Collaborative Filtering with Temporal Dynamics

Prepared by Thomas Perry

School of Computational Science & Engineering
College of Computing
Georgia Institute of Technology

Introduction

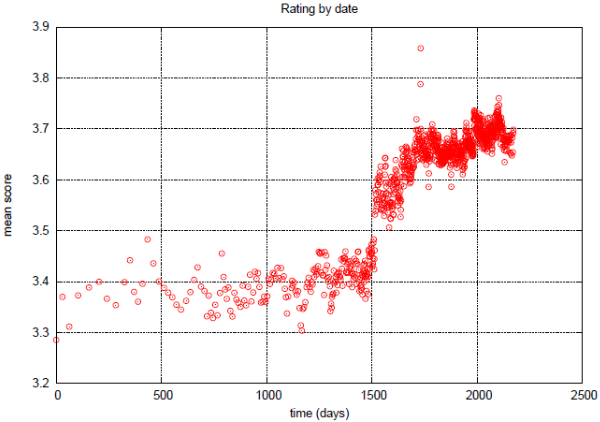
Customer preferences for products drift over time. Preference drift can be sudden or gradual.

Possible causes of drift

- New product or services change.
- Localized factors change (e.g. a change in family structure)
- Seasonal trends

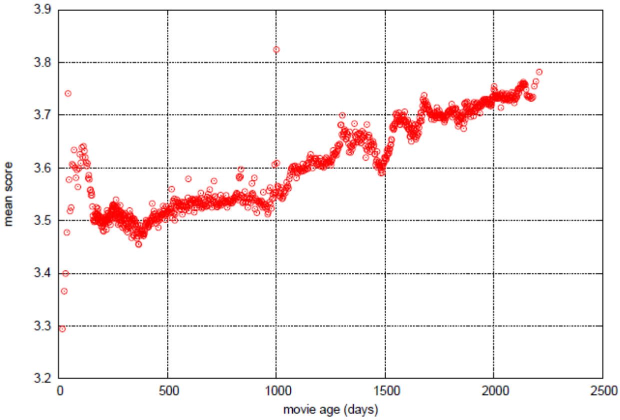
Goal: To balance between discounting temporary effects that have very low impact on future behavior, while capturing longer-term trends that reflect the nature of the customer.

Empirical Examples of Preference Drift



Netflix Data: The average movie rating made a sudden jump in 2004.

Empirical Examples of Preference Drift



Netflix Data: Ratings tend to increase with the movee age at the time of the rating. Here, movie age is measured by the time span since its first rating event within the data set.

Three Approaches to Concept Drift

Instance selection: Discards instances of less relevance. A time variant, only recent instances are considered.

Instance weighting: Instances are weighted based on their estimated relecance. Often, a time decay function is used, under-weighting inances as they occur deeper into the past.

Ensemble learning: A family of predictors that togethor produce the final outcome.

Experiments consistently showed improvements with time decay reached best quality with no decay at all.

Problem Setup

- m users, $u, v \in U$.
- n items, $i, j \in V$.
- $r_{ui}(t)$ indicates preference of user u for item i at day t.
- Experiments are performed on Netflix data using RMSE.

Baseline Model

Matrix factorization setup. Each user is associated with a vector $p_u \in R^f$. Each item is associated with the vector $q_i \in R^f$.

Predictor

$$\hat{r_{ui}} = q_i^T p_u \tag{1}$$

$$min \sum_{(u,i,t)} (r_{ui} - q_i^T p_u)^2 + \lambda(\|q_i\|^2 + \|p_u\|^2) \tag{2}$$

Baseline Predictor

CF data exhibits large user and item biases - i.e. systematic tendencies for some user to give higher ratings.
A baseline predictor for r_{ui} is denoted by b_{ui}.

Baseline Predictor

$$b_{ui} = \mu + b_u + b_i \tag{3}$$

Integrated Baseline Predictor

$$b_{ui} = \mu + b_u + b_i + q_i^T \left(p_u + |R(u)|^{-1/2} \sum_{y \in R(u)} y_j \right) \tag{4}$$

- The overall average rating, μ
- The observed deviations of user u from the average rating, b_u
- The observed deviations of item i from the average rating, b_i
- The set of items rated by user u, $R(u)$.

Advantages of Decomposition

Decomposition of a rating into distinct portions allows us to treat different temporal aspects in seperation.

- User biases, b_u, change over time
- Item biases, b_i, change over time
- User preferences, p_u, change over time
- Item characteristics, p_i, rarely changes over time

Time Changing Baseline Predictor

Time Changing Bias

$$b_{ui} = \mu + b_u(t) + b_i(t) \tag{5}$$

Time Changing Item Bias

$$b_i(t) = b_i + b_{i,Bin(t)} \tag{6}$$

- b_{ui} is a real valued function.
- Split the item biases into time-based bins. For Netflix data, bins are 10 consecutive weeks.
- Works well for items but is more challanging for users.

User Changing Bias

Linear function to capture possible gradual drift of user bias.

$$b_u^1(t) = b_u + \alpha_u \cdot dev_u(t) \tag{7}$$

$$dev_u(t) = sign(t - t_u) \cdot |t - t_u|^{\beta} \tag{8}$$

t_u denotes the mean date of rating by user u. Where β is set by cross validation.

Simplied Learning from Venue Matrix

A more flexible model uses splines.

- We designate k_u time points - $\left\{t_1^u, \cdots, t_{k_u}^u\right\}$ - spaced uniformly across the dates of user's ratings as kernels that control the following function:

$$b_u^2(t) = b_u + \frac{\sum_{l=1}^{k_u} e^{-\gamma|t-t_1^u|} b_{t_l}^u}{\sum_{l=1}^{k_u} e^{-\gamma|t-t_l^u|}} \tag{9}$$

γ sets the smoothness of the spline.

Thus far, only smooth functions have been discussed to model user bias. There are also sudden drifts, "spikes". To address short lived changes, we assign a single parameter per user and day, absorbing the day specific variability. This parameter is denoted $b_{u,t}$. The notion of day can be exchanged with a user session.

$$b_u^3(t) = b_u + \alpha_u \cdot dev_u(t) + b_{u,t} \tag{10}$$

$$b_u^4(t) = b_u + \frac{\sum_{l=1}^{k_u} e^{-\gamma|t-t_1^u|} b_{t_l}^u}{\sum_{l=1}^{k_u} e^{-\gamma|t-t_l^u|}} + b_{u,t} \tag{11}$$

Learn Parameters

$$min \sum_{(u,i,t)\in\mathcal{K}} \left(r_{ui}(t) - \mu - b_u - \alpha_u dev_u(t) - b_{u,t} - b_i - b_{i,Bin(t)}\right)^2 \quad (12)$$

$$+\lambda \left(b_u^2 + \alpha_u^2 + b_{u,t}^2 + b_i^2 + b_{i,Bin(t)}^2\right) \quad (13)$$

Must learn parameters $b_u, \alpha_u, b_{u,t}, b_i$, and $b_{i,Bin(t)}$

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: $b_{ui}(t) = \mu + b_u + b_i + b_{i,\text{Bin}(t)}$.
- *linear* linear modeling of user biases: $b_{ui}(t) = \mu + b_u + \alpha_u \cdot \text{dev}_u(t) + b_i + b_{i,\text{Bin}(t)}$.
- *spline* spline modeling of user biases: $b_{ui}(t) = \mu + b_u + \frac{\sum_{l=1}^{k_u} e^{-\gamma|t-t_l^u|} b_{t_l}^u}{\sum_{l=1}^{k_u} e^{-\gamma|t-t_l^u|}} + b_i + b_{i,\text{Bin}(t)}$.
- *linear+* linear modeling of user biases and single day effect: $b_{ui}(t) = \mu + b_u + \alpha_u \cdot \text{dev}_u(t) + b_{u,t} + b_i + b_{i,\text{Bin}(t)}$.
- *spline+* spline modeling of user biases and single day effect: $b_{ui}(t) = \mu + b_u + \frac{\sum_{l=1}^{k_u} e^{-\gamma|t-d_l|} b_{t_l}^u}{\sum_{l=1}^{k_u} e^{-\gamma|t-t_l^u|}} + b_{u,t} + b_i + b_{i,\text{Bin}(t)}$.

Time Changing Factor Model

Temporal dynamics also affect user preferences.

The same way we treat user biases we also treat each component of the user preferences $p_u(t)^T = \{p_{u1}(t), \dots, p_{uf}(t)\}$

$$p_{uk}(t) = p_{u,k} + \alpha \cdot dev_u(t) + p_{uk,t} \tag{14}$$

where $k = 1, \dots, f$.

Final Predictor

$$\hat{r}_{ui}(t) = \mu + b_i(t) + b_u(t) + q_i^T \left(p_u(t) + |R(u)|^{-1/2} \sum_{j \in R(u)} y_j \right) \qquad (15)$$

where $b_i(t)$, $b_u(t)$, and $p_u(t)$ were given by (6), (10), and (14).

Experiments

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.