Topic: Time aware Recommender system

IT492: Recommendation Systems

Kevin Jadiya	:	202111010
Rahul Vansh	:	202111035
Juned Mansuri	:	202111048

Introduction & Motivation:

- Traditional RecSys uses User-Item matrix to give recommendations.
- If we want more powerful RecSys than traditional one then considering the context in which users express their preferences has been proven very valuable for increasing the performance of recommendations.
- Among existing contextual dimensions, time information can be considered as one of the most useful ones.
- It facilitates tracking the evolution of user preferences. for example, to identify periodicity in user habits and interests.(ex. youtube)
- It also leads to significant improvements on recommendation accuracy.

Traditional Rec : $U \times I \to R$ CARS : $U \times I \times C \to R$

- Presence of a context C requires us to learn the mapping from the $U \times I \times C$ to the ratings. TARS is nothing but special case of CARS where our context C is time.
- Time can be treated as a modeling variable by expressing the predicted ratings as a function of time.
- The parameters of this function can be learned by minimizing the squared error of the predicted ratings.
- An example of such a model is time-SVD++.
- This approach is considered one of the state-of-the-art techniques for temporal prediction. The main advantage of this approach is that it can capture future trends, which are not easily captured by other models.

Temporal collaborative filtering

Temporal information can be used in one of three ways in order to improve the effectiveness of prediction:

1. ***Recency-based models***: Some models consider recent ratings >>> older ratings. In these cases, **window-based** and **decay-based** models are used for more accurate prediction. The basic idea is that the recent ratings are given more importance within the collaborative filtering model.

2. ***Periodic context-based models***: In periodic context-based models, the specific property of a period, such as the time at the level of specificity of the hour, day, week, month, or season,is used to perform the recommendation(i.e. Recommendation in online shopping sites during Diwali will be different than other days)

3. ***Models that explicitly use time as an independent variable:*** A recent approach, referred to as **time-SVD++**, uses time as an independent variable within the modeling process. Such an approach uses more refined user-specific and item-specific trends to handle local temporal variations. Generally, such models are more sophisticated than recency-based models because they include an element of forecasting.

- We’ll discuss more about these models in following slides.

1) Recency based models

- In recency-based models, recent ratings are given greater importance than older ones.
- There are two types of method in recency based model one is **decay-based methods** another is **window-based methods.**
- In decay method it uses decay function for giving less importance to older ratings.
- Window-based methods are the extension of decay-based methods in which the older ratings are completely disregarded and we give full weightage to recent ratings.

Decay-Based Methods

- **T**uj is a time-stamp associated with every rating of item **j** given by user **u** Therefore, the number of observed value of **t**uj exactly same of total number of ratings.
- It is assumes that all recommendations made at future time referred as target time Tf.
- Finding out weight of the rating at target time Wuj(Tf) called decay function,which is commonly used, is the exponential function.
- **λ** is the decay rate,it is regulated parameter given by user.
- Larger value of **λ** de-emphasize older ratings to a greater degree.

$$w_{uj}(t_f) = \exp[-\lambda(t_f - t_{uj})]$$

Use of Decay function

- We can use the Decay function in collaborative filtering technique instead of normal weightage function we use decay function for biasing the for recent ratings.
- We can use this function in both user-based and item-based neighborhoods method.
- The optimal value of λ can be learned using cross-validation methods.

$$\hat{r}_{ui} = h^{-1}\left(\frac{\sum_{v \in N_{ui}} w_{uv} . h(r_{vi})}{\sum_{v \in N_{ui}} |w_{uv}|}\right)$$

Discount Factor

- After using Decay function a slightly more refined model in which an item-based neighborhood method is used for collaborative filtering.
- Discount factor is not a simple exponential decay function.
- Each item is assigned a discount factor by estimating its expected future error and then assigning a weight that is inversely proportional to this error.
- Qj(u) is specifies ratings of peer items of target item j given by user u.
- Dui is discounted factor need to determine for all the item in Qj(u) to modify final prediction function.

$$\hat{r}_{uj} = \frac{\sum_{i \in Q_j(u)} \mathrm{Sim}(i,j) \cdot D_{ui} \cdot r_{ui}}{\sum_{i \in Q_j(u)} |\mathrm{Sim}(i,j)| \cdot D_{ui}}$$

How is each discount factor D(ui) computed?

- It is achieved by taking normalized differences between rating of item i Rui and Oui is the average rating item i’s peer items.peer items can be found using item-item similarity.
- Dui ∈ (0,1) and α is a tuning parameter, which can be chosen using cross-validation.
- Here,Rmax and Rmin are the maximum and minimum values on the scale of ratings.

$$D_{ui} = \left(1 - \frac{|O_{ui} - r_{ui}|}{r_{max} - r_{min}}\right)^{\alpha}$$

Use of Discount Factor

- To tackle the problem of concept drift, we assign the weights for items by estimating their expected prediction error on the future purchase preferences.
- We assume that the most recent rating is closest to the future preference.
- Consequently, the weight of the items can be approximated by computing the deviation of their ratings from the most recent rating.

Window-Based Methods

- In window-based methods, ratings that are older than a particular time are pruned from consideration.
- these methods can also be viewed as (discrete) special cases of decay-based methods.
- There are several ways in which windows can be modeled:
- If the difference between the target time T(f) and the rating time T(ij) is larger than a particular threshold, then the rating is dropped.Which means older ratings are completely discarded.
- In another case, the windows can be set in a domain and item-specific way. For example, the method uses not only the most recent ratings, but also the ratings from the same month in the previous years.it is combination of time and periodic model.

2) Handling periodic context:

- Periodic context is designed to handle cases in which the time dimension may refer to a specific period in time, such as hour of the day, day of the week, season, or the time intervals in the specific periodic events (e.g., Christmas)
- In this case, the target recommendation time defines the context in which the recommendation is made. This context can sometimes play an all-important role in the recommendation process.
- For example, for a supermarket, the recommendations targeted for the weekend before the Thanksgiving holiday would be very different from those targeted during other times.
- Several natural ways of handling periodic context are discussed in the following slides

Pre-Filtering and Post-Filtering (in context of TARS):

- In ***pre-filtering***, a significant part of the ratings data are removed that are not relevant to the specific target time (i.e., context) within which the recommendation is being performed or executed.
- For example, one might use only the ratings from the last two weeks before Thanksgiving of each year, in order to construct the models for making recommendations on the weekend before Thanksgiving.
- In ***post-filtering***, the recommendations are adjusted based on the context, after a non contextual method has been used to generate the recommendation on all the data. Therefore,the basic approach of post-filtering uses the following two steps:
  1. Generate the recommendations using a conventional collaborative filtering approach on all data, while ignoring the temporal context.
  2. Adjust the generated recommendation list with the use of temporal context as a post processing step. Either the ranks of the recommended list may be adjusted based on context, or the list may be trimmed of contextually irrelevant items.
- After the recommendation lists have been generated, the ranking is either re-adjusted by weighting with a contextual relevance weight, or the items with very low contextual relevance weights are removed

Direct Incorporation of Temporal Context

- It is possible to directly modify existing models such as neighborhood methods in order to incorporate temporal context.
- It works directly with the 3-dimensional representation corresponding to user, item,and context.
- For example, in the user-based neighborhood scheme, one might modify the distance computation between two users with the use of contextual attributes.
- If two users give the same rating to an item during weekends, they should be considered more similar than a pair of users that have specified these ratings in different temporal contexts.
- By using the modified distance computation, the context is automatically incorporated into the recommendation process.

User Id	Item 1	Item 2	Item 3	Time
Alice	5	4	3	weekdays
Bob	5	4	3	weekend
Katty	5	4	3	weekend

3) Modeling ratings as function of time

- In these model based methods, the ratings are modeled as a function of time.
- These methods can intelligently separate long-term trends from transient and noisy trends.
- These distinctions are important for making such temporal models robust.

The Time SVD++ Model:

- The main change to the original latent factor model is that, significant part of weightage is given to user bias and item bias, rather than any specific personalized preferences of users for items
- if a movie is a box-office hit, likely to have large positive values of p_j, which indicates that the relevant user is also more likely to appreciate it.

$$\hat{r}_{ij} = \sum_{s=1}^{k} u_{is} \cdot v_{js}$$

$$\hat{r}_{ij} = o_i + p_j + \sum_{s=1}^{k} u_{is} \cdot v_{js}$$

o_i : user bias
p_j : item bias

- This basic bias-based model is further enhanced with the notion of implicit feedback variables Y =[y_{ij}] for each user-item pair, which encode the propensity of each factor-item combination to contribute to implicit feedback.
- Ex. users express their interests by buying items.

$$\begin{pmatrix} 1 & -1 & 1 & ? & 1 & 2 \\ ? & ? & -2 & ? & -1 & ? \\ 0 & ? & ? & ? & ? & ? \\ -1 & 2 & -2 & ? & ? & ? \end{pmatrix} \Rightarrow \begin{pmatrix} 1/\sqrt{5} & 1/\sqrt{5} & 1/\sqrt{5} & 0 & 1/\sqrt{5} & 1/\sqrt{5} \\ 0 & 0 & 1/\sqrt{2} & 0 & 1/\sqrt{2} & 0 \\ 1/\sqrt{1} & 0 & 0 & 0 & 0 & 0 \\ 1/\sqrt{3} & 1/\sqrt{3} & 1/\sqrt{3} & 0 & 0 & 0 \end{pmatrix}$$

- Here, if y_{ij} is large, then it means that in the act of rating item i, user’s preference is more leaning towards jth latent component
- So in our case, in below eqn, it adjusts the s^th latent factor of user i by adding implicit feedback term to user factor u_{is}

$$\hat{r}_{ij} = o_i + p_j + \sum_{s=1}^{k} \left(u_{is} + \sum_{h \in I_i} \frac{y_{hs}}{\sqrt{|I_i|}} \right) \cdot v_{js}$$

- In time-SVD++ model, model parameters are function of time. Therefore, user biases , item biases, and the user factors are functions of time.
- Item variables and the implicit feedback variables have not been temporally parameterized, and are assumed to stay constant with time.
- Although its possible to temporally parameterize these variables as well but SVD++ model chooses a simplified approach in which each temporal parametrization can be justified with intuitive arguments.

$$\hat{r}_{ij}(t) = o_i(t) + p_j(t) + \sum_{s=1}^{k} \left(u_{is}(t) + \sum_{h \in I_i} \frac{y_{hs}}{\sqrt{|I_i|}} \right) \cdot v_{js}$$

- The idea here is that the average rating of the user can **vary significantly from that at the mean date v_i of rating**. The user might be rating most items positively now, but her mean rating might decrease in a couple of years.
- User bias is constant part, a time-dependent part, and transient noise.
- This portion of variability is captured by $\alpha_i \cdot \mathrm{dev}_i(t)$. In real life, transient mood changes from day to day might lead to sudden and unpredictable spikes or dips in ratings.
- A user might rate all items poorly when she/he is having a bad day. Such variations are captured by noise parameter epsilon.

$$\mathrm{dev}_i(t) = \mathrm{sign}(t - \nu_i) \cdot |t - \nu_i|^\beta$$

$$o_i(t) = K_i + \alpha_i \cdot \mathrm{dev}_i(t) + \epsilon_{it}$$

- **Popularity of an item can vary significantly with time**, but it shows a high level of continuity and stability over shorter periods.
- For example, a box-office hit will have an approximately stable distribution of ratings in the short period after release, but it might be rated very differently after a couple of years.
- Therefore, the time horizon can be split into bins, and the ratings belonging to a particular bin have the same bias.
- Netflix movie ratings, a total of 30 bins were used, and each bin represented about 10 consecutive weeks of ratings. Note that the value of $p_j(t)$ will be different for different ratings, depending on when they were rated.

$$p_j(t) = C_j + \text{Offset}_{j,Bin(t)}$$

- The key thing is, the **affinity of users towards various concepts**. For example, a younger user who used to appreciate animated movies more today he/she might be interested in action movies more.
- Similar approach to user biases is used for modeling the temporal change in the user factors.

$$u_{is}(t) = K'_{is} + \alpha'_{is} \cdot \mathrm{dev}_i(t) + \epsilon'_{ist}$$

- We assume that the observation time of all the ratings is known.
- if S contains the set of user-item pairs, then one must solve the following optimization problem

$$\text{Minimize } J = \frac{1}{2} \sum_{(i,j)\in S} [r_{ij} - \hat{r}_{ij}(t_{ij})]^2 + \lambda \cdot (\text{Regularization Term})$$

$$\hat{r}_{ij}(t) = o_i(t) + p_j(t) + \sum_{s=1}^{k} \left(u_{is}(t) + \sum_{h\in I_i} \frac{y_{hs}}{\sqrt{|I_i|}} \right) \cdot v_{js}$$

Applications of TARS:

- In e-commerce when the winter season starts it recommends winter clothings like jackets and gloves.
- In netflix ,If the new movie released recently and if it has older version then netflix start recommending that old version movies.
- The TARS widely used in news article recommendations system by recommending most recent articles.
- In spotify it recommend peaceful music in the morning.
- In food application when it’s time to take dinner then it will notify at that time.

Thank You