A Practical Comparison of Denoising Autoencoder and Factorization Machine for Recommending Movies

CS420 Coursework: Recommender System

Runzhe Yang
RzAlpaca, 5140309562
Shanghai Jiao Tong University
yang_runzhe@sjtu.edu.cn

May 14, 2017

Abstract

In this coursework, the author implemented and compared two most successful models for Recommender System: Denoising Autoencoder (DAE) and Factorization Machine (FM). Although the DAE is a powerful unsupervised deep learning method to recover corrupted data, on the specific recommender system task it is still less powerful than traditional method such as the FM, because of the high sparsity of data and poor compatibility with side information.

1 Introduction

1.1 Background

The objective of a Recommender System is to automatically recommend some items or goods satisfying users, based on users behavioral history. The input data of a Recommender System is varied. Generally speaking, it can be divided into four main parts:

- User: user IDs, age, gender, number of watched movies, etc..
- Item: item IDs, category, release data, number of watchers, etc..
- Time: year, month, week, hour, etc..
- Observed ratings.

The output of a Recommender System is the prediction of some user's rating on some item. Inputs other than user IDs and item IDs and given ratings are called side information. Tuples $\langle userid, itemid, rating\rangle$ composed a sparse user-item rating matrix R. The goal of Recommender System can be restated as the blank filling in R with or without the auxiliary side information.

If only the sparse R is given, a traditional model of recovering R is the so-called *topic model*, which assume R is low-ranked and is the production of two small matrix U and V. Suppose there are u users and i items and R is rank t. Then $R = UV^T$ indicates that the

size of R is $u \times i$, the size of U is $u \times t$ and the size of V is $i \times t$. The t here can be interpreted as the number of topics. Therefore, U and V are user preference matrix and item attribute matrix in fact .

1.2 Collaborative Filtering

The Recommender System based on *Collaborative Filtering* use users past behaviors learn implicit information about user preference or similarity among users or items. Basically, there are two general methods: memory-based and model-based methods to do collaborative filtering. The memory-based method includes item-based k-nearest neighbors (knn), user-based knn[] and some variants with time information. These methods regard each item or user as a vector, then a quantitive similarity can be compute. Using pair-wise similarities we can predict a rating based on ratings from similar users or of similar items.

A typical model based method is mentioned in section 1.1, called matrix factorization[]. However, the simple matrix factorization such as SVD has every low capacity to capture the feature of data, and cannot integrate the side information. To address this problem, many variants like SVD++[] and timeSVD++[] are proposed. Deep learning as a advance representation extraction method, such as Autoencoder (AE) is also used in this area to improve the capacity of simple SVD. Factorization Machine can be seen as a generalization of SVD++, which is also a practical model-based method for the Recommender System.

2 Methodology

In this course work, the author mainly compared two models: Denoising Autoencoder (DAE) and Factorization Machine.

2.1 Item-Based Denoising Autoencoder

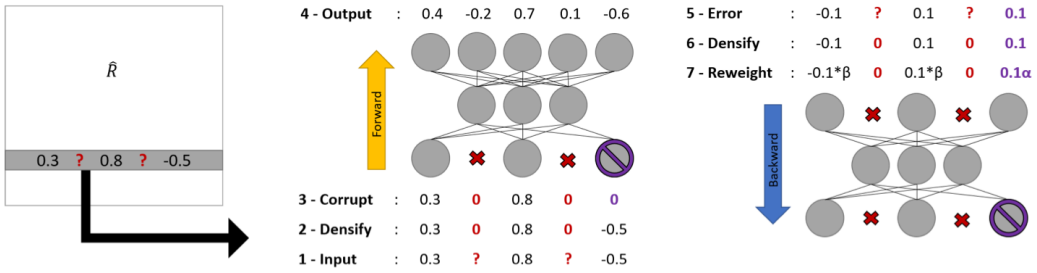

Figure 1: This illustration is borrowed from []. Feed Forward/Backward process for sparse Autoencoders. The sparse input is drawn from the matrix of ratings, unknown values are turned to zero, some ratings are masked (input corruption) and a dense estimate is finally obtained. Before backpropagation, unknown ratings are turned to zero error, prediction errors are reweighed by ℵ and reconstruction errors are reweighed by β .

Denoising Autoencoder is a extension of basic idea of autoencoder, which adds some noise to the training data deliberately. The DAE has to learn how to "reveal the mask". Hence it

learns robuster representation for the data, and therefore improves the generalization ability of the model.

For a given sparse rating matrix $R \in \mathbb{R}_{u \times i}$, a user is represented by a sparse row $u_i \in \mathbb{R}_i$ and item is represented by a sparse column $v_j \in \mathbb{R}_u$. The item-base autoencoder use the batch of item vectors $\{v_j\}$ as inputs. If an Autoencoder only has one hidden layer, the corresponding network is $nn(x) = \sigma(W_2\sigma(W_1 x + b_1) + b_2)$. Actually, it is analogue to "matrix factorization". The predicted vector $\hat{v}_j$ has the form

$$\hat{v}_j = nn(v_j) = \sigma\left([W_2' I_u]\begin{bmatrix}\sigma(W_1' v_j b_1) \\ b_2'\end{bmatrix}\right)$$

The training loss is

$$L_{\alpha,\beta}(\boldsymbol{x}, \tilde{\boldsymbol{x}}) = \alpha\left(\sum_{j \in \mathcal{C}(\tilde{\boldsymbol{x}} \cap \mathcal{K}(\boldsymbol{x})} [nn(\tilde{\boldsymbol{x}})_j - \boldsymbol{x}_j]^2\right) + \beta\left(\sum_{j \notin \mathcal{C}(\tilde{\boldsymbol{x}} \cap \mathcal{K}(\boldsymbol{x})} [nn(\tilde{\boldsymbol{x}})_j - \boldsymbol{x}_j]^2\right) + \Omega(\boldsymbol{W})$$

where $\mathcal{K}(x)$ are the indices of known values of x and $\mathcal{C}(x)$ are the indices of corrupted values of x.

2.2 Factorization Machine

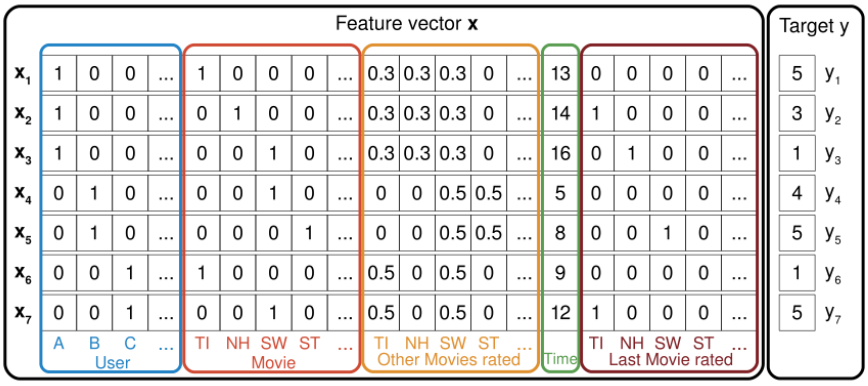

Figure 2: This illustration is b orrowed from []. Example (from Rendle [2010]) for representing a recommender problem with real valued feature vectors x. Every row represents a feature vector x i with its corresponding target y i . For easier interpreta-tion, the features are grouped into indicators for the active user (blue), active item (red), other movies rated by the same user (orange), the time in months (green), and the last movie rated (brown).

Factorization Machine is proposed to address the feature combination problem of sparse data. For a recommender system, FM can be used as a model exploit all pairwise combination of features. The factorization machine model of order $d = 2$ is defined as

$$\hat{y}(\boldsymbol{x}) = w_0 + \sum_{j=1}^{p} w_j x_j + \sum_{j=1}^{p} \sum_{j'=j+1}^{p} x_j x_{j'} \sum_{f=1}^{k} v_{j,f} v_{j',f}$$

where k is the dimention of the factorization. While training, we use MCMC to turn the parameters by seeing the model as a probabilistic graph.

3 Experiments and Results

3.1 Data

There are 94317 users and 99782 items in the given dataset. 5974450 records are the training data. 1524458 records are the test data.

The features the author fed to the model are

1. `userId`
2. `itemId`
3. `weekday`
4. `hour`
5. `movie_feature1`
6. `movie_feature2`
7. `1/number_of_watched movies of userId`
8. `1/number_of_watchers of itemId`

The author also do some observation on the training data. The code is in ‘‘`code/RecommenderSystem/data_observation.ipynb`"

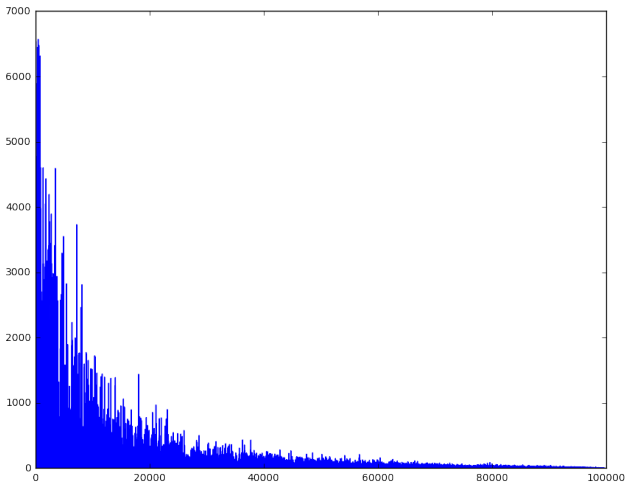

Figure 3: x-axis: movie id; y-axis: the number of watchers. It is a long-tail distribution.

3.2 Results

The best RMSE gained by the Denoising Autoencoder is about 1.51714 on the test set. The best RMSE gained by the Factorization machine is 1.33867. By linearly combining multiple submissions, the author got the final RMSE as 1.32984.

If time permits, I will revise this report [3, 6, 5, 1, 2, 4].

References

[1] X. Dong, L. Yu, Z. Wu, Y. Sun, L. Yuan, and F. Zhang. A hybrid collaborative filtering model with deep structure for recommender systems. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[2] S. Rendle. Factorization machines with libfm. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 3(3):57, 2012.

[3] R. Salakhutdinov, A. Mnih, and G. Hinton. Restricted boltzmann machines for collaborative filtering. In *Proceedings of the 24th international conference on Machine learning*, pages 791–798. ACM, 2007.

[4] S. Sedhain, A. K. Menon, S. Sanner, and L. Xie. Autorec: Autoencoders meet collaborative filtering. In *Proceedings of the 24th International Conference on World Wide Web*, pages 111–112. ACM, 2015.

[5] F. Strub, J. Mary, and R. Gaudel. Hybrid collaborative filtering with autoencoders. *arXiv preprint arXiv:1603.00806*, 2016.

[6] H. Wang, N. Wang, and D.-Y. Yeung. Collaborative deep learning for recommender systems. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1235–1244. ACM, 2015.