Collaborative Filtering with Temporal Dynamics

Kyusu Ahn

2021. 05. 03.

ABSTRACT

- **Temporal dynamics**
  - Customer preferences for products are *drifting over time*.
  - Product perception and popularity are constantly *changing* as new selection emerges.

  ➔ Modeling **temporal dynamics** is required

- **Unique challenges**
  - Many different characteristics are shifting simultaneously and influence each other.
  - Different with concept drift explorations, where mostly a single concept is tracked.

  ➔ Classical *time-window* or *instance decay* approaches cannot work, as they lose too much signal when discarding data instances.

  ➔ A more sensitive approach is required, which can make better distinctions between transient effects and long-term patterns.

- **The paradigm we offer**
  - Tracking the **time changing** behavior throughout the **life span** of the data to **exploit the relevant components** of all data instances, while **discarding irrelevant components**.

Introduction

- **Concept drift**
  - Data is changing over time, and up to date modeling should be continuously updated to reflect its present nature.
  - Need to find the right **balance** between **discounting temporary effects** that have very low impact on future behavior and **capturing longer-term trends** that reflect the inherent nature of the data.

- **Global concept drift**
  - Traditional studies on concept drift

    ex) seasonal changes, or specific holidays; All those changes influence the whole population.

- **Localized factors**
  - Each occurs at a **distinct time frame** and is driven towards a **different direction.**

    ex) A change in the family structure

    ex) Individuals gradually change their taste in movies and music.

3. TRACKING DRIFTING CUSTOMER PREFERENCES

- Complicated form of concept drift
  - Requires the learning algorithm to keep track of **multiple changing concepts**

- Tsymbal's three approaches for concept drift [22]

1-1) Instance selection

Discards instances that are less relevant to the current state

1-2) Time window approaches

Instance selection의 변형, only recent instances are considered

❖ **Disadvantage**

+ Giving the same significance to all instances within the considered time window, while completely discarding all other instances.

[22] A. Tsymbal. The problem of concept drift: Definitions and related work. Technical Report TCD-CS-2004-15, Trinity College Dublin, 2004.

3. TRACKING DRIFTING CUSTOMER PREFERENCES

- Tsymbal's three approaches for concept drift [22]

2) instance weighting

+ Instances are **weighted** based on their **estimated relevance**

+ **A time decay function** **under-weights** instances as they occur deeper into the past.

❖ **Experiment**

Trying different exponential time decay rates on both neighborhood and factor models

❖ **Results**

+ Prediction quality improves as moderating time decay, reaching **best quality without decay**

+ Much of the **old preferences still persist**

+ Help in establishing useful **cross-user** or **cross-product** patterns in the data

+ Underweighting past actions **loses too much signal**

[22] A. Tsymbal. The problem of concept drift: Definitions and related work. Technical Report TCD-CS-2004-15, Trinity College Dublin, 2004.

3. TRACKING DRIFTING CUSTOMER PREFERENCES

- Tsymbal's three approaches for concept drift [22]

3) Ensemble learning

+ Having multiple predictor that together produce the final outcome.

+ Those predictors are **weighted** by their perceived **relevance** to the **present time point**

+ Capturing a **collective signal** requires building a **single model** encompassing **all users and items together**.

3. TRACKING DRIFTING CUSTOMER PREFERENCES

- Guidelines for **modeling drifting user preferences**
  1. Models should explain user behavior along the **full extent of the time period**, not only the present behavior.
  2. **Multiple changing concepts** should be captured.
     - Some are **user-dependent** and some are **item-dependent**.
     - Some are **gradual** while others are **sudden**.
  3. While we need to model separate drifting “concepts” or preferences per user and/or item, it is essential to **combine** all those concepts **within a single framework**.
  4. In general, do **not** try to **extrapolate future temporal dynamics**.

4. TIME-AWARE FACTOR MODEL
- *A Factor Model*

- **Matrix factorization models**
  - Each user u vector = $p_u \in \mathbb{R}^f$, each item i vector = $q_i \in \mathbb{R}^f$
  - Ratings are modeled as inner products: $\hat{r}_{ui} = q_i^T p_u$
- **Baseline predictors**
  - A **pure factor model** captures the **interaction between users and items**.
  - However, much of the **observed rating values** are due to effects associated with either **users or items, independently of their interaction**.

    ex) some users give higher ratings than others some items receive higher ratings than others.
  - **Encapsulate those effects**, which do not involve user-item interaction, **within the baseline predictors**.
  - ***Baseline predictors*** capture much of the temporal dynamics within the data.

<Baseline predictor>

$$b_{ui} = \boxed{\mu + b_u + b_i}$$

b_{ui} : a baseline predictor for an unknown rating r_{ui}
μ : the overall average rating
b_u : user bias b_i : item bias

<Extended factor model>

$$\hat{r}_{ui} = \boxed{\mu + b_u + b_i} + q_i^T p_u$$

Presenter: Kyusu Ahn

4. TIME-AWARE FACTOR MODEL
- A Factor Model

- **SVD++**
  - Offer superior accuracy and account for the **implicit information** (regardless of their rating value)

$$\hat{r}_{ui} = \mu + b_i + b_u + q_i^T \left(p_u + |\mathrm{R}(u)|^{-\frac{1}{2}} \sum_{j \in \mathrm{R}(u)} y_j \right)$$

R(u) : the set containing the items rated by user *u*

$\boldsymbol{y_j}$ **: item factors characterizing users** based on the set of items they rated

  - Decomposition of a rating into distinct portions allows to treat **different temporal aspects in separation**.
    1. User biases (bu) change over time
    2. Item biases (bi) change over time
    3. User preferences (pu) change over time

❖ As **items**, unlike humans, are **static** in their nature,

not expect a temporal variation of item characteristics (qi)

4. TIME-AWARE FACTOR MODEL
- Time changing baseline predictors

- **Time sensitive baseline predictor**
  - Much of the temporal variability is included within the baseline predictors

- **Temporal effects** within the baseline predictors
  - **Item's popularity is changing over time** *ex) the appearance of an actor in a new movie*
  - **Users change their baseline ratings over time**

    ex) a user who tended to rate an average movie "4 stars", may now rate such a movie "3 stars"

    ➔ **Bias bi and bu → function that changes over time**

    $$b_{ui}(t) = \mu + b_u(t) + b_i(t)$$

    $b_{ui}(t)$ represents the baseline estimate for u's rating of i at day

  - *Temporal effects that span extended periods of time* VS *more transient effects*

    ex) In the movie rating case, movie likeability does not fluctuate on a daily basis, but on extended periods.

  - On the other hand, user effects can change on a daily basis.

    ➔ This requires ***finer time resolution*** when modeling ***user-biases*** compared to item-biases.

4. TIME-AWARE FACTOR MODEL
- Time changing baseline predictors

- **Time-changing item biases - bi(t)**
  - Do not need finest resolution
  - Split the item biases into time-based bins
  - How to split the timeline into bins?

    balance between *achieving finer resolution (smaller bins)* and *having enough ratings per bin (larger bins)*
  - In our implementation each bin corresponds to ten consecutive weeks of data, leading to an overall number of 30 bins spanning all days in the dataset
  - A day t is associated with an integer Bin(t) (a number between 1 and 30 in our data)

<Movie bias>

$$b_i(t) = \underbrace{b_i}_{\text{stationary part}} + \underbrace{b_{i,\mathrm{Bin}(t)}}_{\text{time changing part}}$$

4. TIME-AWARE FACTOR MODEL
- Time changing baseline predictors

- **Time-changing user biases - bu(t)**
  - Finer resolution for users to detect very short lived temporal effects
  - Capture a possible **gradual drift** of user bias
  - **Time-linear model**
    - Uses a linear function to capture a possible gradual drift of user bias
    - ***Time derivation***: $\mathrm{dev}_u(t) = \mathrm{sign}(t - t_u) \cdot |t - t_u|^\beta$
    - ***Time dependent user-bias (1)***: $\boxed{b_u^{(1)}(t) = b_u + \alpha_u \cdot \mathrm{dev}_u(t)}$

      ➔ requires learning two parameters per user: **b**u and **α**u

t_u : for each user u, the mean date of rating
$|t - t_u|$: the time distance (e.g., number of days) between dates t and t_u
set the value of β by cross validation; in our implementation β = 0.4

  - **Spline-based model**
    - A more **flexible** parameterization is offered by **splines**
    - Designate k_u time points – $\{t_1^u, \ldots, t_{k_u}^u\}$ – spaced uniformly across the dates of u's ratings as kernels.
    - ***Time dependent user-bias (2)***: $\boxed{b_u^{(2)}(t) = b_u + \frac{\sum_{l=1}^{k_u} e^{-\gamma|t-t_l^u|} b_{t_l}^u}{\sum_{l=1}^{k_u} e^{-\gamma|t-t_l^u|}}}$

b_{tl}^u : associated with the control points (or, kernels), automatically learnt from the data

    - User bias is formed as a time-weighted combination of those parameters

4. TIME-AWARE FACTOR MODEL
- Time changing baseline predictors

- **Time-changing user biases - bu(t)**
  - **Sudden drifts** emerging as “spikes” associated with **a single day or session**

    ex) multiple ratings a user gives in a single day, tend to concentrate around a single value
  - The effect does **not span more than a single day**
  - bu,t : the day-specific variability

    it serves as an **additive component** within the previously described schemes

  - **Time-linear model becomes:**

$$b_u^{(3)}(t) = b_u + \alpha_u \cdot \mathrm{dev}_u(t) + b_{u,t}$$

  - **Spline-based model becomes:**

$$b_u^{(4)}(t) = b_u + \frac{\sum_{l=1}^{k_u} e^{-\gamma|t-t_l^u|} b_{t_l}^u}{\sum_{l=1}^{k_u} e^{-\gamma|t-t_l^u|}} + b_{u,t}$$

4. TIME-AWARE FACTOR MODEL
- Time changing baseline predictors

- Compare the ability of various suggested baseline predictors

model	static	mov	linear	spline	linear+	spline+
RMSE	.9799	.9771	.9731	.9714	.9605	.9603

Table 1: Comparing baseline predictors capturing main movie and user effects. As temporal modeling becomes more accurate, prediction accuracy improves (lowering RMSE).

- *static* no temporal effects: $b_{ui}(t) = \mu + b_u + b_i$.
- *mov* accounting only to movie-related temporal effects: $b_{ui}(t) = \mu + b_u + b_i + b_{i,\mathrm{Bin}(t)}$.
- *linear* linear modeling of user biases: $b_{ui}(t) = \mu + b_u + \alpha_u \cdot \mathrm{dev}_u(t) + b_i + b_{i,\mathrm{Bin}(t)}$.
- *spline* spline modeling of user biases: $b_{ui}(t) = \mu + b_u + \frac{\sum_{l=1}^{k_u} e^{-\gamma|t-t_l^u|} b_{t_l}^u}{\sum_{l=1}^{k_u} e^{-\gamma|t-t_l^u|}} + b_i + b_{i,\mathrm{Bin}(t)}$.
- *linear+* linear modeling of user biases and single day effect: $b_{ui}(t) = \mu + b_u + \alpha_u \cdot \mathrm{dev}_u(t) + b_{u,t} + b_i + b_{i,\mathrm{Bin}(t)}$.
- *spline+* spline modeling of user biases and single day effect: $$b_{ui}(t) = \mu + b_u + \frac{\sum_{l=1}^{k_u} e^{-\gamma|t-d_l|} b_{t_l}^u}{\sum_{l=1}^{k_u} e^{-\gamma|t-t_l^u|}} + b_{u,t} + b_i + b_{i,\mathrm{Bin}(t)}.$$

4. TIME-AWARE FACTOR MODEL
- Time changing baseline predictors

- **Another temporal effect : changing scale of user ratings**
  - Consider **item bias $b_i(t)$** is a **user-dependent** measure

    ex) Different users employ different rating scales, and a single user can change his rating scale over time.
  - The changing scale of user ratings affects item bias
  - $c_u(t)$: time-dependent scaling feature
  - **linear+ becomes**:

    $$b_{ui}(t) = \mu + b_u + \alpha_u \cdot \mathrm{dev}_u(t) + b_{u,t} + (b_i + b_{i,\mathrm{Bin}(t)}) \cdot c_u(t)$$
  - Ways to implement $b_u(t)$ would be valid for implementing $c_u(t)$ as well

    $$c_u(t) = \underbrace{c_u}_{\textit{Stable part}} + \underbrace{c_{u,t}}_{\textit{Day-specific variability}}$$
  - Adding the multiplicative factor $cu(t)$ to the baseline predictor **lowers RMSE to 0.9555**.

    (**lower than linear+'s RMSE**)

4. TIME-AWARE FACTOR MODEL
- Time changing factor model

- **Temporal dynamics affect the interaction between users and items.**
  - Users change their preferences over time

    ex) a fan of the “psychological thrillers” genre may become a fan of “crime dramas” a year later.
  - This effect is modeled by taking the **user factors (the vector p_u) as a function of time**.
  - We need to model those changes at the **very fine level of a daily basis**,

    while facing the built-in **scarcity of user ratings**.

  - **User factor($p_{uk}(t)$)**

$$p_{uk}(t) = p_{uk} + \alpha_{uk} \cdot \mathrm{dev}_u(t) + p_{uk,t} \quad k = 1, \ldots, f$$

f : factorization dimensions

each component of the user preferences $p_u(t)^T = (p_{u1}(t), \ldots, p_{uf}(t))$

p_{uk} : the stationary portion of the factor

$\alpha_{uk} \cdot \mathrm{dev}_u(t)$: changes linearly over time

$p_{uk,t}$: day-specific variability

4. TIME-AWARE FACTOR MODEL
- Time changing factor model

- **Compare results of three algorithms; SVD, SVD++ and timeSVD++**
  - All methods benefit from a growing number of factor dimensions(f)
  - The improvement delivered by timeSVD++ over SVD++ is consistently more significant than the improvement SVD++ achieves over SVD.
  - Importance of properly addressing **temporal effects**.
  - TimeSVD++ model of dimension 10 is already more accurate than an SVD model of dimension 200.
  - TimeSVD++ model of dimension 20 is enough to outperform an SVD++ model of dimension 200

Model	f=10	f=20	f=50	f=100	f=200
SVD	.9140	.9074	.9046	.9025	.9009
SVD++	.9131	.9032	.8952	.8924	.8911
timeSVD++	.8971	.8891	.8824	.8805	.8799

Table 2: Comparison of three factor models: prediction accuracy is measured by RMSE (lower is better) for varying factor dimensionality (f). For all models accuracy improves with growing number of dimensions. Most significant accuracy gains are achieved by addressing the temporal dynamics in the data through the timeSVD++ model.

- SVD

$$\hat{r}_{ui} = q_i^T p_u$$

- SVD++

$$\hat{r}_{ui} = \mu + b_i + b_u + q_i^T \left(p_u + |\mathrm{R}(u)|^{-\frac{1}{2}} \sum_{j \in \mathrm{R}(u)} y_j \right)$$

- *time*SVD++

$$\hat{r}_{ui}(t) = \mu + b_i(t) + b_u(t) + q_i^T \left(p_u(t) + |\mathrm{R}(u)|^{-\frac{1}{2}} \sum_{j \in \mathrm{R}(u)} y_j \right)$$

5. TEMPORAL DYNAMICS AT NEIGHBORHOODMODELS

- **Item-item neighborhood model**
  - The most common approach to CF is based on neighborhood models.
  - Static model, without temporal dynamics

$$\hat{r}_{ui} = \boxed{\mu + b_i + b_u} + |\mathrm{R}(u)|^{-\frac{1}{2}} \sum_{j \in \mathrm{R}(u)} (r_{uj} - b_{uj}) w_{ij} + c_{ij}$$

    - It was proven greatly beneficial to use two sets of item-item weights(w_{ij} and c_{ij}):
    - w_{ij} is related to the values of the ratings
    - c_{ij} disregards the rating value, considering only which items were rated
    - These weights are automatically learnt from the data together with the biases bi and bu
  - To address **temporal dynamics**, two components should be considered separately
    1. **Baseline predictor** portion explains most of the observed signal.
    2. **User-item interaction** portion captures the more informative signal.

5. TEMPORAL DYNAMICS AT NEIGHBORHOODMODELS

- **Item-item model with temporal dynamics**

$$\hat{r}_{ui}(t) = \boxed{\mu + b_i(t) + b_u(t)} + \boxed{|\mathrm{R}(u)|^{-\frac{1}{2}} \sum_{(j,t_j)\in \mathrm{R}(u)} e^{-\beta_u \cdot |t-t_j|}((r_{uj} - b_{uj})w_{ij} + c_{ij})}$$

$$\boxed{\begin{aligned} b_i(t) &= b_i + b_{i,\mathrm{Bin}(t)} \\ b_u(t) &= b_u + \alpha_u \cdot \mathrm{dev}_u(t) + b_{u,t} \end{aligned}}$$

- Item-item weights (w_{ij} and c_{ij}) reflect inherent item characteristics, not expected to drift over time.
- Learning process should make **item-item weights** capture **unbiased long term values**

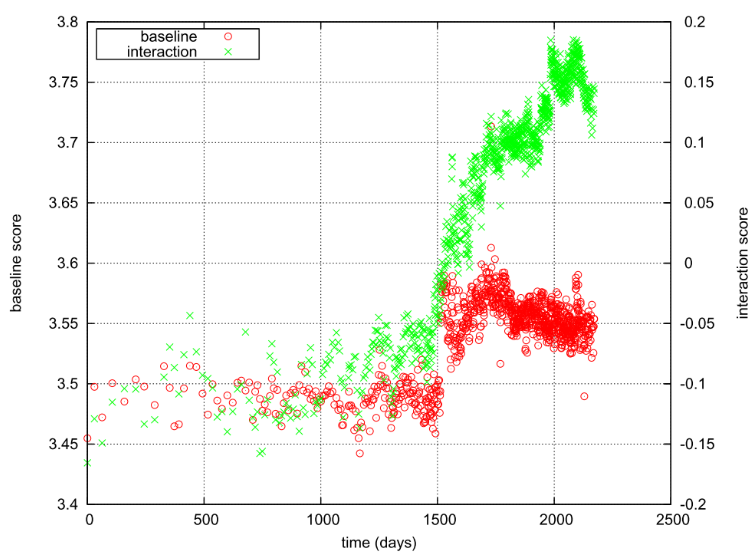
  Ex) a user rating both items i and j high in a short time period, is a good indicator for relating them, thereby pushing higher the value of w_{ij}
- Those considerations are pretty much **user-dependent** – some users are more consistent than others
- Our goal : distill accurate values for the item-item weights, despite the interfering temporal effects
- Properly considering temporal dynamics improves the accuracy of the neighborhood model within the movie ratings dataset.
- This result is even better than using hybrid approaches such as applying a neighborhood approach on residuals of other algorithms.

6. AN EXPLORATORY STUDY

- **First effect : a sudden rise in the average movie rating**
  - Interaction part of the models
    - Users are increasingly rating movies that are more suitable for their own taste

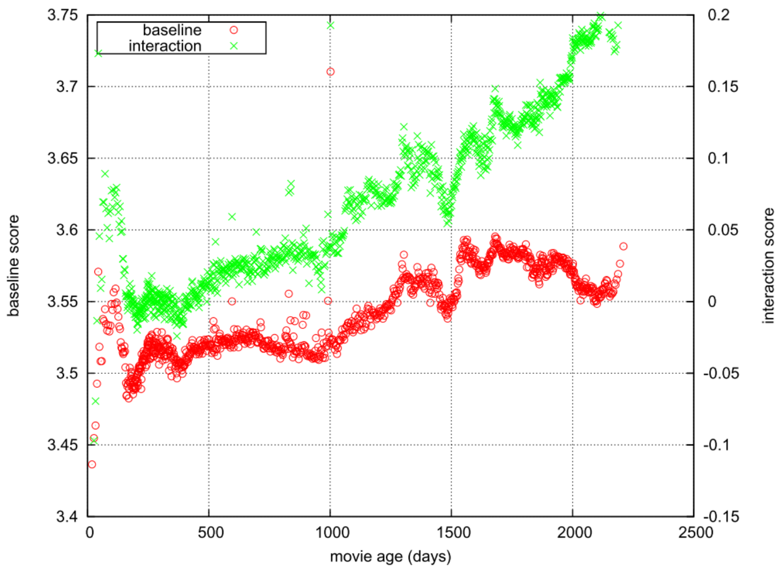
      $q_i^T \left(p_u(t) + |\mathrm{R}(u)|^{-\frac{1}{2}} \sum_{j \in \mathrm{R}(u)} y_j \right)$ for the timeSVD++ model
  - Baseline predictor portion of the model
    - General biases that have nothing to do with the matching of users to movies

      $\mu + b_i(t) + b_u(t)$



6. AN EXPLORATORY STUDY

- **Second effect : higher ratings as movies become older**
  - Older movies are getting rated by users better matching them.
    - Captured by that the interaction part of the model is rising with movies' age.
  - Older movies are just inherently better than newer ones.
    - Captured by the baseline part of the model



8. CONCLUSIONS

- **Unique challenges**
  - Each user and product potentially goes through a distinct series of changes in their characteristics.
  - Need to model all those changes within a single model
  - A mere decay of older instances or usage of multiple separate models lose too much signal, thus degrading prediction accuracy
- **The solution we adopted**
  - Modeling the temporal dynamics along the whole time period to separate transient factors from lasting ones.
  - Applied this methodology with factorization model and neighborhood model.
- **Factorization model**
  - Modeling the way user and product characteristics change over time, in order to distill longer term trends from noisy patterns
- **Item-item neighborhood model**
  - Showed how the more fundamental relations among items can be revealed by learning how influence between two items rated by a user decays over time

Thank You

Presenter: Kyusu Ahn