

Modeling Temporal Behavior of Awards Effect on Viewership of Movies

Basmah Altaf¹, Faisal Kamiran², and Xiangliang Zhang¹(✉)

¹ King Abdullah University of Science and Technology, Thuwal, Saudi Arabia
{basmah.altaf,xiangliang.zhang}@kaust.edu.sa

² Information Technology, University of the Punjab, Lahore, Pakistan
faisal.kamiran@itu.edu.pk

Abstract. The “rich get richer” effect is well-known in recommendation system. Popular items are recommended more, then purchased more, resulting in becoming even more popular over time. For example, we observe in Netflix data that awarded movies are more popular than non-awarded movies. Unlike other work focusing on making fair/neutralized recommendation, in this paper, we target on modeling the effect of awards on the viewership of movies. The main challenge of building such a model is that the effect on popularity changes over time with different intensity from movie to movie. Our proposed approach explicitly models the award effects for each movie and enables the recommendation system to provide a better ranked list of recommended movies. The results of an extensive empirical validation on Netflix and MovieLens data demonstrate the effectiveness of our model.

Keywords: Awards effect estimation · Popularity bias · Recommender systems

1 Introduction

Recommendation systems have been widely used in e-commerce to assist users in finding their potentially interested products [10], e.g., in Amazon, Ebay, Netflix, LinkedIn, YouTube, and IMDB. The ranked list of products suggested by recommendation systems can be based on the overall popularity (number of views/ratings/downloads) for the product, or based on the overall average rating, or based on user purchase history. Often, the ranked list of items in recommendation system is dominated by the most popular items [11] due to the “rich get richer” effect. Popular items are recommended more, thence purchased more, resulting in becoming even more popular over time. Though “rich get richer” is a well-known phenomenon, there has been little attention paid in studying the temporal behavior of such effects in popularity.

In this work, we model the varying effect of awards on the popularity of items over time in the context of movies domain¹. The most popular awards for

¹ The same framework can be applied to study award effect in other domains like songs (Grammy awards), scientific papers (best paper award), books (best seller book) and others.

movies include Oscar Award, Golden Globe Award, Bafta Award and Satellite Award. In general awards are considered a way to acknowledge the contribution of extraordinary movies to the industry. However many critics question the integrity of the award’s mechanism and consider them as boosting mechanisms for movies to find lucrative marketing re-birth or launch actors and actresses to super-stardom. For instance in case of Oscar award, millions of dollars are spent on efforts to promote nominees to members of the Academy². Considering such skepticism about different movie awards, it is very important to estimate the boosting effects of awards over the popularity of different movies.

In the past few years, there is a growing recognition of popularity bias in recommendation systems [1, 3, 5, 6]. Different techniques have been proposed to penalize the discrimination of popularity on user ratings in recommendation systems [4, 8, 11]. They target on making neutralized recommendation. However, none of these work explicitly models the popularity and the influence of the “rich get richer” effect, e.g., contributed by movies awards.

Estimating the awards effect on popularity of items over time is important for several reasons. It can help in predicting revenues, future earnings, ranking on recommender systems of awarded items, e.g., airlines, hotels, movies, songs, scientific papers, etc. Our proposed framework models the popularity of items by two aspects, the base merit and the awards’ effects, both of which vary with time due to the variation of subscribed users to movie system and the decrease of attractiveness of movies to users over time. The award effect model \mathcal{D} explicitly estimates the extra viewership of awarded movies, while the base merit model \mathcal{M} is learned for predicting movie popularity without award effect. The two models are shown to be effective for predicting the popularity of movies during March to Sept 2005 in Netflix (Jan-2006 to Dec-2008 in MovieLens), after trained on Netflix data from earlier stage: Dec-1999 to Feb-2005 (MovieLens data from Jan-1996 to Dec-2005). In addition, the model \mathcal{D} is used to analyze how award effect varies on movies with different genres. Moreover, model \mathcal{M} is applied to produce a ranked list of recommended movies without award effect. Through checking the ratings of recommended movies in IMDB and NetFlix, we show that our recommended movies without award effect have higher average rating than the recommended movies with award effect.

The remaining part of this paper is organized as follows. Section 2 discusses related work. Section 3 introduces the proposed model. Section 4 presents experimental evaluation. Section 5 concludes the paper.

2 Related Work

Work related to this study falls into two categories. From the application perspective, this study aligns with the popularity bias in recommendation systems [1, 3–6, 11] and from the problem setting perspective, it is connected to uplift modeling [7, 9].

² <http://www.politicalcampaigningtips.com/oscar-campaigning-the-politics-of-the-academy-awards/>.

Kamishima et al. [4] propose to design an information neutral recommendation system, making the recommendations neutral from a specific view point, i.e., a specific feature of user (e.g., gender) or item (e.g., brand or popularity). They propose a penalty term to address the filter bubble problem of personalized recommendation systems. Their proposed penalty term ensures the statistical independence between the considered neutrality view point and preference scores. In this paper, we do not restrict our study to personalized recommendations and quantify the effects of different awards over the popularity of movies.

While it is clear that collaborative filtering algorithms outperform popularity-based recommendations in terms of accuracy and sales diversity, it has been pointed out that they still suffer from bias. Zhao et al. study a similar subject of popularity bias in recommendation system [11]. They mainly follow an opinion-based weighting function. If popular items are similarly rated by two neighboring users, these popular items are given the weight as the inverse log of the popularity. Otherwise (e.g., two users have different opinions), popular items are given the weights as log of the popularity. For less popular items, the assigned weight is 1. Their purpose is to improve diversity and accuracy in recommendation. However, our goal is to estimate the effect of awards on the popularity of items over time. Our work differentiates the user response and popularity of movies with and without awards.

In uplift modeling [7, 9], two data sets (with and without action) are formed. The task is to model the effect of a particular action for a given instance and to identify the instances for which it is worth to do the action. The primary difference in our settings is that our uplift event is not revocable (i.e., awards are perpetual) and in practice it also becomes hard to have two exact movies with and without certain awards. In addition, we study the temporal effect of awards on the popularity of items over time, rather than only considering popularity with or without award.

3 Modeling the Award Effect on Popularity of Items

3.1 Problem Setting

Suppose we have a set of items (e.g., movies) where each item is viewed or rated at a given time t . Each item i can be described by a vector $X(i)$ with non-temporal attributes (e.g. movie genre), and additionally a temporal attribute vector $Z^t(i)$ that records time related attributes (e.g., average rating of a movie per week) at a certain time t . The popularity of an item (i.e., views or sales count) varies with time, and attains an artificial increase after the occurrence of an award event s , which is also a time varying attribute,

$$\begin{aligned} s^t(i) &= 0 \text{ when there was no boosting event for item } i \text{ at } t; \\ s^t(i) &= 1 \text{ when there was a boosting event for item } i \text{ at } t. \end{aligned}$$

The popularity of each item i can be defined by grouping the rating counts into n identically sized time bins, where each time bin represents an instance of the calendar, e.g., day, week, and month. Denoting the popularity of an item i

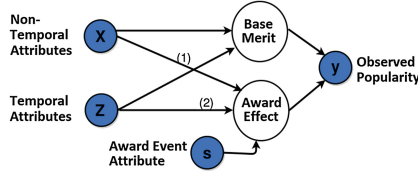


Fig. 1. A model of award affected popularity.

in time bin t by y_{it} , we define the **item instance** e_{it} to be an instance of an item i in time bin t , described as:

$$e_{it} = \{X(i), Z^t(i), s^t(i), y_{it}\} \quad (1)$$

The goal is to model the popularity score y_{it} of an item i at time t (e.g., how many viewers a movie will attract in a given week) given X_i , Z_i^t , and $s^t(i)$.

3.2 Award Effect Modeling

Our proposed model is shown in Fig. 1. The observed popularity is a result of the base merit (depending on X and Z) and the awards effect (introduced by s and affected by X and Z). The observed popularities are modeled as:

$$y = \mathcal{M} + s\mathcal{D}, \quad s \in \{0, 1\}, \quad (2)$$

where \mathcal{M} is the model learned on non-awarded data that predicts base merit of a movie, \mathcal{D} is the model for estimating the award nomination effect or award winning effect. The intuition behind this model is as follows. When the items are not yet awarded/never awarded ($s = 0$), the observed popularity originates from the base merit model (\mathcal{M}) only. When the award event occurs ($s = 1$), the observed popularity comes from two terms: the base merit model (\mathcal{M}) and the award effect term \mathcal{D} . Both models \mathcal{M} and \mathcal{D} can be learned by a regression algorithm, e.g., Support Vector Regression. We focus on designing the learning procedure of using different training data with their corresponding targets. Note that we will consider award nomination and award winning differently. Model \mathcal{D}_{nom} and \mathcal{D}_{win} are learned and applied to nominated items and awarded items, respectively, but following the same learning procedure. Thus we only use subscriptions nom and win when necessary.

For learning \mathcal{M} , \mathcal{D}_{nom} and \mathcal{D}_{win} , we divide training data into three subsets:

- set **A** contains instances of the movies that were neither nominated nor awarded ($s = 0$), or movie instances not yet nominated at time $t < t_{nom}$.
- set **B** contains all the award nominated movie instances after nominations were announced $t \geq t_{nom}$, and instances of award winning movies before winning award but after nomination at time $t_{nom} \leq t < t_{win}$.
- set **C** contains the instances of award winning movies after winning award(s) at time $t \geq t_{win}$.

The learning strategy is generally as follows: Model \mathcal{M} is learned on data \mathbf{A} and its corresponding observed popularity $y^{(A)}$. The award nomination effect model \mathcal{D}_{nom} is learned on data \mathbf{B} and its corresponding effect $d^{(B)} = y^{(B)} - \hat{y}^{(B)}$, where $y^{(B)}$ is the observed popularity and $\hat{y}^{(B)}$ is the predicted popularity when applying the above learned \mathcal{M} on \mathbf{B} . Similarly, the award winning effect model \mathcal{D}_{win} is learned on data \mathbf{C} and its corresponding effect $d^{(C)} = y^{(C)} - \hat{y}^{(C)}$, where $y^{(C)}$ is the observed popularity and $\hat{y}^{(C)}$ is the predicted popularity when applying the above learned \mathcal{M} on \mathbf{C} . As Model \mathcal{M} is free from award nomination/winning effect, $\hat{y}^{(B)}$ and $\hat{y}^{(C)}$ are the non-awarded predictions of popularity for instances in \mathbf{B} and \mathbf{C} , respectively. The observed values $y^{(B)}$ are affected due to the award nomination and $y^{(C)}$ are influenced due to award winning. Taking $d^{(B)}$ and $d^{(C)}$ as the regression targets, the learned \mathcal{D}_{nom} and \mathcal{D}_{win} thus model the award nomination and award winning effect in \mathbf{B} and \mathbf{C} , respectively.

With different forms of award effect, we study three different models given in the following text:

- (1) **Modeling Constant Effect of Awards** ($\mathcal{D}^{(0)}$) assumes that the effect introduced by the awards is constant for each item instance for all the time, i.e., an award introduces a fixed amount of publicity to viewers who want to see the winning movie independently of its content all the time.

The constant effect can be defined as the mean of d , $\mathcal{D}^{(0)} = \frac{\sum_{i \in \mathbf{B}} d_i^{(B)}}{|\mathbf{B}|}$ for nomination instances, and $\mathcal{D}^{(0)} = \frac{\sum_{i \in \mathbf{C}} d_i^{(C)}}{|\mathbf{C}|}$ for awarded instances.

- (2) **Modeling Awards Effect Based on X** ($\mathcal{D}^{(X)}$) assumes that the awards effect depends on the movie non-temporal attributes X , but is independent from temporal attributes, e.g., additional viewers watch the award winning movie because of the genre of this movie. The effect of the awards can be modeled as a function of item non-temporal attributes, $\mathcal{D}^{(X)} = f(X|s = 1)$.
- (3) **Modeling Awards Effect Based on X and Z** ($\mathcal{D}^{(XZ)}$) assumes that the awards effect depends on the movie non-temporal attributes X as well as temporal attributes Z^t . We consider it as the most realistic assumption about the real data. For award nominated or award winning movies, additional viewership will depend on the time since the movie was released and the content of the movie. For instance, if a movie was released eleven months before the award ceremony, maybe most of the people have already seen it anyway or consider it to be an old movie and do not want to see it regardless of the award. The item and time related award effect can be modeled as a function of item and time attributes, $\mathcal{D}^{(XZ)} = f(X, Z^t|s = 1)$.

3.3 Temporal Behavior Award Effect Modeling

We propose Temporal behavior Award Effect Modeling (TAEM) algorithm to solve the learning problem in Eq. (2). Algorithm 1 gives the pseudo code for the learning of \mathcal{M} , \mathcal{D}_{nom} and \mathcal{D}_{win} as discussed in Sect. 3.2, and the prediction after learning. Note that Algorithm 1 applies to the learning of $\mathcal{D}^{(0)}$, $\mathcal{D}^{(X)}$ and $\mathcal{D}^{(XZ)}$, which only differ on the usage of selected feature sets.

Algorithm 1. Temporal Behavior Award Effect Modeling (TAEM)

```

1 begin Model learning
   Input: Data sets  $Train \mathbf{A}$ ,  $Train \mathbf{B}$ ,  $Train \mathbf{C}$ 
   Output: Base merit model  $\mathcal{M}$ ,
           Award effect models  $\mathcal{D}_{nom}$  and  $\mathcal{D}_{win}$ 
2   Learn  $\mathcal{M}$  on training set  $\mathbf{A}$ :
        $\mathcal{M} : e_{it} \mapsto y_{it}, \quad \{e_{it}, y_{it}\} \in Train \mathbf{A}$ ;
3   Make predictions on training set  $\mathbf{B}$  using model  $\mathcal{M}$ :
        $\hat{y}_{it}^{(B)} = \mathcal{M}(e_{it}), \quad e_{it} \in Train \mathbf{B}$ ;
4   Make predictions on training set  $\mathbf{C}$  using model  $\mathcal{M}$ :
        $\hat{y}_{it}^{(C)} = \mathcal{M}(e_{it}), \quad e_{it} \in Train \mathbf{C}$ ;
5   Calculate the residuals  $d_{it}^{(B)} = y_{it}^{(B)} - \hat{y}_{it}^{(B)}$ ;
6   Calculate the residuals  $d_{it}^{(C)} = y_{it}^{(C)} - \hat{y}_{it}^{(C)}$ ;
7   Construct a new set  $\mathbf{B}'$  with  $d_{it}^{(B)}$  as new targets;
8   Construct a new set  $\mathbf{C}'$  with  $d_{it}^{(C)}$  as new targets;
9   Learn the award nomination effect estimation model  $\mathcal{D}_{nom}$  on set  $\mathbf{B}'$ :
        $\mathcal{D}_{nom} : e_{it}^{(B)} \mapsto d_{it}^{(B)}$ ;
10  Learn the award winning effect estimation model  $\mathcal{D}_{win}$  on set  $\mathbf{C}'$ :
        $\mathcal{D}_{win} : e_{it}^{(C)} \mapsto d_{it}^{(C)}$ ;
11 begin Evaluation and Recommendation
   Input: Data sets  $Test \mathbf{A}$ ,  $Test \mathbf{B}$  and  $Test \mathbf{C}$ ,
           Award event attribute  $s_{nom}$ ,  $s_{win}$ 
   Output: Estimated  $y$ , ranking of test data sets
12   $y_{it} = \mathcal{M} + s_{it}\mathcal{D}$ ;
13  Rank testing data by  $y_{it}$  in each time bin  $t$ 

```

According to the application need, our proposed algorithmic solution can be applied in fashion of offline, online and adaptive settings.

Offline Learning: In this setting, we learn models \mathcal{M} , \mathcal{D}_{nom} and \mathcal{D}_{win} from a given training data that include instances collected until time t (line 1–10 in Algorithm 1). Predictions are made for the items from week $t + 1$ on-wards (line 11–13 in Algorithm 1). The learned models \mathcal{M} , \mathcal{D}_{nom} and \mathcal{D}_{win} are fixed without updating. This setting is a realistic scenario when there are enough data to learn accurate models and the relation between models and targets is fixed and will not change over time.

Online Learning: In this setting, models \mathcal{M} , \mathcal{D}_{nom} and \mathcal{D}_{win} are updated over time as more data arrive. The initial model is learned on all data up to time t inclusive. Predictions are made for time $t + 1$. At time $t + 1$, the models are updated or relearned using all the available data up until time $t + 1$ inclusive (run again line 1–10 in Algorithm 1). The new models are used to make predictions for time $t + 2$. This process is repeated for all coming weeks. This scenario is useful when there are not enough training data to learn an accurate initial model. More accurate models are learned as more data arrive.

Adaptive Learning: This scenario is useful when the underlying concept is changing over time, thus the relation between the inputs and the target needs to be updated. Before making prediction for $t + 1$, we re-learn the models from data in the last h (a fixed window size) weeks. These most up-to-date data enable the model to capture the dynamic, usually hidden, and important factors affecting the popularity.

4 Experimental Evaluation

4.1 Data Sets and Experimental Setting

We use two real world data sets Netflix [2] and MovieLens 10 million ratings data set³. The movies selected for our experiments are released from Dec-1999 to Sep-2005 in Netflix and from Jan-1996 to Dec-2008 in MovieLens when ratings are available for analysis. From Netflix (MovieLens), 134 (99) award winning movies, 53 (98) award nominated movies and 178 (102) movies that were never nominated nor awarded are selected. As described in Sect. 3.2, instances of these movies then define three sets **A**, **B** and **C**.

Each movie instance is described by 47 non-temporal attributes (forming a vector X) and 14 time related attributes (forming a vector Z). The attributes of X extracted from IMDB (www.imdb.com), Rotten Tomatoes (<http://www.rottentomatoes.com>) and Box Office Mojo (<http://www.boxofficemojo.com>) contain 9 numerical attributes: *budget*, *languages released count*, *run time*, *overall average rating*, *opening gross*, *total gross*, *opening theaters count*, *total theaters count*, *box office ranking in release year*, and 38 binary attributes, which include an attribute of *is Adult* (i.e., *is this movie only for adults*), and *is Sequel Or Adaptation*, nine binary attributes showing the genre of a movie, ten binary attributes for showing the studio in which movie is released, five binary attributes for showing the release year of movie, twelve binary attributes to show release month of movie. The Z includes 14 numerical attributes: *movie age*, *DVD release age*, *weekly average rating*, *popularity of previous week*, *popularity of second last week*, *popularity of third last week*, *popularity of fourth last week*, *average popularity of last two weeks*, *average popularity of last three weeks*, *number of ratings since release*, *total number of views (of all movies in the previous week)*, *users joined the system in given week*, *total users in the system till this week*, and *Netflix subscribers growth*. We have eventually 58,059 instances from Netflix binned by week, and 33,138 instances from MovieLens binned by month.

In Netflix data sets **A**, **B** and **C**, we use instances from the first 275 weeks (Dec-1999 to Feb-2005) for training and instances of the last 32 weeks (March to Sept 2005) for testing. The selection is for the purpose of evaluating the predictive power of our models. In MovieLens data set, instances of first 120 months (Jan-1996 to Dec-2005) are used for training and the last 36 months (Jan-2006 to Dec-2008) are used for testing. We report only the results obtained on

³ at <https://grouplens.org/datasets/movielens/10m/>.

NetFlix data set here due to the space limitation. Similar results and observations are found in MovieLens data set.

In all the experiments, we use Mean Absolute Error (MAE) as an evaluation measure. It measures how close the predicted values are from the observed ones.

4.2 Award Effect Quantification

In this section, we show the temporal award effect estimation capacity of our proposed method at individual instance level in Figs. 2, 3, and 4, where x-axis is the index of test instances (ordered by their actual popularity) and y-axis is the popularity. The model \mathcal{D} for award effect quantification here is the one $\mathcal{D}^{(XZ)}$ (learned from both feature X and Z), as $\mathcal{D}^{(XZ)}$ performs better than $\mathcal{D}^{(X)}$ and $\mathcal{D}^{(0)}$ (see the comparison in next section).

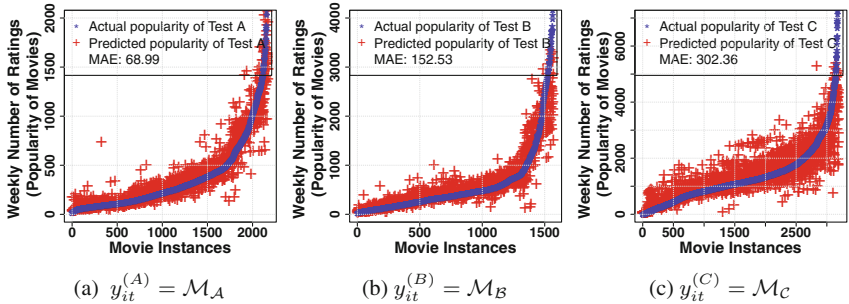


Fig. 2. Prediction in increasing order of popularity of three models learned from training sets of **A**, **B**, and **C** separately. (Color figure online)

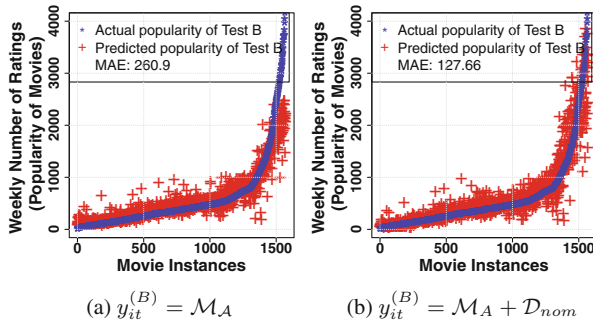


Fig. 3. Prediction of TAEM for award nominated instances in set **B**, estimation without award nomination effect (a) and with individual award nomination effect (b).

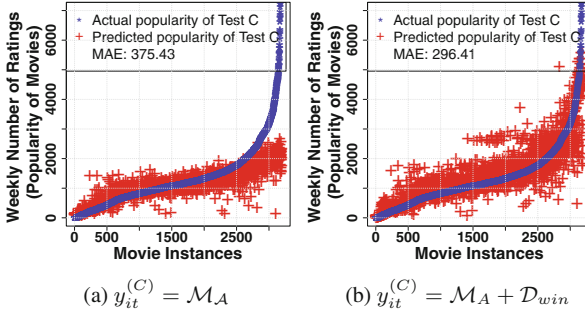


Fig. 4. Prediction of TAEM for awarded winning instances in set **C**, estimation without award winning effect (a) and with individual award winning effect (b).

Figure 2(a–c) show the predictions (red stars) of three specialized models, \mathcal{M}_A , \mathcal{M}_B and \mathcal{M}_C , (based on Random Forest Regression) learned over the training sets **A**, **B**, and **C** and tested over the test sets **A**, **B**, and **C**, respectively. The actual target values (blue crosses) are also shown for a better comparison with predicted values. These models are considered as special cases where enough data in data sets **A**, **B** and **C** are available to learn a specialized model for each category of instances. We can see that the predictions of these models closely follow the actual targets. We expect that such models would be the best performing ones over their respective data sets. Note that our test set is from the later period when subscribers and the number of online movies have increased. Such a growth thus brings difficulties in prediction. Hence, high MAE scores are intuitive and acceptable.

Figure 3(a) shows the performance of our proposed TAEM over test set **B** but ignoring the nomination award effect, i.e., predicting by \mathcal{M}_A only. As expected, such predictions are lower than the actually observed popularity that includes different amount of award effect for different instances. Figure 3(b) shows the prediction of $\mathcal{M}_A + \mathcal{D}_{nom}$, which includes the estimated award effect. We see that our proposed method quantifies accurately the award effect in the popularity of inflated instances due to award nomination’s boosting effect. When we add this calculated award effect to the non-awarded prediction in Fig. 3(a), it becomes very close in Fig. 3(b) to the predictions of specialized but award effect estimation model learned over the award nomination affected popularity data set **B** (shown in Fig. 2(b)).

Figure 4(a–b) shows the award effect quantification of award winning movies (set **C**) by TAEM. Similar to Fig. 3, it can be seen that the predictions with ignorance of awards are lower than the award affected popularity observation. Moreover, the award effect due to winning is greater than that due to nomination. When adding the estimated award effect to the non-awarded prediction in Fig. 4(a), as shown in Fig. 4(b), the scores are close to the predictions made by specialized model in Fig. 2(c). We can conclude from Figs. 3 and 4 that TAEM well quantifies the award effect in the popularity of each individual instance.

4.3 Comparison of Different Award Effect Estimation Models

We now compare different award effect models presented in Sect. 3.2 and three TAEM learning settings in Sect. 3.3. We evaluate the performance of these models by their predictive power measured in MAE.

1. **Single model:** prediction is made by a single model learned from the combination of all training sets **A**, **B**, and **C**.
2. TAEM with constant award effect (**TAEM with $\mathcal{D}^{(0)}$**).
3. TAEM with award effect learned from feature X only (**TAEM with $\mathcal{D}^{(X)}$**).
4. TAEM with award effect learned from feature X and Z (**TAEM with $\mathcal{D}^{(XZ)}$**).
5. **Three models**, which is built in an ideal scenario where enough data are available to build three separate models: \mathcal{M}_A , \mathcal{M}_B , and \mathcal{M}_C from the training sets **A**, **B**, and **C**, respectively. These three models are expected to achieve the highest accuracy when each of them is applied only to instances in its own set, as shown in Fig. 2. However, they failed to quantify and correct the award effect.

Figure 5 compares the MAE (y-axis) of different award effect models (x-axis) tested over the aggregation of test **A**, **B** and **C** in offline, online and adaptive settings. Generally, TAEM with different award effect estimation settings performs better than the single model. The advanced TAEM with award effect learned from feature X and Z (TAEM with $\mathcal{D}^{(XZ)}$) performs as good as the ideal three-models. Importantly, our method has the capability to estimate the award effect for each privileged item (as already shown in Sect. 4.2). In addition, we see that the online and adaptive settings ($h = 100$) have better performance than the offline setting.

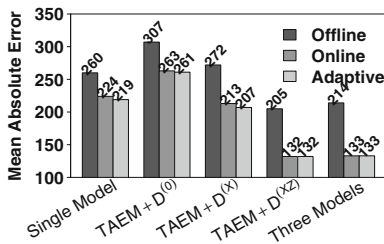


Fig. 5. Comparison of different award effect estimation models.

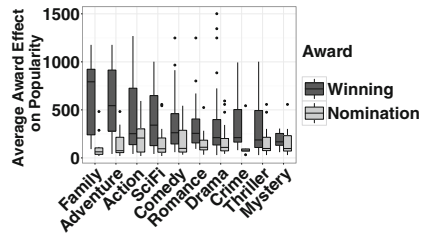


Fig. 6. Award nomination and winning effect by genre of movies

4.4 Award Effect on Different Movie Genres

Since we are able to estimate the award effect by \mathcal{D} , it is interesting to study how the award effect varies on movies with different genres. Figure 6 shows the average effect of award nomination and award winning over all the test weeks, for movies with different genres. Generally, we can see that award winning has higher influence than award nomination on movie popularity, especially for *Family*, *Adventure*, *Action* and *Science Fiction* movies. This conforms with the behavior of movie viewers. Movies in these genres can be accepted by a large population of viewers. Award winnings usually cause people’s interest and curiosity to watch the awarded movies.

4.5 Recommendation of Top-k Based on Model \mathcal{M}

Given the task of generating top-k movies for recommendation, in this section, we validate the usefulness of our model by comparing the quality of top-k movies recommended according to the observed popularity (with award effect) and the predicted popularity based on model \mathcal{M} (base merit of movies, without award effects). In other words, we apply model \mathcal{M} on all movies for evaluating their base popularity by excluding the extra viewership introduced by awards. It is interesting to see whether our selection based on popularity estimated by \mathcal{M} without award effect is better than the selection based on observed popularity with award effect. To evaluate if the ranked top-k movies are *good*, we check their actual ratings given by users not only in NetFlix (1–5 scale), but also in IMDB (1–10 scale) for checking the ratings given by users in a different system, IMDB. High ratings indicate *good* movies.

Figure 7 shows the average rating of top 20 movies in the ranking list produced by our model estimation (Award Effect Treated Ranking) and observed popularity (Observed Ranking). Two independent movie rating systems (IMBD and NetFlix) consistently verify the *better* quality of movies in award effect treated ranking, which has significantly higher average rating than observed ranking in all test weeks.

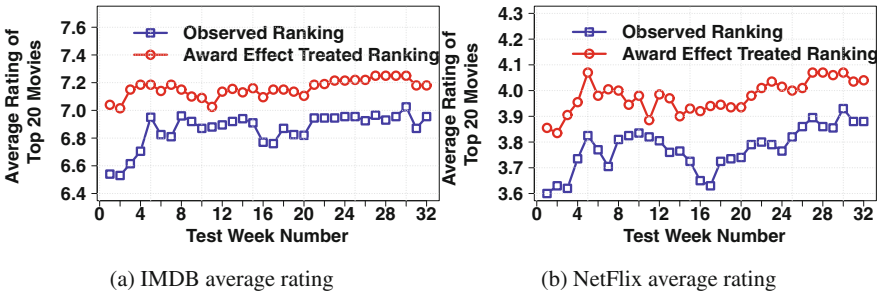


Fig. 7. Average rating of the top-20 movies ranked by observed popularity (with award effect) and by the award effect treated popularity in each test week.

To study if *better* movies are always ranked higher in different settings of k , we take average rating of the top- k movies in all test weeks for each setting of k , and show the results in Fig. 8. The x-axis represents the varying k , while the y-axis is the average ratings of the top- k movies over all 32 test weeks. We see consistently in IMDB and NetFlix that award effect treated ranking generates movies list with *better* quality than observed ranking. Thus, our proposed model serves well the purpose of recommendation based true merit of items.

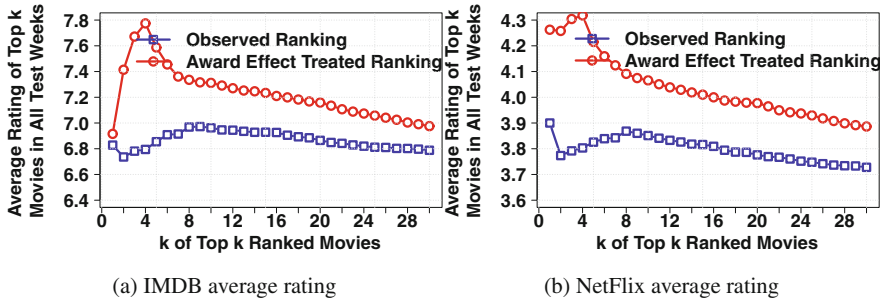


Fig. 8. Average rating of top- k movies when varying the value of k .

5 Conclusion and Future Work

In this paper, we study the temporal effects of award nomination and award winning as popularity boosting events, and propose a framework for modeling the movie popularity in terms of base merit and the award effect. To model the popularity at individual movie level, we collected 47 non-temporal attributes and 14 time-related attributes for each movie. The experimental evaluation confirms the effectiveness of our proposed model on learning the temporal effect of award nomination and award winning. Also, the model can be used for producing a list of top- k recommended movies in better quality.

In this paper, we have assumed a simplified scenario that our task is to determine the award context-aware list of the best items at certain time for a group of users. In future, we plan to extend our methodology to personalized award context-aware recommendations and study the influence of multiple boosting events occurring at the same time.

References

1. Adomavicius, G., Kwon, Y.: Improving aggregate recommendation diversity using ranking-based techniques. *IEEE TKDE* **24**(5), 896–911 (2012)
2. Bennet, J., Lanning, S.: The netflix prize. *KDD Cup and Workshop* (2007)

3. Breese, J.S., Heckerman, D., Kadie, C.: Empirical analysis of predictive algorithms for collaborative filtering. In: Proceedings of the Fourteenth Conference on UAI, pp. 43–52 (1998)
4. Kamishima, T., Akaho, S., Asoh, H., Sakuma, J.: Enhancement of the neutrality in recommendation. In: Proceedings of the RecSys Workshop on Human Decision Making, pp. 8–14 (2012)
5. Lai, S., Liu, Y., Gu, H., Xu, L., et al.: Hybrid recommendation models for binary user preference prediction problem. In: KDD Cup, pp. 137–151 (2012)
6. Oh, J., Park, S., Yu, H., Song, M., Park, S.T.: Novel recommendation based on personal popularity tendency. In: IEEE ICDM, pp. 507–516 (2011)
7. Radcliffe, N.J., Surry, P.D.: Real-world uplift modelling with significance-based uplift trees. White Paper TR-2011-1, Stochastic Solutions (2011)
8. Ruggieri, S., Hajian, S., Kamiran, F., Zhang, X.: Anti-discrimination analysis using privacy attack strategies. In: Calders, T., Esposito, F., Hüllermeier, E., Meo, R. (eds.) ECML PKDD 2014. LNCS (LNAI), vol. 8725, pp. 694–710. Springer, Heidelberg (2014). doi:[10.1007/978-3-662-44851-9_44](https://doi.org/10.1007/978-3-662-44851-9_44)
9. Rzepakowski, P., Jaroszewicz, S.: Decision trees for uplift modeling with single and multiple treatments. *Knowl. Inf. Syst.* **32**(2), 303–327 (2012)
10. Schafer, J.B., Konstan, J., Riedl, J.: Recommender systems in e-commerce. In: Proceedings of the 1st ACM Conference on Electronic Commerce, pp. 158–166 (1999)
11. Zhao, X., Niu, Z., Chen, W.: Opinion-based collaborative filtering to solve popularity bias in recommender systems. In: Proceedings of International Conference on Database and Expert Systems Applications, pp. 426–433 (2013)