Vol.2015-DBS-161 No.9
Vol.2015-IFAT-119 No.9
2015/8/5

Item Age Effect Against Global User Preference in Latent Model of Recommendation System

Pannawit Samathiyadikun[1,a)] Atsuhiro Takasu[1,2,b)]

Abstract: Due to highly competitive among various companies and widely use of recommendation systems, they need additional abilities to attract systems users, beside accuracy. In some cases, recommending items that may be interesting in near future is far more attractive, but analyzing effect of time is very complex and difficult. In this work, we starts at analyzing and stating assumptions about effect of time such as item age,on global user preference of item. Then they are proven by probabilistic modelling with a latent variable included for avoiding sparsity problem, using different distributions on discretized time data. The results of model show the significant improvement of accuracy compared to other traditional recommendation systems, for both individual and list recommendation. Furthermore, they also show that the accuracy, which is related to the number of possible values of the latent variable, is affected by cardinality of discretized time data.

1. Introduction

Recommendation system is used in many electronic commercial (e-commerce) site. For example, recommending goods in Amazon, movies in Netflix, or music in Last.fm. By recommending interesting things to the people who use the system, it keeps people stay, use, and buy goods in the site, because system's users is automatically given the item they might serching, continuously. On the other hand, it increases popularity and revenue of company. Then the accuracy is the most important characteristic the system would have.

One of the common problems for recommendation system is the data sparsity, which is occurred when there are a lot of users and items, but only a few ratings is given compared to all possible ratings. and the system may not have enough data to make an effective recommendation. Normally, users' history behaviors is used in the most of the systems to produce recommendation for them. Many different methods are researched and used in the system to improve an it's accuracy and overcome sparsity problem.

Funk's SVD is one of the traditional latent model, the model includes of latent factors, that won Netflix Prize among other methods for a million dollar. This method maps numerical format of item characteristics and user preferences into the same lower latent factor dimension. That is, both item characteristics and user preferences are associated with numerical vector of latent factors, where each element in vector represent to each latent factor, then user's rating score for item is inferred from them. Technically, idea of this method is similar to singular value decomposition (SVD), but all matrice values are stochastically learned by gradient descent or alternating least squares [1].

Other kind of approach of latent model is latent probabilistic model, which it's parameters are learned in probabilistic way. Bayesian approach of Flexible Mixture Model (BFMM) is latent probabilistic model that uses 2 different latent factors for users preferences and items characteristics, seperately. Different from Funk's SVD that it's latent factors is presented in numerical vector, which doesn't have any meaning or hard to infer to what do those numerical value mean. Each element in vectors of item or user latent factors used in BFMM is represent to proportion or likelihood of item or user latent factor in the whole given data. From this angle of view, the latent factor in latent probabilistic model can be inferred as class or group of the same characteristics of data. Moreover, each user and item also has it own latent factor vector, that is the model can softly categorize user and item to each latent factor proportionally [2].

Due to widely used of recommendation systems nowadays, beside accuracy, additional abilities to attract systems users is needed. Concept drift presents changing of things over time, is introduced to be another model assumption for increasing model's accuracy. The examples of concept drift in recommendation system are is that popularity of item and user's identity or preference may change, and those also effect item's rating given by user. Furthermore, if the model implements this concept correctly, then the changes in future is possible to be discovered, and can be used for predicting the ratings that will be given in future.

SVD++, which is modified SVD that makes use of implicit feedback information[1], is integrated with concept drift to be able to tracking drifting of user's preference over time, named timeSVD++. It's result shows that timeSVD++ archives improving accuracy beyond it's ancestors, SVD and SVD++[3]. However, future concept drift still cannot be tracked in this work,

1 SOKENDAI (The Graduate University for Advanced Studies), Miura, Kanagawa 240–0193, Japan
2 National Institute of Informatics, Chiyoda, Tokyo 101–8430, Japan
a) s-pannawit@nii.ac.jp
b) takasu@nii.ac.jp

Table 1: Characteristics of MovieLens 10M data set after integrating with IMDb

	MovieLens 10M
Ratings	6,339,088
Users	69,875
Items	5,379
Scores	0.5,1,1.5,...,5
Time range	01/29/1996 - 01/05/2009
Sparsity	98.9831%

which is what we are interested, but we realize that concept drift on the timeline is really hard to track, so we try to use another kind of time data instead of using time stamp when rating is generated.

In the time aspect for the most kind of item, the new item would be preferred by customer more than the old one. Because of this truth, item age when rating is given is introduced to use as time data in this work. Our final goal is modelling the future concept drift, but this work is still in the early stage. We model the relation only between item's characteristics, given rating, and age when it is rated by using latent probabilistic graphical model with different distributions. User's preference is not included, to keep model be simple, Evaluation result of this model with various number of latent factors helps us judge that item's age is usable time data that affect given rating or not, and which distribution is suitable for this model.

2. Data Analysis

MovieLens data set of 10 millions rating data [6], which include rating scores of some pairs of user and item with timestamp when it was given, are used in our experiments. However, the data we really need to experiment with is item age when rating score was given, but it is not available. Fortunately, it can be obtained by subtracting rating timestamp with item release date, which available in Internet Movie Database (IMDb) [5]. We linked MovieLens and IMDb together via movie title through the procedure provided by [4], but some ratings is missing because of different movie title used in each source. The details of 2 different data sets after linking with IMDb are listed in **Table 1**.

After data merging is done, item age at the time rating is given can be calculated as:

$$item\,age = timestamp(\boldsymbol{x}) - release\,date(x_v)$$
$$x_t = \left\lceil \frac{item\,age}{time\,window\,size} \right\rceil \quad (1)$$

where a tuple of observed data that includes of item ID, rating score, and item age when it is rated, is defined as $\boldsymbol{x} = (x_v, x_r, x_t) \in \mathcal{X}$, respectively. Finally, item age is discretized to be an integer for ease of calculation, which results x_t. Various sizes of time window are experimented to find the suitable one that can visualize patterns of rating score and item age, which a half year (182.625 days) is decided to use in this work.

Fig. 1 shows frequency of each rating score in each item age bin. As inferred from peak of **Fig. 1a**, it takes at least 6 months after movie releases date to get popular. One of possible reasons is most of the audience prefers to watch movie in somewhere else from cinema. **Fig. 1b** shows that movie age is the observable factor which makes scores proportions a little bit vary for each time window. However, if we consider ratings where score equal and above 3.5 as good ratings, else are bad ratings. It tends to have more of ratings with good scores for old movies. According to normal behaviors of users when making decision to pick up old movie, nobody wants to watch old movie actually as described from the right side tail of Fig. 1a. They watch them on purpose other than just updated their knowledge, such as studying or researching. So those movies are carefully picked up by many criteria, e.g., synopsis, director.

3. Proposed Model

In this section, we introduce the probabilistic model that is built to prove our assumptions about effect of item age on global preference of item.

3.1 Time-Aware Latent Model

Due to complexity and uncertainty caused by effect of time context, we would like to build a simple model that is able to reasonably classify or capture a pattern of relation of item age among other kinds of data. Firstly, let us define some notation we use through this paper. $\mathcal{X}$ is set of all observed data, which includes of item $x_v \in \mathcal{X}_\mathcal{V}$, rating score $x_r \in \mathcal{X}_\mathcal{R}$, and item age $x_t \in \mathcal{X}_\mathcal{T}$. $\mathcal{C}$ is set of classes c that indicates pattern of each tuple of observed data $\boldsymbol{x} = (x_v, x_r, x_t) \in \mathcal{X}$. Latent variable z, which is only an unobserved data in this model, indicates the class membership for a tuple of observed data $\boldsymbol{x}$. Then $\mathcal{Y}$ is set of complete-data, where each tuple $\boldsymbol{y} \in \mathcal{Y}$ is $(\boldsymbol{x}, z)$.

Different from the most of classification methods, by using latent variable in this task, it lets $\boldsymbol{x}$ be a member in multiple classes that is called soft-clustering, where $\theta_{z_{(i)}=c}$ is the probability that describes how much is i-th of $\boldsymbol{x}$ likely to be a member of class c.

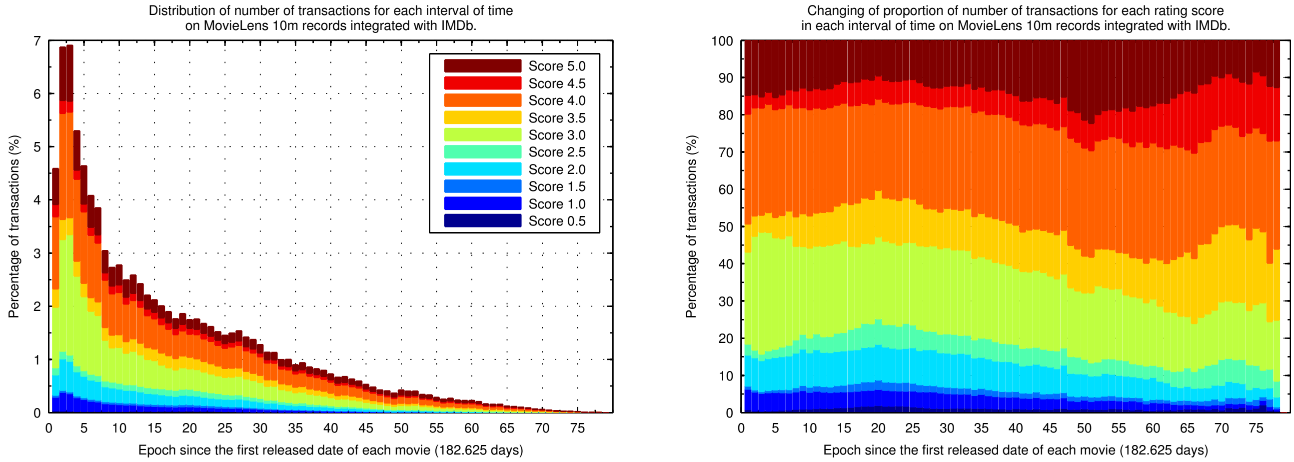
We assume all data is generated from probability that is multinomial distribution for simplification, as follows:

$$z \sim p(z) \equiv \text{Multi}(\vec{\theta}) \quad (2)$$
$$x_v \sim p(x_v|z) \equiv \text{Multi}(\vec{\nu}_z) \quad (3)$$
$$x_r \sim p(x_r|z) \equiv \text{Multi}(\vec{\rho}_z) \quad (4)$$
$$x_t \sim p(x_t|z) \equiv \begin{cases} \text{Multi}(\vec{\eta}_z) \\ \text{Pois}(\eta_z) \end{cases} \quad (5)$$

Except for item age, which can be also generated from Poisson distribution in addition to multinomial distribution. Poisson distribution is experimented in this model, because of the shape of trend of number of observed data for each time bin discovered by plotting the graph shown as Fig. (1-left), is fitted to Poisson distribution. It may results the greater accuracy than using multinomial distribution.

Then likelihood of the complete-data in this model is defined as:

$$\begin{aligned} p(\mathcal{Y};\Theta) = & p(\mathcal{X},\mathcal{Z};\Theta) = \prod_i p(\boldsymbol{x}_{(i)}, z_{(i)};\Theta) \\ & = \prod_i p(x_{t(i)}|z_{(i)}) p(x_{r(i)}|z_{(i)}) p(x_{v(i)}|z_{(i)}) p(z_{(i)}) \\ & = \prod_i p(x_{t(i)}|z_{(i)}) \rho_{z_{(i)} x_{r(i)}} \nu_{z_{(i)} x_{u(i)}} \theta_{z_{(i)}} \end{aligned} \quad (6)$$



(a) Bar graph shows distribution of ratings along item age.

(b) Bar graph shows proportion of rating score in each item age bin.

Fig. 1: Graph shows trend of proportion of each rating score over discretized item age on MovieLens 10M data set

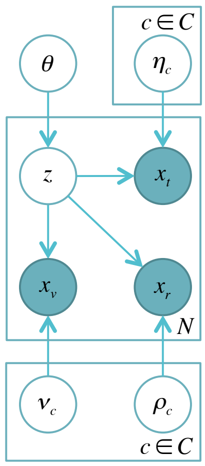


Fig. 2: Probabilistic graphical model of proposed model

where $p(\boldsymbol{x}_{(i)}, z_{(i)}; \Theta)$ is joint probability and

$$p(x_{t(i)}|z_{(i)}) = \begin{cases} \eta_{z_{(i)} x_{t(i)}} & \text{if } x_{t(i)} \sim \text{Multi}(\vec{\eta}_{z_{(i)}}) \\ \frac{\eta_{z_{(i)}}^{x_{t(i)}}}{x_{t(i)}!} e^{-\eta_{z_{(i)}}} & \text{if } x_{t(i)} \sim \text{Pois}(\eta_{z_{(i)}}) \end{cases} \tag{7}$$

3.2 Learning and Prediction Procedures

The well-known maximum-likelihood estimator, Expectation and Maximization (EM) algorithm is used to learn proposed model's parameters $\Theta^{(t)} = \{\theta, \vec{\nu}_z, \vec{\rho}_z, \vec{\eta}_z\}$.

This iterated algorithm switches back and forth between 2 steps, where for each iteration t, firstly, joint-posterior probability of the latent class $c \in C$ for each i-th observed data is computed in expectation step (E-step) with following equation:

$$p(c|\boldsymbol{x}_{(i)}; \Theta^{(t)}) = \frac{p(\boldsymbol{x}_{(i)}, c; \Theta^{(t)})}{\sum_{z=c \in C} p(\boldsymbol{x}_{(i)}, z; \Theta^{(t)})} \tag{8}$$

Then in the maximization step (M-step), the model parameters are gradually optimized according to computed joint-posterior probability of the latent class in E-step as:

$$\theta_c^{(t+1)} = \frac{\sum_i p(c|\boldsymbol{x}_{(i)}; \Theta^{(t)})}{\sum_i \sum_{z=c \in C} p(z|\boldsymbol{x}_{(i)}; \Theta^{(t)})} \tag{9}$$

$$\nu_{cx_v}^{(t+1)} = \frac{\sum_{i:x_{v(i)}=x_v} p(c|\boldsymbol{x}_{(i)}; \Theta^{(t)})}{\sum_i p(c|\boldsymbol{x}_{(i)}; \Theta^{(t)})} \tag{10}$$

$$\rho_{cx_r}^{(t+1)} = \frac{\sum_{i:x_{r(i)}=x_r} p(c|\boldsymbol{x}_{(i)}; \Theta^{(t)})}{\sum_i p(c|\boldsymbol{x}_{(i)}; \Theta^{(t)})} \tag{11}$$

if $x_t \sim \text{Multi}(\vec{\eta}_z)$:

$$\eta_{cx_t}^{(t+1)} = \frac{\sum_{i:x_{t(i)}=x_t} p(c|\boldsymbol{x}_{(i)}; \Theta^{(t)})}{\sum_i p(c|\boldsymbol{x}_{(i)}; \Theta^{(t)})} \tag{12}$$

if $x_t \sim \text{Pois}(\eta_z)$:

$$\eta_c^{(t+1)} = \frac{\sum_i p(c|\boldsymbol{x}_{(i)}; \Theta^{(t)}) x_{t(i)}}{\sum_i p(c|\boldsymbol{x}_{(i)}; \Theta^{(t)})} \tag{13}$$

Although, model parameters are all randomized and normalized before going through EM algorithm, we named this way of initialization as *RandomAll*. Because latent class is the only variable that is unseen from data set, then we randomly initialize only model parameter of latent classes ($\vec{\theta}$). On the other hand, another model parameters are initialized in frequentism way, for example, probability that an item v is member of class c ($\nu_{cv} = p(v|c)$) is computed by normalized N_v.

Unlike multinomial parameters, Poisson parameters is totally different. It doesn't has value boundary, while the multinomial's must be between 0 and 1. So we decide to initialize them to have the most coverage on all time windows. For example, if there are 79 time windows and 5 classes in this experiment. The Poisson parameter for each class will be (0,20,40,60,79), respectively. We named this alternate initialization of model parameters that only model parameter of latent classes is randomized as *RandomClass*.

In the end, those parameters that are already well-optimized for a set of data, is used to predict global preference of target item, at the specific item age, by computing expected value of score from marginalized model's joint probability, as follows:

$$\hat{r}(\boldsymbol{x}) = \sum_{r\in\mathcal{R}} r \frac{p(\boldsymbol{x}; x_r = r)}{\sum_{r'\in\mathcal{R}} p(\boldsymbol{x}; x_r = r')} \tag{14}$$

$$p(\boldsymbol{x}; x_r = r) = \sum_{z=c\in C} p(x_t|z)\rho_{zx_r}\nu_{zx_u}\theta_z$$

4. Experiments

In this section, experiment results are presented in order to address two topics.

(1) Model of both multinomial and Poisson distributions are experimented for observing which distribution is better at fitting to test data?

(2) Various models with different number of of latent classes are experimented for observing effect to model accuracy, and analyzing the relation between number of latent classes and number of time windows.

(3) Performance of Funk's SVD, which is traditional personalized model, is compared for observing and analyzing how much does item age effect an accuracy of unpersonalized prediction?

4.1 Evaluation Metrics

Model accuracy is evaluated for 2 different kinds of results: predicted numeric rating score, and list of k-ranked item recommended by the model. Mean absolute error (MAE) is used to evaluate difference of the prediction from the actual rating scores on items that user voted.

$$MAE = \frac{1}{|\mathcal{X}|} \sum_{\boldsymbol{x}\in\mathcal{X}} |\hat{r}(\boldsymbol{x}) - r(\boldsymbol{x})| \tag{15}$$

List of k-ranked item is produced by descending ordering those items rated from user by their predicted rating score, where the recommended list of k-ranked item for user x_u is defined by $\hat{\mathcal{X}}_{\mathcal{V}}(x_u) = \{\hat{x}_{v(1)}, \ldots, \hat{x}_{v(rank)}, \ldots, \hat{x}_{v(k)}\}$. We have 2 evaluation metrics for measuring the accuracy in this case. The first is precision, which can be calculated by counting how many items in the ranked list user preferred. In this work, we assume that user prefers item that is rated more than the half of score's range ($\mathcal{R}/2$). That is user gives an actual rating more than 5 score of 10-score range, if user likes the item. Mean precision of k-ranked item list is calculated as:

$$P@k = \frac{1}{|\mathcal{X}_{\mathcal{U}}|} \sum_{x_u\in\mathcal{X}_{\mathcal{U}}} \sum_{\hat{x}_v\in\hat{\mathcal{X}}_{\mathcal{V}}(x_u)} \sigma(r(x_u, \hat{x}_v) > \mathcal{R}/2) \tag{16}$$

Another evaluation metric for ranked item list is normalized discounted cumulative gain (nDCG). This metric is different from precision, because it evaluate priority-wise by order of the item. That is value of nDCG is large, if the beginning part of the list is correctly ordered. In contrast, value of nDCG is small in case of it's the tail part is correctly ordered, but the beginning part is incorrectly ordered. Formula for calculating nDCG is as follows:

$$nDCG@k = \frac{1}{|\mathcal{X}_{\mathcal{U}}|} \sum_{x_u\in\mathcal{X}_{\mathcal{U}}} \frac{DCG@k(\hat{\mathcal{X}}_{\mathcal{V}}(x_u))}{IDCG@k(\hat{\mathcal{X}}_{\mathcal{V}}(x_u))} \tag{17}$$

$$DCG@k(\hat{\mathcal{X}}_{\mathcal{V}}(x_u)) = \sum_{rank=1}^{k} \frac{2^{r(x_u, \hat{x}_{v(rank)})}}{log_2(1 + rank)} \tag{18}$$

where $IDCG@k$ is an ideal $DCG@k$ that can be calculated in case of the list of recommended items is correctly ordered, which results in high relevance score, or actual rating score in this experiment.

4.2 Evaluation Result

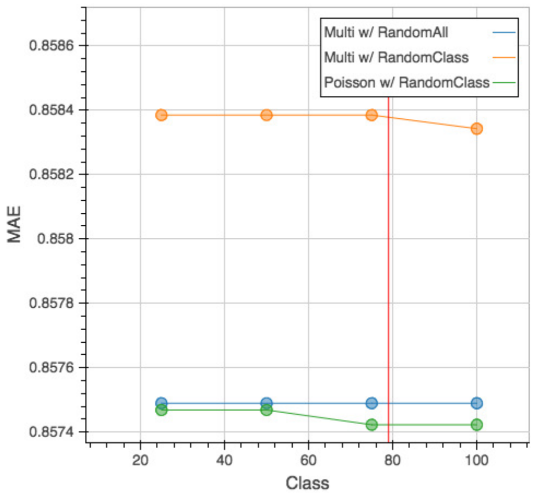


Fig. 3: Evaluation result of predicted rating score by MAE metrics with different number of classes and model, where red line is at 79 classes

We experiment the proposed model with various cardinality of class: 25, 50 ,75, 100 classes, 2 different ways of model's parameters' initialization, and distributions for item age as described in Section 3.

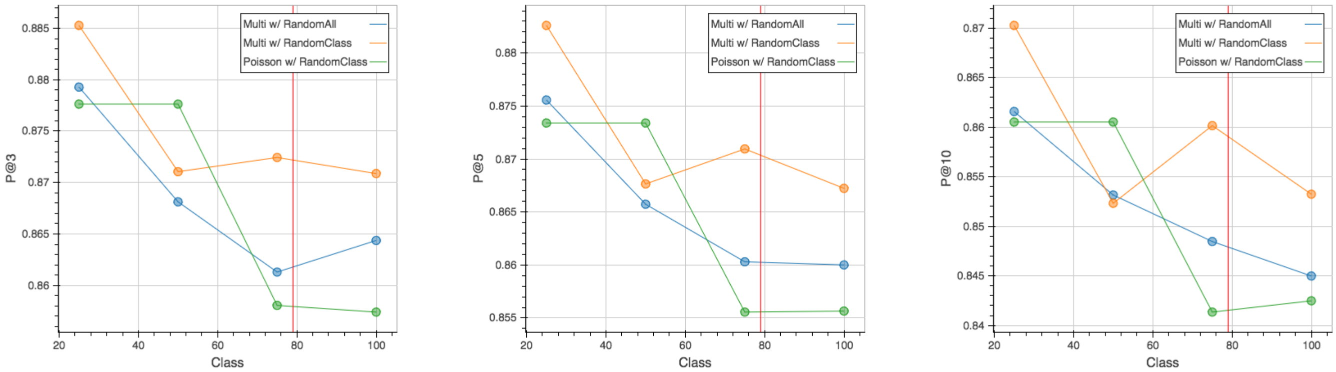
By comparing models' accuracy among various number of classes from different matrics, model with 25 classes provides the most accurate result, and its accuracy is decreased as number of classes is increased. Therefore, the suitable number of classes may be less than 25. We will figure this out in future work.

Although, Poisson model has the greatest accuracy measured by MAE, but it is not significantly greater than others that much (different between best and worst result of MAE is 0.001), as shown in Fig. (3). So we will not take MAE metric into account for judging model's accuracy.

The winner is 25-classes multinomial model with RandomClass, for any k. Precision and nDCG of k-ranked items list for all model settings is decreased as k is increased, because items in tail of recommended list are not predicted accurately. However, according to trend of accuracy of each model, they are really hard to analyze how does cardinality of class effect model's accuracy.

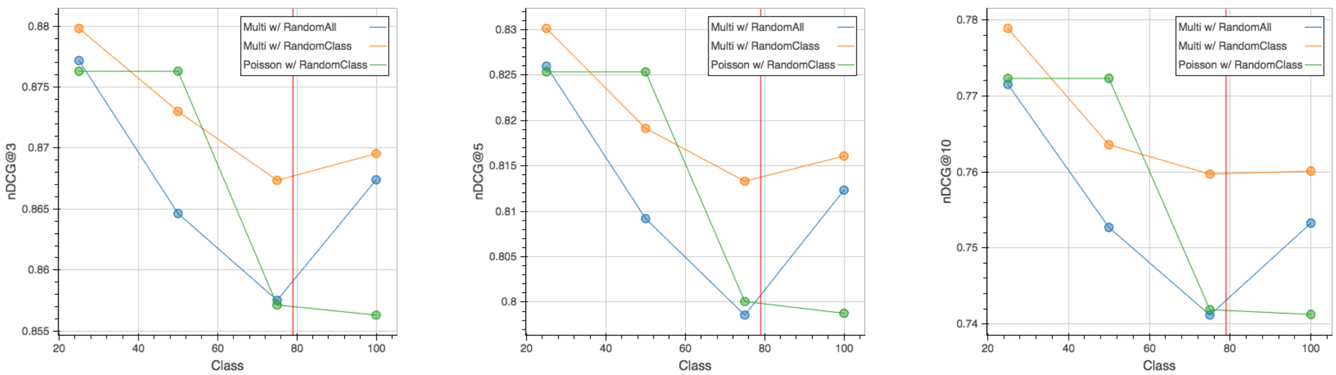
The things we can see is that, trends' slope is became shallow as k is increased. From the aspect of cardinality of class, in some case, accuracy is also decreased as number of classes is increased until around 50-75 classes, then it turns to be increased again as shown in Fig. (4) and (5). However, this phenomenon may involve with assigned number of time windows, 79 time windows in this experiment.

Furthermore, there is conflict of obtained evaluation results via precision and nDCG in some cases. When taking experiment of 75-classes multinomial model with RandomClass, it produces



(a) $k = 3$ (b) $k = 5$ (c) $k = 10$

Fig. 4: Evaluation result of k-ranked items list from precision metrics with different number of classes and model, where red line is at 79 classes



(a) $k = 3$ (b) $k = 5$ (c) $k = 10$

Fig. 5: Evaluation result of k-ranked items list from nDCG metrics with different number of classes and model, where red line is at 79 classes

high precision but low nDCG, that means there are high-rating items in obtained k-ranked items list, but in incorrect order. While 100-classes multinomial model with RandomAll produces low precision but high nDCG, which means k-ranked items list is ordered correctly, but it contains low-rating score items.

As for Poisson model with RandomClass, it is very interesting that accuracy of model doesn't change, until number of classes goes beyond 50 classes. Unfortunately, log-likelihood doesn't seem to be relevant to trends of model's accuracy aginst number of classes, but still relevant to log-likelihood plotted in **Fig. 6** that the most accurate model has the lowest value of $|Log\text{-}Likelihood|$. We will find out the reason behind this through learned parameters of model in future work.

Finally, multinomial model with RandomClass is compared with Funk's SVD in 25 latent factors. Parameters of Funk's SVD are learned via stochastic gradient descent for 1000 iterations, which makes those parameters optimized in most cases. We experiment with various value of learning rate and regularization value for Funk's SVD. Learning rate is tuned for the largest value that doesn't go out of optimal point, which is 0.005. **Fig. 7** shows that multinomial model with RandomClass has more accuracy than Funk's SVD significantly. That is unpersonalized recommendation by using item characteristics, and item age data provides more accuracy than personalized recommendation.

In the future work, we would like to compare this model with timeSVD++, which results time-personalized recommendation.

5. Conclusion

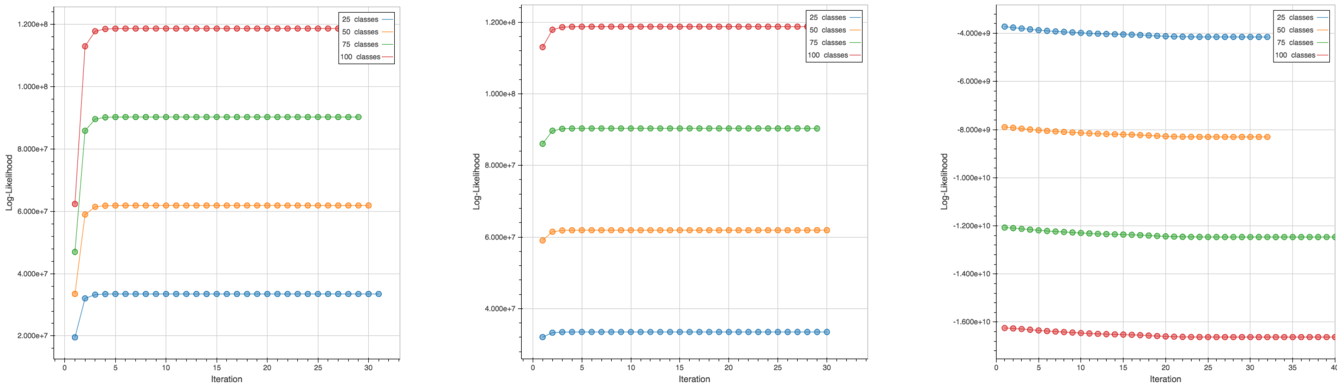
In this work, we present a possible way to predict future trend by using other type of time data, item age, instead of normal time stamp. Simple non-personalized latent probabilistic model is built to prove that item age can be used in place of normal time stamp. The experimental results show that the model provides significantly greater accuracy than traditional latent model, Funk's SVD. That is time non-personalized recommendation is better than personalized recommendation. However, many mysteries still are remained and uncleared. In future work, we would like to do more detailed experiment, model with various cardinality of class. Finally, the model is planned to be personalized to able to compare with timeSVD++.

References

[1] Koren, Y., Bell, R. and Volinsky, C.:Matrix Factorization Techniques for Recommender Systems, *Computer*, Vol. 42, No. 8, pp. 30–37 (online),
DOI: 10.1109/MC.2009.263 (2009).

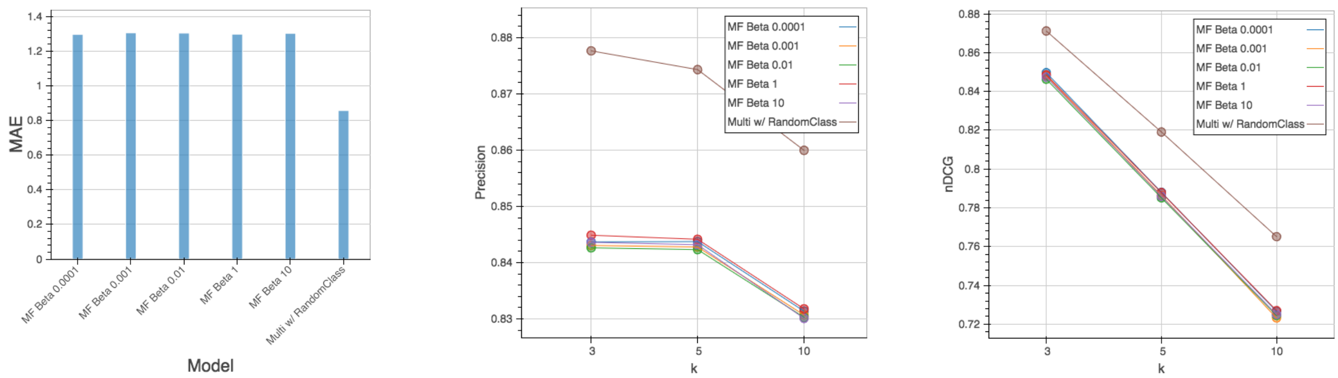
[2] Samatthiyadikun, P., Takasu, A. and Maneeroj S.: Multi- criteria collaborative filtering by bayesian model-based user profiling, *Proc. Intl. Conf. Information Reuse and Integration (IRI)*, pp. 124–131 (online) doi=10.1109/IRI.2012.6303000 (2012).

[3] Koren, Y.: Collaborative Filtering with Temporal Dynamics, *Proc.*



(a) Multinomial model with RandomAll (b) Multinomial model with RandomClass (c) Poisson model with RandomClass

Fig. 6: Convergence of various models with different settings and number of classes.



(a) Mean Absolute Error (MAE) (b) Precision (c) Normalized Discounted Cumulative Gain

Fig. 7: Comparison between evaluation result of Multinomial model with RandomClass against Funk's SVD

Intl. Conf. Knowledge Discovery and Data Mining (KDD), pp. 447–456 (online) doi=10.1145/1557019.1557072 (2009).

[4] Peralta, V.: Extraction and Integration of MovieLens and IMDB Data, Technical Report, APDM project, Laboratory PRISM, University of Versailles, Versailles, (2007).

[5] Internet Movie Database (IMDb), available from ⟨http://grouplens.org/datasets/movielens/⟩ (accessed 2015-3-5).

[6] GroupLens Research Group: MovieLens 10M Data Set, available from ⟨http://www.imdb.com/interfaces/⟩ (accessed 2015-3-5).