

Paper Summary

David Miller
CIS 5930: Social Network Mining

February 26, 2018

Current recommender systems provide accurate recommendations to static data, that is temporal dynamics are inferred after they are observed. The Recurrent Recommender Network (RRN) is able to predict future behavior trajectories. The model computes update functions

$$\hat{r}_{ij|t} = f(u_{it}, m_{jt}) \text{ and } u_{i,t+1} = g(u_{it}, \{r_{ij|t}\}) \\ m_{j,t+1} = h(m_{jt}, \{r_{ij|t}\})$$

where $\hat{r}_{ij|t}$ denotes the predicted rating for user i on movie j , $r_{ij|t}$ is the actual rating, at time step t , and u_i and m_j be latent attributes for user i and movie j . The functions f , g and h are learned such that we can infer a new users state directly without need for further optimization [1]. The way the RNN algorithm determines these functions is through four steps:

1. Movie and user states that have ratings k_1, \dots, k_N for movies m_1, \dots, m_N and 0 otherwise (user is either new or has not seen the movie) for each time value t
2. Compute stationary rating components, such as long term preferences or liked genres, and incorporate them with profile vectors u_i and m_j from step 1
3. Take the latest observations and update states and make predictions based on the newly updated states.
4. Find optimal parameters that yield predictions that are close to actual ratings via

$$\min_{\theta} \sum_{(i,j,t) \in I_{train}} (r_{ij|t} - \hat{r}_{ij|t}(\theta))^2 + R(\theta)$$

where θ are all the parameters to be learned, $I_{learned}$ is the set of observed , and R denotes some regularization function [1].

In terms of results, RNN outperforms other models with respect to root mean square error (RMSE).

In figure 1 , RNN only used 20-dimensional stationary factors and 40-dimensional embedding factors whereas PMF and TimeSVD++ use dimensionality

160 and AutoRec used 500-dimensional embeddings. In terms of future work, I think this algorithm should be commercialized due to its accuracy with low dimensionality compared to current recommender systems in place. Another future direction they can take is to locally mount this "light-weight" algorithm on peoples computer to reduce power consumption of the company. It seems as if the algorithm may be efficient enough to be run on persoanl computers without causing any slowdowns.

	PMF	I-AR	U-AR	T-SVD	RRN
IMDb	2.3913	2.0521	2.0290	2.0037	1.9703
Netflix 6 months	0.9584	0.9778	0.9836	0.9589	0.9427
Netflix full	0.9252	0.9364	0.9647	0.9275	0.9224

Three strengths I found with the paper are

1. The accuracy RNN achieves with such low dimensionality.
2. While being sparse (90 times sparser than other methods), it does not suffer from this fact. This is due to the model learning the functions rather than the parameters.
3. RNN still provides accurate results for users with very few ratings.

Three weaknesses I found with the paper are

1. RNN relies on datasets that are private to Netflix, IMDb, etc.
2. Although accurate, users with few ratings produces noisy predictions.
3. Although smaller time steps allow for frequent updates and the ability to capture short term effects, it increase the time complexity (and inaccuracy) by increasing FLOPS.

Questions for the reader

1. What is the time complexity of RNN?
2. The authors are from Google, LinkedIn, UT Austin and Carnegie Melon. I assume Google will take most, if not all, of the copyrights for the algorithm but who "owns" this algorithm?
3. Need some clarity with Rating Emissions. Ask presenter about this.

References

- [1] Chao-Yuan Wu, Amr Ahmed, Alex Beutel, Alexander J. Smola, How Jing *Recurrent Recommender Networks*, KDD17, August 13-17, 2017, Halifax, NS, Canada.