Recommendation System : Leveraging Heterogeneous Information Networks in Deep Neural Network

Aakash Patel[1] **Shadab Ahmed**[1] **Ugesh Vayugundla**[2]

Abstract

Recommendation Systems are essentials for online as well as offline marketing. Heterogeneous information network (HIN) provides flexibility in modeling data heterogeneity, therefore, it has been adopted to characterize complex and heterogeneous auxiliary data in recommendation systems. Through modelling the rich object properties and relations in recommender system as a heterogeneous information network, our model, NACF, finds aspect-level latent factors and use them along with interaction history to produce top-N recommendations.

1. Introduction

In this age of internet, overloaded online information becomes overwhelming. Recommender System are used to guide users in a personalized way to recommend things of interest.

Collaborative Filtering is one of the most common and successful methods of recommendations. It uses users purchase history to recommend new items. Existing models usually focus on extracting latent factors of users and items by only exploiting their purchase history. But, latent factors are dependent on other factors as well. With the availability of a huge amount of side-information for users, users' characteristics and items' properties may reflect in many aspects.

In our project, we implement a Neural network based Aspect-level Collaborative Filtering model (NACF). NACF utilizes similarity information collected from HINs to effectively produce various aspect-level latent factors. It incorporates an attention mechanism to fuse these latent factors and output top-N recommendations.

[1]Department of Computer Science and Automation, IISC, Bangalore, India. [2]Department of Chemical Engineering, IISC, Bangalore, India.. Correspondence to: [1]Aakash Patel, [1]Shadab Ahmed <aakashpatel@iisc.ac.in>, <shadabahmed@iisc.ac.in> .

2. Preliminaries

2.1. Heterogenous Information Network

A heterogeneous information network (HIN) is a special type of information network with underneath data structure as a directed graph, which contains either multiple types of objects or multiple types of links.

Specifically, given a schema $S = (A, R)$ which consists of a set of entity types $A = \{A\}$ and a set of relations $R = \{R\}$, an information network is defined as a directed graph $G = (V, E)$ with an object type mapping function $\varphi : V \to A$ and a link type mapping function $\psi : E \to R$. If types of objects $| \ A \ | > 1$ or types of relations $| \ R \ | > 1$, the network is called heterogeneous information network.

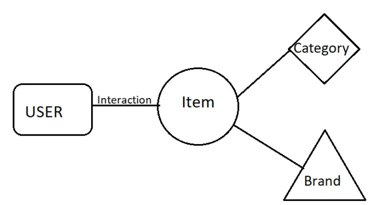

Figure 1. Network Schema for Amazon HINs.

2.2. Meta Paths

Objects in an HIN can be connected via different path which is called met-path. Formally defined, a meta-path P is a path defined on a schema S = (A,R), and is denoted in the form of$A_1 \xrightarrow{R_1} A_2 \xrightarrow{R_2} \cdots \xrightarrow{R_l} A_{l+1}$ (abbreviated as A1A2 $\cdots$ Al+1), which defines a composite relation $R = R_1 \circ R_2 \circ \cdots \circ R_l$ between type A1 and Al+1, where denotes the composition operator on relations. For the network schema shown in figure 1, we can have a meta-path *item-category-item* (ICI), *user-item-user* (UIU), etc. These meta-paths capture semantically different aspects. For eg.,UIU captures the similarity of users who have bought the same item. A meta-path is symmetric if it starts and ends with the same entity type.

2.3. Similarity Matrix

For various meta-paths, we quantify the similarity between two objects using similarity measures. There are several path-based similarity measures to evaluate the similarity of objects in HIN [(Lao N, 2010),(Shi C, 2014)]. PathSim (SunY, 2011) is one very commonly used similarity measure. It finds similarity between objects in the range [0,1]. In case of a symmetric meta-path, a similarity matrix for path *P*, S^P is a square matrix of size *no. of objects* x *no. of objects* such that s_{ij} is the similarity between $object_i$ and $object_j$.

3. The Model

3.1. Model Overview

The basic idea of NACF is to extract different aspect-level latent features for users and items, and then learn and fuse these latent factors with deep neural network. First, we construct the aspect level similarities in HINs for users and items under various meta-paths. We then design a neural network which finds aspect level latent factors for each user and item for each aspect. The input to this neural network is the similarity matrix. Finally, the latent factors corresponding to users (and items) for diffent aspects are fused together to form and overall latent factor for each user (and item). We discuss each of these steps in detail in further subsections.

3.2. Similarity Matrix Extraction

We use HINs to capture the underlying relations that exist in the recommender systems. We use meta-paths use find different aspect level similarities. For example, we take UIU and IUI meta paths to capture the purchase history factor. For a meta-path, there are many choices to capture the undying similarity exhibit by them. We choose similarity matrix for this purpose. We use pathsim (SunY, 2011) to calculate these similarities. Pathsim is a symmetric, similarity measuring method. It computes similarity between objects from HINs in the range [0,1]. An object has a similarity of 1 with itself. For our model we have used the following 4 metapaths:

1. **UIU**: Captures the purchase history of users . Users who have bought the same item.
2. **IUI**: Captures the purchase history of items. Items bought by the same user.
3. **ICI**: Captures the categorical similarity of items. Items having same category.
4. **UICIU**: Captures the categorical similarity of users. Users who have bought items having the same category.

3.3. Generating Latent Factors

At this stage, we have user-user and item-item similarity matrices for different aspects. The next step is to find the latent factors for users and items for each aspect. A deep neural network is used for this purpose. The architecture is show in the figure.

We extract the similarity of each user (or item) in from the similarity matrix for each user (or item) for each aspect. Concretely, for each user in each aspect, we extract the user's similarity vector from the aspect-specific similarity matrix. This acts as the input to the Multilayer Perceptron(MP). The MP takes this vector as input. At each layer, it projects the an input vector into a smaller vector space. Finally, it produces an output in a low dimensional vector space which acts as the latent feature for that user (or item) for that aspect. We have the following mapping in the MP:

$$
\begin{aligned}
&H_0 = S_i^{aspect_k}\\
&H_1 = f\left(W_1^T * H_0 + b_1\right)\\
&H_l = f\left(W_l^T * H_{l-1} + b_l\right)\\
&\cdots\\
&u_i^B = f\left(W_n^T * H_{n-1} + b_n\right)
\end{aligned}
$$

where $S_i^{aspect_k}$ is the i^{th} row in the similarity matrix for users for the aspect $aspect_k$ and u_i is the latent factor for that user. W_l and b_l are the weight matrix and bias for the l^{th} layer and $f(x)$ is RELU.

For each aspect we have an MP. Therefore, we will have 4 MP. Two to find latent factors for users and item for purchase history aspect and two to find latent factors for users and items for category aspect.

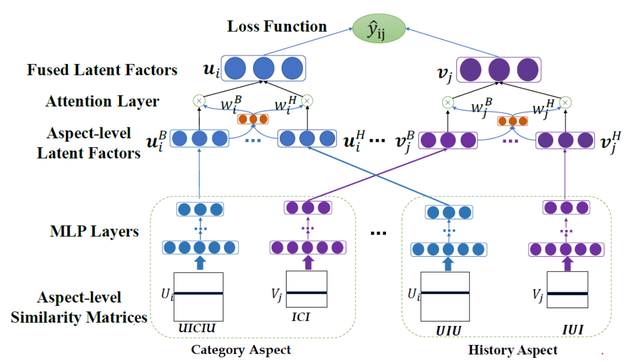

Figure 2. Deep Neural Network in NACF

3.4. Fusing Latent Factors

At this step, we want to fuse the latent factors for users for various aspects into one latent factor. Similarly, for the items. Each aspect has different importance to the recommendations. Therefore, we add them as weighted sum giving each for users and items separately. One could resort to domain knowledge to set these weights but that is not going to be very accurate. Thus, we use attention based mechanism to find the weights. For each aspect, for each users, we calculate an score using a two layer neural network, called the Attention Layer, according to the following equation:

$$s_i^{aspect_k} = W_2^T f\left(W_1^T * u_i^{aspect_k} + b_1\right) + b_2$$

where W_1, W_2 are the weight vectors and b_1, b_2 are the biases for the two layers of the network.
To calculate final weights for each aspect level latent factor, we normalize the above scores with the Softmax function. Therefore, the final weights for user, u_i, is given by the following expression:

$$w_i^{aspect_k} = \frac{\exp\left(s_i^{aspect_k}\right)}{\sum_{j=1}^{no.of aspects} \exp\left(s_i^{aspect_j}\right)}$$

And the fused latent factor for u_i is given by

$$\mathbf{u_i} = \sum_{j=1}^{no.of aspects} w_i^{aspect_j} \cdot u_i^{aspect_j}$$

3.5. Objective Function

We model the top-N recommendation as a classification problem. It predicts the probability of a user-item interaction. In use the sigmoid function to constrain the output value in the range [0,1]. The probability of interaction for user, u_i, and item, i_j is given by the expression:

$$\hat{y}_{ij} = sigmod\left(u_i * v_j\right) = \frac{1}{1 + e^{-u_i * v_j}}$$

where $\mathbf{u_i}$ and $\mathbf{i_j}$ are the fused latent factors. On the training set, the likelihood function is:

$$p\left(\mathcal{Y}, \mathcal{Y}^- | \Theta\right) = \prod_{i,j \in \mathcal{Y}} \hat{y}_{ij} \prod_{i,k \in \mathcal{Y}^-} \left(1 - \hat{y}_{ik}\right)$$

, where$\mathcal{Y}$ and $\mathcal{Y}^-$ are the positive and negative samples respectively. $\mathcal{Y}^-$ is obtained by sampling random interactions which have not occured yet.Θ is the model parameter.

The loss function used is binay cross entropy and is given as below:

$$Loss = -\sum_{i,j \in \mathcal{Y} \cup \mathcal{Y}^-} \left(y_{ij} \log \hat{y}_{ij} + \left(1 - y_{ij}\right) \log\left(1 - \hat{y}_{ij}\right)\right)$$

where y_{ij} is the ground truth. The optimization is done using stochastic gradient method.

4. Experiments

4.1. Datasets

We used publicly available Amazon dataset to evaluate our model performance. For Amazon dataset, we remove the users who buy less than 10 items and restricted total number of users to 3000 and total number of items to 1000 due to computational complexity. Amazon dataset contains users' rating data in Amazon. In our experiment, we select the items of Electronics categories for evaluation.

	#users	**#items**	**#ratings**	**#density**
Amazon	3532	3105	57,104	0.521%

4.2. Evaluation Metric

As suggested by previous work in recommendation, we use Hit Ratio (HR) and Normalized Discounted Cumulative Gain (NDCG) for performance evaluation. We adopt the leave-one-out method for evaluation. The latest rated item of each user is held out for testing, and the remaining data for training. For negative samples, we have randomly selected 99 items that are not rated by the users and rank the 100 sampled items for the users.

$$HR = \frac{\#hits}{\#users}, NDCG = \frac{1}{\#users} \sum_{i=1}^{\#users} \frac{1}{\log_2(p_i + 1)}$$

where $\#hits$ is the number of users whose test item appears in the recommended top k-list and p_i is the position of the test item in the list for the i-th hit. In our experiments, we truncate the recommendation list at K $\in$ [5, 10, 15, 20] for both metrics.

4.3. Results

We have achieved comparable results on Amazon dataset.

	k=5	**k=10**	**k=15**	**k=20**
HR	.2517	.3772	.5049	.6003
NDGC	.1610	.2007	.2344	.2570

5. Conclusion and Future Work

Deep Neural network is able to learn non-linear complex functions. In this work, we performed collaborative filtering for recommendation such that it also incorporates Heterogeneous Information Network.
One limitation of the this work is that for large datasets, computing similarities for different aspects becomes infeasible. So one solution that we propose is that, we can compute only top-k similarities(SunY, 2011) (k most similar users/items) for each aspect, and modify our neural network to incorporate this change.

References

Chuan Shi, e. a. Semantic path based personalized recommendation on weighted heterogeneous information networks. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, 2015.

Chuan Shi, e. a. Integrating heterogeneous information via flexible regularization framework for recommendation. In *Springer-Verlag London*, 2016.

Chuan Shi, Member, X. H. L. S. X. W. S. W. J. D. P. S. Y. Deep collaborative filtering with multi-aspect information in heterogeneous networks. In *IEEE Transactions on Knowledge and Data Engineering*, 2019.

Daniel D. Lee, H. S. S. Algorithms for non-negative matrix factorization. In *14th Annual Neural Information Processing Systems Conference, NIPS 2000 - Denver*, 2000.

Lao N, C. W. Fast query execution for retrieval models based on path constrained random walks. In *KDD*, 2010.

Lin, C. J. Projected gradient methods for non-negative matrix factorization. in neural computation. In *Neural Computation, pages 2756-2279*, 2007.

Shi C, Kong X, H. Y. Y. P. W. B. Hetesim: a general framework for relevance measure in heterogeneous networks. In *IEEE Trans Knowl Data Eng*, 2014.

SunY, Han J, Y. Y. P. W. T. Pathsim: meta path-based top-k similarity search in heterogeneous information networks. In *VLDB*, 2011.