Recommender systems : when memory matters

Aleksandra Burashnikova[1,3], Marianne Clausel[2], Massih-Reza Amini[3], Yury Maximov[4], Nicolas Dante[2]

[1] *Skolkovo Institute of Science and Technology*
[2] *University of Lorraine*
[3] *University Grenoble Alpes*
[4] *Los Alamos National Laboratory*

Introduction

- We put in evidence the impact of homogeneous user/items interactions for prediction, after removal of non-stationarities.
- We discuss the need of designing specific strategies to remove non-stationarities due to a specificity of recommender systems, namely the presence of memory in user/items interactions.
- We turn our preliminary study into a novel and successful strategy combining sequential learning per blocks of interactions and removing user with non–homogeneous behavior from the training.

Framework

We suggest to model homogeneity and non-stationarity of user feedbacks, using stationarity and memory mathematical tools developed for sequential data analysis.

- **Stationarity:**

  Definition 0.1 (Stationarity)**.** The sequence of user's feedback $X = X_t, t \in Z$ is said to be (wide-sense) stationary if its two first orders moments are homogeneous with time:

  $$\forall t, k, l \in \mathbb{Z},\ \mathbb{E}[X_t] = \mu, \text{ and}$$

  $$Cov(X_k, X_l) = Cov(X_{k+t}, X_{l+t})$$

  The autocovariance of a stationary process only depends on the difference between the terms of the series $h = k - l$. We set $\gamma(h) = Cov(X_0, X_h)$.

- **Memory: the definition is done in the Fourier domain and is based on the spectral density:**

  $$f(\lambda) = \frac{1}{2\pi} \sum_{h=-\infty}^{+\infty} \gamma(h) e^{-ih\lambda}, \lambda \in (-\pi, \pi].$$

  Definition 0.2 (Memory)**.** A time series X admits memory parameter $d \in \mathbb{R}$ iff its spectral density satisfies :

  $$f(\lambda) \sim \lambda^{-2d} \text{ as } \lambda \to 0\,.$$

  When the memory parameter is large, the time series tends to have a sample autocorrelation function with large spikes at several lags which is the signature of non-stationarity.

- **GPH memory estimator:**

  - **One first defines a biased estimator of the spectral density function $I(\lambda)$ and evaluate it on $\lambda_k = \frac{2\pi k}{N}$, N is the length of the sample :**

    $$I_N(\lambda_k) = \frac{1}{N} \left| \sum_{t=1}^{N} X_t e^{it\lambda_k} \right|^2$$

  - **The estimator of the memory parameter is therefore:**

    $$\hat{d}(m) = \frac{\sum_{k=1}^{m} (Y_k - \bar{Y}) \log(I(\lambda_k))}{\sum_{k=1}^{m} (Y_k - \bar{Y})^2},$$

    where $Y_k = -2\log|1 - e^{i\lambda_k}|$, $\bar{Y} = (\sum_{k=1}^{m} Y_k)/m$ and m is the number of used frequencies.

Learning Scheme

The goal of the sequential part of our algorithm is to learn a relevant representation of the couples users/items $\omega = (U, V)$ where $U = (U_u)$, $V = (V_i)$ are low-dimensional vectors. Weights are updated by minimizing the ranking loss corresponding to this block constituted by non-preferred items, N_u^t, followed by preferred ones Π_u^t:

$$\hat{L}_{\mathcal{B}_u^t}(\omega_u^t) = \frac{1}{|\Pi_u^t||\mathrm{N}_u^t|} \sum_{i \in \Pi_u^t} \sum_{i' \in \mathrm{N}_u^t} \ell_{u,i,i'}(\omega_u^t)\,,$$

where $\ell_{u,i,i'}$ is the logistic loss:

$$\ell_{u,i,i'} = \log\left(1 + e^{-y_{u,i,i'} U_u (V_i - V_{i'})}\right) + \lambda \left(\|U_u\|_2^2 + \|V_i\|_2^2 + \|V_{i'}\|_2^2\right)$$

with $y_{u,i,i'} = 1$ if the user u prefers item i over item i', $y_{u,i,i'} = -1$ otherwise.

Dataset

We have considered four publicly available benchmarks, for the task of personalized Top–N recommendation:

Data	$\lvert U\rvert$	$\lvert Stat\_U\rvert$	$\lvert I\rvert$	Sparsity	Avg. # of +	Avg. # of −
Kassandr	2,158,859	26,308	291,485	.9999	2.42	51.93
Pandor	177,366	9,025	9,077	.9987	1.32	10.36
ML-1M	6,040	5,289	3,706	.9553	95.27	70.46
Outbrain	49,615	36,388	105,176	.9997	6.1587	26.0377

Identifying stationary users. We keep only users whose embeddings have four stationary components, using a preliminary estimation of the memory parameter.

Quality metrics estimation

- `BPR`: a stochastic gradient-descent algorithm, based on bootstrap sampling of training triplets.
- `GRU4Rec`: considers the session as the sequence of clicks of the user and learns model parameters by optimizing a regularized approximation of the relative rank of the relevant item.
- `Caser`: CNN based model that embeds a sequence of clicked items into a temporal image and find local characteristics of the temporal image using convolution filters.
- `MOSAIC`: our algorithm, allowing to include stationarity in the pipeline.
- `SAROS`: our approach trained on the full dataset

	MAP@5				MAP@10			
	ML-1M	Kasandr	Pandor	Outbrain	ML-1M	Kasandr	Pandor	Outbrain
BPR	.826	.522	.702	.573	.797	.538	.706	.537
Caser	.718	.130	.459	.393	.694	.131	.464	.397
GRU4Rec	.777	.689	.613	.477	.750	.688	.618	.463
SAROS	.832	.705	.710	.600	.808	.712	.714	.563
MOSAIC	**.842**	**.706**	**.711**	**.613**	**.812**	**.713**	**.715**	**.575**

	NDCG@5				NDCG@10			
	ML-1M	Kasandr	Pandor	Outbrain	ML-1M	Kasandr	Pandor	Outbrain
BPR	.776	.597	.862	.560	.863	.648	.878	.663
Caser	.665	.163	.584	.455	.787	.198	.605	.570
GRU4Rec	.721	.732	.776	.502	.833	.753	.803	.613
SAROS	.788	**.764**	**.863**	.589	.874	**.794**	.879	.683
MOSAIC	**.794**	**.764**	**.863**	**.601**	**.879**	**.794**	**.880**	**.692**

Conclusion

- We introduce a strategy to filter the dataset with respect to homogeneity of the behavior in the users when interacting with the system, based on the concept of memory.
- From the results, it comes out that taking into account the memory in the case where the collection exhibits long range dependency allows to enhance the predictions of the proposed sequential model.