} \frac{a_{j\alpha}}{k_{\alpha}^{1-\lambda}} \sum_{\beta} \frac{a_{i\beta} a_{j\beta} t_{j\beta}}{k_{\beta}^{\lambda}} \tag{7}$$

When $\gamma = 0$, AWC is the Hybrid with λ is the tunable parameter. Rewriting $f_{i\alpha}$ of the Hybrid by expanding g_{ij} from (1), we have:

$$f_{i\alpha} = \sum_{j} \frac{a_{j\alpha} g_{ij}}{k_{\alpha}^{1-\lambda} k_{j}^{\lambda}} = \sum_{j} \frac{a_{j\alpha}}{k_{\alpha}^{1-\lambda}} \sum_{\beta} \frac{a_{i\beta} a_{j\beta}}{k_{\beta}^{\lambda} k_{j}} \tag{8}$$

In case of $\gamma = 0$, $t_{j\beta} = \dfrac{1}{k_j}$, so that (8) is equal to (7).

When $\lambda = 0$, AWC is DWC with γ is the tunable parameter. Express g_{ij} of DWC in Eq. (4) by $t_{j\alpha}$ in Eq. (5), we obtain:

$$g_{ij} = \sum_{\alpha} a_{i\alpha} a_{j\alpha} t_{j\alpha} \tag{9}$$

Replacing g_{ij} in $f_{i\alpha}$ by (9), expansion of DWC:

$$f_{i\alpha} = \sum_{j} \frac{a_{j\alpha} g_{ij}}{k_\alpha} = \sum_{j} \frac{a_{j\alpha}}{k_\alpha} \sum_{\beta} a_{i\beta} a_{j\beta} t_{j\beta} \tag{10}$$

Obviously, in case $\lambda = 0$, Eq. (10) is equal to (7).

Using dual parameters in the first step only unfolds the space for exploring the problem of capturing user interests. Although it is able to increase the accuracy, reaching the best performance on all evaluation metrics is the other challenge. We still keep balancing state between product display aspects when getting the possible highest precision. Due to inheriting Hybrid of ProbS and HeatS, solving the accuracy-diversity dilemma, AWC could remedy the low diversity while optimizing precision, which means the model found a global or local optimal point that efficiently recommends both popular and unpopular items to users correctly. Naturally, using a wide range of popularity, coverage is enlarged, and the congestion problem, caused by suggesting solely a small set of items, is reduced.

Therefore, bringing precisely both popular and unpopular items to particular users subsequently balance all other metrics.

3.3 Adaptive Collaborative Filtering (ACF)

Both ProbS, HeatS, the Hybrid and DWC go through two steps of diffusion, each step has to consider all pairs of users and all common items of them, which has high computing cost. Meanwhile, item-based methods calculate similarities between items instead of users, extremely reduce the complexity. We embed AWC formula into an item-item matrix, then construct an item graph to speed up our method.

Item-Based Recommendation. [19] rewrote two-step diffusion as $f_i = W a_i$, with $W_{n\times n}$ is a hybridization matrix of ProbS and HeatS. Inspired by this idea, from (7) we extract the hybridization matrix for AWC:

$$w_{\alpha\beta} = \frac{1}{k_\alpha^{1-\lambda} k_\beta^{\lambda}} \sum_{j} a_{j\alpha} a_{j\beta} t_{j\beta} \tag{11}$$

The recommended matrix $F_{m\times n}$ is result of multiplication between history matrix $A_{m\times n}$ and item-item matrix $W_{n\times n}$: $F = AW^\top$.

Item-Graph Recommendation. Let call $G^I(I, E^I)$ is a Weighted Directed Graph expressing $W_{n\times n}$, where I is item set and E^I is all available connections between items. Because $W_{n\times n}$ is not symmetric, (α, β) and (β, α) has different weights. In detail, if (β, α) is an edge in E, its weight is equal to $w_{\alpha\beta}$. Any pair of items with zero-weight in $W_{n\times n}$ does not have any edge in E.

When two items α and β both appear in the history of a user j, they are connected to each other, two edges (α, β) and (β, α) are established. For more convenient in computing, let call:

$$s_j = \sum_{\alpha} a_{j\alpha} k_\alpha^\gamma \tag{12}$$

so that $t_{j\beta} = \dfrac{a_{j\beta} k_\beta^\gamma}{s_j}$ and Eq. (11) becomes:

$$w_{\alpha\beta} = k_\alpha^{\lambda-1} k_\beta^{\gamma-\lambda} \sum_j \frac{a_{j\alpha} a_{j\beta}}{s_j} \tag{13}$$

Obviously, for every user j selected both α and β, weight of edge (β, α) is computed by:

$$\sum_j \frac{k_\alpha^{\lambda-1} k_\beta^{\gamma-\lambda}}{s_j} \tag{14}$$

Two steps to construct the $G^I(I, E^I)$:

Step 1 - Prepare features of users and items: Read all user history and calculate k_α for every items, k_i and s_i for every user i.

Step 2 - Establish edges of item graph: Visit history of all users again. For every pair of α and β both appear in the history of user j, connect them in $G^I(I, E^I)$ with weight computed by Eq. (14).

To recommend for user i on graph, preference score of item α is equal to sum of all edge weights from historical item β of i to α: $f_{i\alpha} = \sum_\beta w_{\alpha\beta} a_{i\beta}$.

Complexity. Let d is the number of ratings or user selections in history, or $d = \sum_m k_i$. Let s reflects sparsity of graph, is calculated by $\dfrac{d}{m * n}$, which means the portion of the actual edges in maximum possible edges. Averaging the number of items selected by a user, denoted by K, is equal to $\overline{k_i} = \dfrac{d}{m} = s * n$. It is clearly to see that $K \leq n$. $K = n$ occurs only when the graph is fully connected, or every item connects to each other. As usual, because of data sparsity, $K \ll n$.

When implementing the pure formula of the Hybrid or DWC, we visit all pairs of users and match them by all items, spend $\mathcal{O}(m^2 n)$ to compute g and f. Using the definition of Bipartite Network and just considering items in common, the computational cost is reduced to $\mathcal{O}(m^2 K)$.

Obviously, item-based version of ACF has $\mathcal{O}(mK^2)$ to measure item-item matrix. With a simple matrix multiplication algorithm, it takes $\mathcal{O}(mn^2)$ for

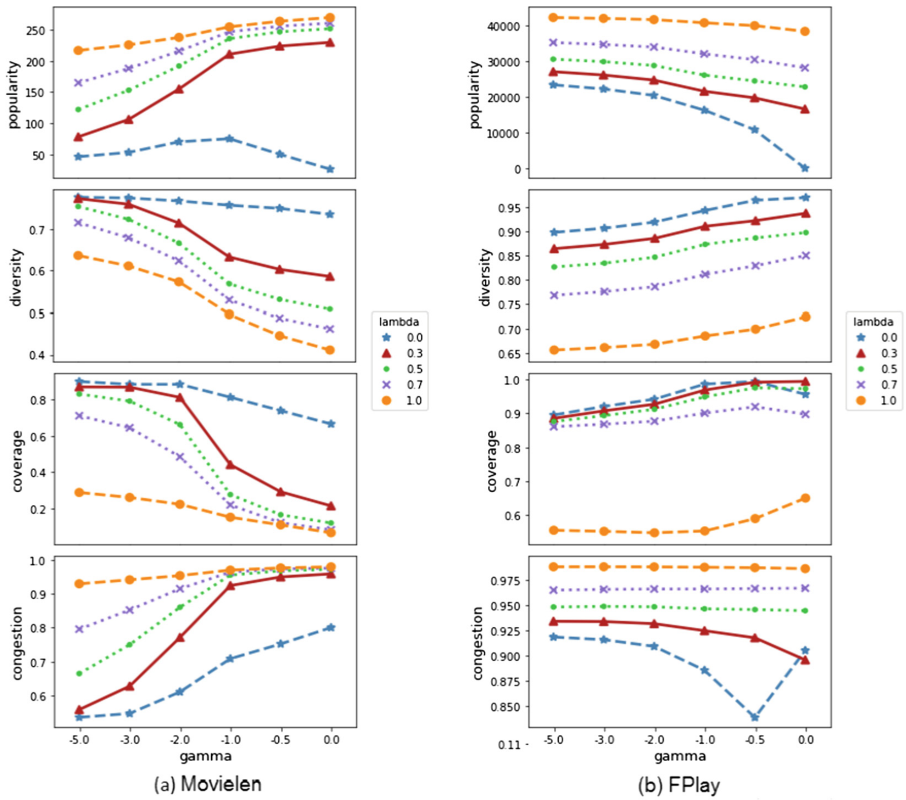

Fig. 1. The change of four properties on testing data of Movielens (left) and FPT Play (right). To visualize, we select the same sets of parameters for all datasets, in which $\lambda = 0.3$ is the optimal line (solid line), subjecting to the precision. Via γ (horizontal axis) and the fixed order of λ, the decrease of popularity causes better tendencies other properties.

making recommendation. In comparison to using Bipartite Network, this way is slower only when the number of users is less than the number of items, not a popular case in recommender systems.

With the item-graphversion of ACF, to construct the graph, the first step requires $\mathcal{O}(mK)$ to read all user history, the second step takes $\mathcal{O}(mK^2)$ to generate all pairs of items within the history of each user. Suppose that each historical item β connects to T recommended items α on average, making recommendation for all users needs $\mathcal{O}(mTK)$. Therefore, the complexity is $\mathcal{O}(mK^2)$ to pre-construct the network and $\mathcal{O}(mTK)$ in total. In case of fully connected graph, $T = n$, otherwise $T < n$, so that item-net model has extremely reduced complexity.

Table 1. Statistical properties of Movielens and FPT Play dataset. Sparsity is the rate of connections between nodes on graph. FPT Play has larger numbers of users and items and a sparser network.

Movielens	Users	Items	Links	Sparsity
Total	943	1,682	100,000	6.82×10^{-2}
Train	617	1,515	62,414	6.68×10^{-2}
Test	134	1,040	4,910	
FPT Play	Users	Items	Links	Sparsity
Total	1,377,579	19,371	13,741,845	0.05×10^{-2}
Train	1,252,058	18,615	10,757,016	0.05×10^{-2}
Test	476,385	14,774	2,445,341	

Table 2. Performances of different methods on testing data of Movielens (top) and FPT Play (bottom). For parameter-dependent algorithms, the results are obtained by the optimal set of parameters, which is label in brackets. In AWC, the first one is λ, the next is γ. We do not judge methods by popularity directly. For remaining metrics, except the congestion, the higher is the better. The first and second best results are emphasized.

Movielens	Popularity	Diversity	Coverage	Congestion	Precision	Rating
ProbS	270	0.4112	0.0695	0.9808	0.0884	3.0385
HeatS	26	0.7351	0.6642	0.8005	0.0470	1.7143
Hybrid (0.2)	202	0.6483	0.3284	0.9366	0.0821	**3.1739**
DWC (−5.0)	46	**0.7762**	**0.8960**	**0.5340**	**0.0981**	2.8675
AWC (0.3, −5.0)	78	**0.7730**	**0.8665**	**0.5569**	**0.1037**	**3.1605**
FPT Play	Popularity	Diversity	Coverage	Congestion	Precision	
ProbS	38481	0.7243	0.6507	0.9860	**0.1021**	
HeatS	59	**0.9689**	0.9557	0.9050	0.0709	
Hybrid (0.1)	3234	**0.9769**	0.9877	**0.8676**	0.0967	
DWC (−0.5)	10868	0.9634	**0.9936**	**0.8386**	0.0978	
AWC (0.3, −0.5)	19811	0.9212	**0.9915**	0.9176	**0.1078**	

Table 3. The improvement of AWC in comparison to other methods. It is computed by $\frac{r_1 - r_0}{r_0}$, where r_0, r_1 is respectively value of AWC and the best of previous method targeting the precision. Except for the congestion, the more positive is the better. Outperformed values of AWC are emphasized.

	Diversity	Coverage	Congestion	Precision	Rating
Movielens	−0.41%	−3.29%	4.29%	**5.70%**	**10.22%**
FPT Play	**27.19%**	**52.37%**	**−6.94%**	**5.58%**	

4 Experiments and Results

In this section, we focus only on demonstrating the ability of AWC in dealing with multiple evaluation metrics. We separate each dataset into two parts: training and testing. The graph is constructed from training data, then predicts L items for each user and compares to testing data. We experiment with many values of L and take $L = 20$ for example and visualization. Characteristics of an algorithm are evaluated based on its outcome through four metrics: popularity, diversity, coverage and congestion. Comparing lists of L recommended items and testing data figures out the precision. Rating is considered on Movielens in order to check the quality of results. Details of datasets, evaluation metrics and results are presented below.

4.1 Datasets and Evaluation Metrics

Datasets. Statistical properties of datasets are described in Table 1.
Movielens. A movie rating dataset of the GroupLens project at the University of Minnesota [8]. It can be downloaded from www.grouplens.org. Each row includes the id of a user, the id of an item and rating on range $[1, 5]$. Training and testing part are extracted with ratio 0.95 : 0.05, then we filter the testing by removing ratings which have strange user or item to the training.

FPT PLAY. This is taken from history of users on fptplay.vn [9], our online service that allows customers to watch a wide variety of movies, TV shows and more. We use 40-day history, where the testing part is consists of the last 10-day activities of all users on the system and applied the filtering as on Movielens testing part.

Evaluation Metrics
Popularity. This reflects how popular items are on result lists. In DWC, assuming that new items have low counting values, this metric is called as novelty and minimized by algorithms. We suppose that low popularity and novelty are not the same and do not optimize this value. To calculate, for a target user i and list of L recommended items α, his or her popularity is: $N_i(L) = \frac{1}{L}\sum_\alpha k_\alpha$. The popularity of an algorithm is taking average N_i of all users.

Diversity. Diversity shows how different the suggesting outcomes are between users. The higher value, the more personalized. It is also related to popularity and accuracy. Bringing mostly popular items is a safe strategy for fitting customer interest, but leads to low difference between users. In contrast, using more unpopular products enhances personalization capability but has a high risk of satisfying customers.

Let call D_{ij} is the number of distinct objects between results of user i and j. Diversity is computed by averaging D_{ij} of all pairs of users.

Coverage. This metric demonstrates the efficiency of resource usage by a recommendation algorithm. It is equal to the rate of items which are recommended to at least one user on the total resource (Table 4).

Table 4. Statistical properties of Item Graphs

	Nodes	Edges	Sparsity
Movielens	1,515	1,414,988	0.61
FPT Play	18,615	35,264,316	0.10

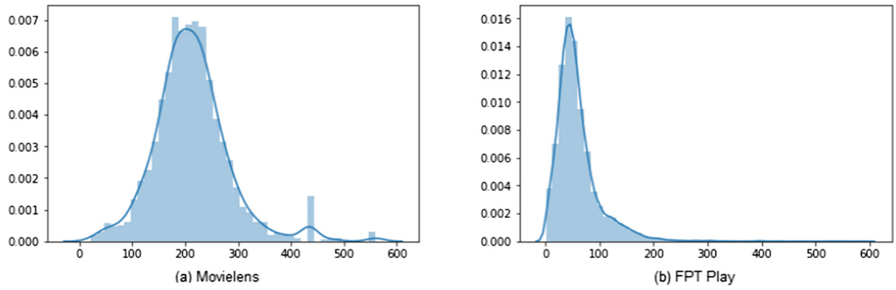

Fig. 2. Distribution of items by averaging degrees their users on Movielens (top) and FPT Play (bottom). On Movielens, the shape is normal distribution, while on FPT Play is extremely right skewed.

Congestion. [20] introduced the congestion problem and proposed this metric. Congestion occurs when a few distinct items are in suggesting lists of numerous users. Therefore, the lower is the better. It can be seen as hard version of coverage. Coverage just checks whether an item is in result of any user, while congestion compares the number of times it is recommended.

Precision. Precision is one of accuracy metrics, which are always the first priority. If a method has bad accuracy, no need to consider other properties. It is taken by counting the number of items that are correctly given to a user, then getting average of these values of all users.

Rating. Rating is a score pointing out how users are satisfied with content of items. In this paper, together with precision, rating is another accuracy metric, while other methods use it as a feature. Our purpose is to validate whether the model exploits low-degree items but distinguishes between novelty and bad quality. It is measured by averaging scores of all items recommended for each user first, then taking average these values of all users.

4.2 Results

Complicated relationships between properties of recommendation come from practical requirements. To balance all product display aspects, we use a wide range of objects, from trending to unpopular. Figure 1 visualizes this ability of AWC. Regardless of the change of duel parameters γ and λ, relationships of metrics are always a unified pattern: diversity, coverage, and congestion are followed by popularity. By γ, the popularity decreases and the other properties tend to be better: diversity, coverage goes up and congestion falls. Following λ, the order of lines stays the same on every characteristic: the lower λ, the lower popularity and congestion, the higher diversity and coverage. Due to this pattern, AWC drives popularity and thus indirectly manages other metrics. Therefore, it allows us to understand and control all evaluation aspects of recommender systems.

Discovering a dimension which has boundaries $\gamma = 0$ (the Hybrid) and $\lambda = 0$ (DWC), AWC unfolds combinations of strategies to acquire better performance. All values are presented in Table 2, where results of three parameter-dependent algorithms (Hybrid, DWC, AWC) are obtained by the optimal set of parameters targeting accuracy. Overall, AWC has the highest precision, goes up by $\approx 6\%$ compared to previous methods, from 0.0981 to 0.1037 on Movielens and from 0.1021 to 0.1087 on FPT Play. It also overcomes disadvantages of different strategies on other metrics. On Movielens dataset, together with DWC, it exceeds far away from the best of ProbS, HeatS and the Hybrid: coverage enlarges from 0.6642 to 0.8665, congestion declines from 0.8005 to 0.5569, each metric grows $\approx 30\%$; diversity has a soft increase, from 0.7351 to around 0.77. Like DWC, small popularity means recommending mostly unpopular items, but the rating of our model is 10.22% higher and very close to the value of the Hybrid, which takes the first rank. This contributes to illustrating the insufficiency of elevating novelty by minimizing popularity; simultaneously, showing the capability of AWC to excommunicate low-quality items. We conclude that on Movielens, AWC has the greatest precision and approximates to the best values on other metrics. On FPT Play, within previous methods, focusing on popular items but still holding 65% coverage and 0.7243 diversity, ProbS is the most suitable strategy. That means users should be recommended by trending products, fitting their interests by unpopular items is precarious. Obviously, the Hybrid and DWC compromise their accuracy. Nevertheless, our proposal succeeds in selecting more unpopular objects. Not exclusively rise the precision, AWC exceeds ProbS and reaches close to other methods on remaining metrics. Diversity extends 27%, from 0.7243 to 0.9212. Coverage dramatically spreads 52%, from 0.6507 to 0.9915 and very close the best value, which is 0.9936 of DWC. Congestion has a 7% reduction, from 0.9860 to 0.9176. Therefore, AWC has a dramatic growth from ProbS and a competitive performance on FPT Play dataset.

To clarify the distinction between FPT Play and Movielens that leads to the difference in strategies applied, we present simple analysis of two graphs. Averaging length of user history, Movielens is around 100 and FPT Play approximates 10. Figure 2 shows that while Movielens has a normal user-item distribution, almost FPT Play items are selected by low-degree users. A too short history and a large number of items cause sparsity of graph: a few edges are established with low reliability, items have a small number of connections, especially unpopular ones. In this circumstance, recommending not widely-known products has a very high risk. This explicates the poor precision of HeatS, the Hybrid and DWC on FPT Play. Meanwhile, Movielens faces another problem. Regardless dense or sparse graph, trending objects attract more connections, dominate other products. The repeat of recommending mostly popular items is the root of the congestion problem. Solving this challenge, DWC outperforms previous methods. Instead of examining user-item distribution likes this, tuning parameters of AWC reveals and adapts to characteristics of two graph types, reaches better performance on both datasets.

5 Conclusion

The recommender system is a powerful information filtering technique which has two important roles: bring suitable products to user and control product display. User satisfaction is difficult to be offline evaluated and focusing solely on the accuracy of predicting user selection may not make realistic effects. Meanwhile, item visibility can be judged directly via diversity, coverage and congestion. ProbS, HeatS, the Hybrid and DWC have their own strategies and targeting metrics when dealing with these metrics. We propose a hybrid formula, called Adaptive Weighted Conduction (AWC), combining strengths of those methods for better capturing user interests. Improving the Hybrid by similarity computing of DWC, our model is able to balance all remaining metrics. We also overcome the issue of high computing cost, which leads to bad scalability, by forming Adaptive Collaborative Filtering (ACF) model from AWC. Experiments on Movielens and FPT Play show that our scalable model automatically reveals and adapts to varied requirements of recommender systems, not only outperforms on precision, but also has competitive results on other aspects.

When solving evaluation and scalability problems of recommender systems, this study provides three main contributions. Firstly, we present the insight into the evaluation: the difference between novelty and popularity. Instead of minimizing novelty or solving dilemma relationships of metrics as previous works, utilizing relation of popularity to remaining properties, we harness counting values to control all product display aspects. The second is the success of AWC formula in exploiting strengths of previous methods to acquire better precision and balance other metrics. Moreover, due to the combining of different strategies, AWC uncovers varied traits of particular systems when tunning dual parameters. The final contribution is the scalability of ACF model. It provides evidence for the power of item-graph based approach.

In conclusion, with multi-purpose adaptability, handling from evaluation challenge to scalability issue, our proposal is a practical model for recommender systems.

References

1. 2018 internet trends report. www.recode.net/2018/5/30/17385116. Accessed 13 Jan 2019
2. Gogna, A., Majumdar, A.: Balancing accuracy and diversity in recommendations using matrix completion framework. Knowl. Based Syst. **125**, 83–95 (2017)

3. Javari, A., Jalili, M.: A probabilistic model to resolve diversity-accuracy challenge of recommendation systems. Knowl. Inf. Syst. **44**(3), 609–627 (2015)
4. Putra, A.A., Mahendra, R., Budi, I., Munajat, Q.: Two-steps graph-based collaborative filtering using user and item similarities: case study of E-commerce recommender systems. In: International Conference on Data and Software Engineering, pp. 1–6. IEEE, Indonesia (2017)
5. Rostami, B., Cremonesi, P., Malucelli, F.: A graph optimization approach to item-based collaborative filtering. In: Fidanova, S. (ed.) Recent Advances in Computational Optimization. Studies in Computational Intelligence, vol. 470, pp. 15–30. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-319-00410-5_2
6. Shams, B., Haratizadeh, S.: Graph-based collaborative ranking. Expert. Syst. Appl. **67**, 59–70 (2017)
7. Wang, F., Ma, S., Yang, L., Li, T.: Recommendation on item graphs. In: ICDM 2006 the sixth International Conference, pp. 1119–1123. IEEE, Hong Kong (2006)
8. Harper, F.M., Konstan, J.A.: The movielens datasets: history and context. ACM Trans. Interact. Intell. Syst. (TIIS) **5**(4), 19 (2016)
9. FPT Play. http://fptplay.vn. Accessed 13 Jan 2019
10. Linden, G., Smith, B., York, J.: Amazon.com recommendations: item-to-item collaborative filtering. IEEE Internet Comput. **1**, 76–80 (2003)
11. Internet World Stats. https://www.internetworldstats.com/stats.htm. Accessed 13 Jan 2019
12. Item Graph Based Recommendation System for Bookopolis of Stanford Course. http://snap.stanford.edu/class/cs224w-2014/projects2014/cs224w-84-final.pdf. Accessed 13 Jan 2019
13. Bobadilla, J., Ortega, F., Hernando, A., Gutirrez, A.: Recommender systems survey. Knowl. Based Syst. **46**, 109–132 (2013)
14. Ge, M., Delgado-Battenfeld, C., Jannach, D.: Beyond accuracy: evaluating recommender systems by coverage and serendipity. In: The Fourth ACM Conference on Recommender Systems, pp. 257–260. ACM, New York (2010)
15. Castells, P., Vargas, S., Wang, J.: Novelty and diversity metrics for recommender systems: choice, discovery and relevance. In: International Workshop on Diversity in Document Retrieval (DDR). ACM, New York (2011)
16. Vargas, S., Castells, P.: Rank and relevance in novelty and diversity metrics for recommender systems. In: The Fifth ACM Conference on Recommender Systems, pp. 109–116. ACM, Chicago (2011)
17. McNee, S.M., Riedl, J., Konstan, J.A.: Being accurate is not enough: how accuracy metrics have hurt recommender systems. In: CHI 2006 Extended Abstracts on Human Factors in Computing Systems, pp. 1097–1101. ACM, Montral (2006)
18. Ha, T., Lee, S.: Item-network-based collaborative filtering: a personalized recommendation method based on a user's item network. Inf. Process. Manag. **53**(5), 1171–1184 (2017)
19. Zhou, T., Kuscsik, Z., Liu, J.G., Medo, M., Wakeling, J.R., Zhang, Y.C.: Solving the apparent diversity-accuracy dilemma of recommender systems. In: the National Academy of Sciences, USA, pp. 4511–4515 (2010)
20. Ren, X., Lü, L., Liu, R., Zhang, J.: Avoiding congestion in recommender systems. New J. Phys. **16**(6), 063057 (2016)
21. Yang, X., Guo, Y., Liu, Y., Steck, H.: A survey of collaborative filtering based social recommender systems. Comput. Commun. **41**, 1–10 (2014)
22. Xu, Y., Ma, J., Sun, Y., Hao, J., Sun, Y., Zhao, Y.: Using social network analysis as a strategy for e-commerce recommendation. In: The Thirteenth Pacific Asia Conference on Information Systems, pp. 106. AIS, Hyderabad (2009)