Mathematical Challenge November 2018

Autoencoder-based collaborative filtering algorithms

References

1. Koren, Y.; Bell, R. & Volinsky, C. (2009), 'Matrix Factorization Techniques for Recommender Systems', Computer 42 (8), 30-37
2. Sarwar, B. M.; Karypis, G.; Konstan, J. A. & Reidl, J. (2001), Item-based collaborative filtering recommendation algorithms, in 'Proceedings of the 10th international conference on World Wide Web' pp. 285--295
3. Sedhain, S.; Menon, A. K.; Sanner, S. & Xie, L. (2015), Autorec: Autoencoders meet collaborative filtering, in 'Proceedings of the 24th International Conference on World Wide Web', pp. 111-112
4. Wang, H.; Wang, N. & Yeung, D.-Y. (2014), 'Collaborative Deep Learning for Recommender Systems.', CoRR abs/1409.2944
5. He, X., Liao, L., Zhang, H., Nie, L., Hu, X. and Chua, T.S., 2017, April. Neural collaborative filtering. In Proceedings of the 26th International Conference on World Wide Web (pp. 173-182). International World Wide Web Conferences Steering Committee.

Description

Recommender systems aim to exploit information about users' preferences for some items (e.g ratings provided with a product review upon purchase) to provide personalized product recommendations. Collaborative filters (CF) are a widely used model class for this purpose, where preferences are inferred leveraging on the concept of collaboration. I.e. a user A is deemed more likely to share an opinion on product P_j with user B rather than with a random user, given that users A and B have shared similar opinions on a set of products $\{P_i\}_{i=1}^{m}$. Owing to the Netflix challenge, plethora of approaches have been developed to solve this problem, mainly relying on matrix-factorization [1] and neighborhood model [2] approaches.

Lately, due to its success for vision, speech and natural language processing tasks, a few papers have proposed CF models based on neural network / autoencoder paradigm. Here, we will summarize the approaches taken in three most prominent papers.

Autoencoder unit for collaborative filtering [3]

In the cited paper, the authors consider the partially observed user-item rating matrix R_{ui} as stacked partially observed user and item vectors $r^{(u)} = (R_{u1}, \ldots R_{un}) \in R^n$ and $r^{(i)} = (R_{1i}, \ldots R_{mi}) \in R^m$ representations, respectively. The task is to reconstruct these partially observed vectors to predict missing ratings for purposes of recommendation, fitting the autoencoder model structure
(see figure 1).

In the simplest case, the autoencoder structure corresponds to a neural network with a single, k-dimensional hidden layer, whose parameters are learned backpropagating the reconstruction error. During the learning process the fact that the input vector is partially observed is accounted for by updating only the weights associated to observed inputs. Moreover, regularization is introduced to prevent overfitting on observed ratings. This leads to expression (1):

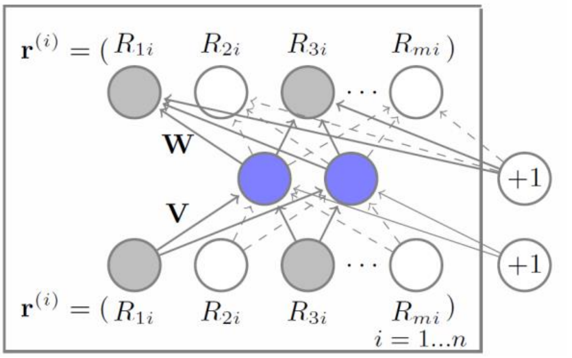

Figure 1: AutoRec: Autoencoder based collaborative filter

$$\min_{\theta} \left\| r^{(i)} - h\left(r^{(i)}; \theta\right) \right\|_O^2 + \frac{\lambda}{2} (\|W\|_F^2 + \|V\|_F^2)$$

Where O denotes the observed rating set, $\theta = \{W, V\}$ the set of network parameters, and $h(r; \theta)$ the reconstruction function.

Collaborative deep learning model [4]

In this paper (stacked denoising) autoencoders are used to determine a lower dimensional representation $X_{L/2} \in \mathbb{R}^{JxL/2}$ of a content-based representation of the items $X_c \in \mathbb{R}^{JxS}$ (where J denotes the set of items, while S denotes the set of item content-related features). This representation is in turn used to construct the item representation $V \in \mathbb{R}^{Jxk}$ in a rating latent space, which together with the (to be learned) user representation $U \in \mathbb{R}^{Jxk}$ determines the unobserved ratings R through a probabilistic SVD representation $R = \mathcal{N}(U^+V, C)$, trained with an EM algorithm. Figure 2 depicts the Bayesian representation of the hybrid model.

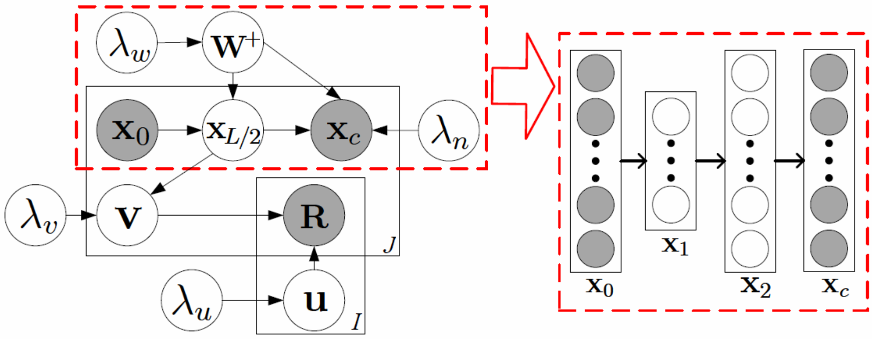

Figure 2 Collaborative deep learning model

Neural collaborative filtering [5]

Finally, a more recent related work builds up on [3]. Sparse inputs vectors (implicit feedbacks or one-hot encoded content features) are embedded into a lower dimensional space as dense representations, which are subsequently fed to a multi-layer neural classifier for the scores $\hat{y}_{ui}$. The proposed model is very flexible, and allows for integration of matrix-factorization-type perspective, and various objective paradigms, such as pairwise learning.

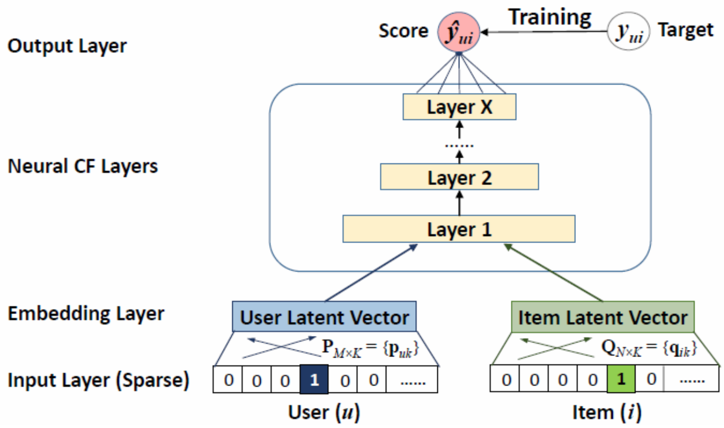

Figure 3 Neural based collaborative filtering

Questions

- *In [3], missing ratings are implicitly imputed with zeros during the training process. Suggest a more suitable way to handle missing ratings in the training phase.*
- *Long tail issue consists in bias introduced by frequently-rated items. Some items find themselves in the long tail (non-frequent items) and are therefore excluded from recommendations or wrongly ranked among recommended products. Suggest a way to remedy this issue using the autoencoder structure of the CF algorithm.*
- *Pairwise ranking is an objective paradigm that encourages ranking observed entries higher than unobserved ratings. Therefore, instead of minimizing the loss between $\hat{y}_{ui}$ and y_{ui}, pairwise learning maximizes the margin between observed entry $\hat{y}_{ui}$ and unobserved entry $\hat{y}_{uj}$. Suggest modifications to approaches outlined above, to utilize pairwise ranking.*
- *Investigate generic form of [5] which includes the matrix factorization approach, as outlined in the paper.*

We look forward to your opinions and insights.

Best Quant Regards,

swissQuant Group Leadership Team